nAxiom

Leksykon nAxiom

Wersja nAxiom 1.15.8.1, data publikacji: 2026-09-01

1. Zmiany w Leksykonie nAxiom ... 7
2. Wstęp ... 8
2.1. Budowa aplikacji w nAxiom ... 12
2.2. Logowanie: Klient, Klient uproszczony ... 14
2.3. Logowanie: Konsultant ... 15
3. DANE ... 17
3.1. Kreator modeli danych ... 17
3.1.1. Okno Kreator modeli danych ... 18
3.1.2. Tworzenie nowego modelu danych ... 19
3.1.3. Definiowanie pól modelu danych ... 22
3.1.4. Nowe pole modelu danych ... 23
3.1.5. Tabela dokumentu biznesowego — kolumny systemowe ... 26
3.1.6. Edytowanie modelu danych ... 29
3.1.7. Nowa funkcja tabelaryczna ... 36
3.1.8. Powiązania tabel ... 40
3.1.9. Generowanie elementów aplikacji ... 42
3.1.10. Okno Generowanie elementów aplikacji ... 46
3.2. Źródła danych ... 48
3.3. Repozytoria załączników ... 50
3.4. Kategorie załączników ... 57
3.5. Słowniki ... 59
3.5.1. Słowniki proste ... 61
3.5.2. Słowniki hierarchiczne ... 64
4. INTERFEJS UŻYTKOWNIKA ... 66
4.1. Formularze ... 66
4.1.1. Właściwości formularza ... 72
4.1.2. Toolbar ... 81
4.1.3. Wiersz formularza ... 83
4.1.4. Kolumna formularza ... 85
4.1.5. Sekcja formularza ... 86
4.1.6. Kontrolki formularza ... 129
4.1.7. Uprawnienia ACL formularza ... 182
4.2. Formularze publiczne ... 190
4.3. Listy ... 193
4.3.1. Okno Kreator listy ... 195
4.3.2. Właściwości listy — karta Ogólne ... 199
4.3.3. Właściwości listy — karta Ustawienia listy ... 204

4.3.4. Właściwości listy — karta Edycja inline ... 214
4.3.5. Właściwości listy — karta Globalne ustawienia kolumn ... 218
4.3.6. Właściwości listy — karta Automatyczne odświeżanie ... 219
4.3.7. Właściwości listy - karta Wygląd ... 220
4.3.8. Właściwości listy — karta Pomoc dla użytkownika ... 222
4.3.9. Właściwości listy — karta Eksport listy ... 223
4.3.10. Kolumna listy — karta Ogólne ... 223
4.3.11. Kolumna listy — karta Wygląd ... 228
4.3.12. Kolumna listy — karta Widoczność ... 229
4.3.13. Kolumna listy — karta Eksportowanie danych ... 231
4.3.14. Kolumna listy — karta Edycja inline ... 231
4.3.15. Kolumna HTML ... 236
4.3.16. Właściwości przycisku — karta Ogólne ... 244
4.3.17. Właściwości przycisku — karta Dostępność ... 248
4.3.18. Właściwości przycisku — karta Notyfikacje ... 248
4.3.19. Przyciski listy do edycji inline ... 250
4.4. Pulpity aplikacji ... 253
4.5. Widżety ... 254
4.5.1. Obsługa biblioteki chart.js ... 259
4.6. Kontrolki ... 266
4.7. Menu użytkownika ... 269
4.8. Kreator stylów systemu ... 272
5. PROCESY BIZNESOWE ... 275
5.1. Dokumenty biznesowe (definicje dokumentów) ... 276
5.1.1. Karta Ogólne ... 278
5.1.2. Karta Reguły walidacji ... 280
5.1.3. Karta Blokady ... 281
5.1.4. Karta Audyt ... 282
5.1.5. Karty Uprawnienia, Uprawnienia statusów ... 286
5.1.6. Karta Uprawnienia — brak odczytu ... 289
5.1.7. Karta Eskalatory procesu ... 291
5.1.8. Karta Zdarzenia ... 292
5.1.9. Karta Formularze ... 295
5.1.10. Karta Interesariusze ... 295
5.1.11. Karta Pomoc dla użytkownika ... 297
5.2. Typy dokumentów ... 298
5.3. Autonumeracja dokumentów ... 300

5.4. Diagramy procesów ... 303
5.4.1. Edytor diagramów procesów ... 305
5.4.2. Wspólne właściwości bloków i przejść ... 309
5.4.3. Blok Start ... 318
5.4.4. Start warunkowy ... 318
5.4.5. Zakończenie ... 318
5.4.6. Zadanie ręczne ... 319
5.4.7. Akceptacja ... 319
5.4.8. Blok decyzyjny ... 321
5.4.9. Zadanie automatyczne ... 324
5.4.10. Oczekiwanie ... 325
5.4.11. Wiadomość ... 326
5.4.12. Dodatkowa akcja ... 326
5.4.13. Podprocesy ... 326
5.4.14. Podproces synchroniczny ... 330
5.4.15. Podproces asynchroniczny ... 332
5.4.16. Synchronizator ... 334
5.5. Definicje statusów ... 335
5.6. Modele decyzyjne DMN ... 336
5.6.1. Edytor modeli decyzyjnych DMN ... 337
5.6.2. ... 0
6. NARZĘDZIA ... 343
6.1. Akcje ... 344
6.1.1. Wspólne parametry akcji ... 347
6.1.2. Tabela *core.Actions* ... 349
6.1.3. Uprawnienia do rekordów ... 352
6.1.4. Przypisywanie użytkowników do grupy interesariuszy ... 357
6.1.5. Przypisywanie ról globalnych do interesariuszy ... 360
6.1.6. Przypisywanie ról w jednostkach organizacyjnych do interesariuszy ... 362
6.1.7. Odświeżanie uprawnień ... 365
6.1.8. Dodaj zastępstwo ... 368
6.1.9. Edytuj zastępstwo ... 369
6.1.10. Usuń zastępstwo ... 370
6.1.11. Zmiana statusu dokumentów ... 371
6.1.12. Przypisywanie zadań ... 374
6.1.13. Zapytania do webserwisów ... 376

6.1.14. SQL do bazy systemowej ... 383
6.1.15. SQL do baz zewnętrznych ... 385
6.1.16. SQL między bazami danych ... 387
6.1.17. Procedury bazodanowe ... 390
6.1.18. SQL zapisu danych ... 391
6.1.19. JavaScript ... 393
6.1.20. Wyślij e-mail ... 395
6.1.21. Import danych z Excel ... 397
6.1.22. Eksport danych do Excel ... 400
6.1.23. Konwersja Word do PDF ... 401
6.1.24. Generowanie PDF z Word ... 403
6.1.25. Raportowanie ... 405
6.1.26. Skanowanie kodów QR/BC (WebAPI) ... 408
6.1.27. Otwieranie formularza ... 412
6.1.28. Otwieranie listy ... 417
6.1.29. Odśwież listę u wszystkich użytkowników ... 418
6.1.30. Odświeżanie sekcji formularza ... 419
6.1.31. Sprawdzenia przed zapisem ... 420
6.1.32. Wysyłanie powiadomień ... 421
6.1.33. Wyświetlanie komunikatów ... 425
6.1.34. Magazyn Local Storage ... 427
6.1.35. Szyfrowanie plików ... 428
6.1.36. Drukowanie ZPL ... 431
6.1.37. Generowanie elementów aplikacji ... 434
6.1.38. Tworzenie pliku tekstowego ... 439
6.1.39. Otwieranie linku zewnętrznego ... 449
6.1.40. Akcja przełączania aplikacji ... 450
6.1.41. Algorytm C# ... 451
6.1.42. Importowanie dokumentów ... 483
6.1.43. Utwórz PDF z warstwą tekstową ... 486
6.1.44. Akcja odczytu kodów QR/BC z plików ... 488
6.1.45. Uruchom model decyzyjny ... 490
6.1.46. Akcje systemowe ... 491
6.2. Szablony word ... 510
6.3. Raporty ... 517
6.3.1. Web Report Designer ... 518
6.3.2. Projektowanie raportów ... 520

6.4. Szablony e-mail ... 530
7. INTEGRACJE ... 533
7.1. Mail Monitor ... 533
7.1.1. Karta Ogólne ... 534
7.1.2. Karta Akcje ... 539
7.1.3. Karta Postproces ... 541
7.2. Zadania cykliczne ... 543
8. ADMINISTRACJA ... 547
8.1. Logi środowiska i bieżącego tenanta ... 547
8.2. Ustawienia systemu ... 549
8.3. Tłumaczenia i pomoc ... 565
8.4. Lista aplikacji ... 571
8.4.1. Karta Ogólne ... 572
8.4.2. Karta Moduły ... 573
8.4.3. Karta Wersje ... 575
8.4.4. Karta Zmienne ... 576
8.4.5. Karta Uprawnienia ... 580
8.4.6. Karty Skrypt SQL przed importem/po imporcie ... 581
8.5. Role biznesowe ... 583
8.6. Licencjonowanie ... 586
8.6.1. Import pliku licencji ... 590
8.7. Migrator aplikacji ... 592
9. Budowanie i debugowanie aplikacji nAxiom ... 600
9.1. Wyszukiwanie pełnotekstowe w nAxiom ... 600
9.2. Smart numbers ... 606
9.3. Profile użytkowników ... 610
9.3.1. Listy użytkowników ... 611
9.3.2. Tworzenie i edycja profilu użytkownika ... 613
9.3.3. Tworzenie nowego konsultanta ... 617
9.3.4. Edytowanie własnego profilu użytkownika ... 618
9.3.5. Logowanie za pomocą hasła ... 619
9.4. Synchronizacja SQL ... 620
9.5. Model uprawnień PBA ... 642
9.5.1. Inicjowanie modelu uprawnień PBA ... 643
9.5.2. Wyznaczanie administratora uprawnień PBA ... 646
9.5.3. Uprawnienia administracyjne ... 647
9.5.4. Ustawianie uprawnień ról/użytkowników ... 648

9.5.5. Obszar Zarządzanie strukturą organizacyjną i użytkownikami ... 649
9.5.6. Obszar Użytkowanie aplikacji ... 654
9.5.7. Obszar Konfiguracja systemu ... 657
9.5.8. Obszar Budowanie aplikacji ... 658
9.6. Uprawnienia ACL do dokumentów ... 660
9.6.1. Włączanie obsługi uprawnień ... 661
9.6.2. Nowy szablon uprawnień do dokumentu ... 664
9.6.3. Włączanie weryfikacji uprawnień ... 666
9.6.4. Administrowanie uprawnieniami ACL ... 667
9.6.5. Mechanika uprawnień ACL (dokumenty) ... 669
9.7. Struktura organizacyjna ... 675
9.7.1. Dane dodatkowe jednostek struktury organizacyjnej ... 677
9.8. Diagnostyka w nAxiom ... 678
9.8.1. Logi bezpieczeństwa ... 680
9.8.2. Konfiguracja logowania serwisów nAxiom ... 684
9.8.3. Logi serwisowe ... 687
9.8.4. MiniProfiler for .NET ... 692
9.9. Obsługa modelu trzech środowisk ... 692
9.10. Integracja nAxiom z usługami Microsoft 365 ... 696
9.11. Wysyłanie i odbieranie korespondencji ePUAP ... 712
9.12. Definiowanie pomocy w aplikacjach biznesowych ... 752
9.12.1. Dodawanie treści pomocy ... 753
9.12.2. Definiowanie formularza pomocy ... 755
9.12.3. Akcje przycisku pomocy ... 758
9.12.4. Przykład pomocy w aplikacji ... 760
9.13. Predefiniowane funkcje i procedury ... 761
9.13.1. core.PrepareEmailToSend ... 762
9.13.2. core.SubstitutionsPerformedByUser ... 764
9.13.3. core.GetAllSubstitutions ... 766
9.13.4. core.CalculateOUId ... 767
9.13.5. JavaScript: showCustomMessage ... 769
9.13.6. core.GetCurrentDocumentACLPermissions ... 770
9.13.7. core.OUInstanceNameTranslations_View ... 772
10. FrontSPA — przewodnik użytkownika ... 772
10.1. Tryb pojedynczej aplikacji ... 773
10.2. Pulpity użytkownika ... 774

10.3. Powiadomienia i alerty ... 776
10.4. Czat ... 780
10.5. Widok listy ... 783
10.6. Obsługa słowników ... 791
10.7. Struktura organizacyjna ... 794
10.8. Akcje zdarzeń systemowych ... 797
10.9. Zastępstwa ... 800
10.10. Aplikacja nAxiom na urządzeniach mobilnych ... 805
11. Obsługa wielu tenantów w nAxiom ... 814
11.1. Wdrożenie obsługi wielu tenantów ... 816
11.2. Zarządzanie tenantami ... 817
11.3. Repozytoria tenantów ... 830
11.4. Ustawienia konfiguracyjne w TenantsAdminSPA ... 833

1. Zmiany w Leksykonie nAxiom

Opis zmian wprowadzonych w Leksykonie nAxiom od ostatniego wydania. Zmiany dotyczą zarówno nowych i zmienionych funkcjonalności z bieżącej wersji, jak i zmian redakcyjnych Leksykonu.

Lp.	Temat	Zakres zmian	Wersja
1	Toolbar	Uwaga dot. GUI właściciela zadania	1.15.8.0
2	Sekcja typu lista	Uwaga dotycząca składni z prefixem Section	1.15.8.1
3	Typowe właściwości kontrolek	Atrybut Tooltip, maksymalny rozmiar	1.15.8.1
4	Uprawnienia ACL formularza	Przeredagowano opis typu walidacji uprawnień	1.15.8.1
5	Właściwości listy — karta Ustawienia listy	Dodatkowe informacje o selekcji rekordów	1.15.8.1
6	Definicja dokumentu, Eskalatory procesu	Uwaga o konieczności restartu	1.15.8.0
7	Wspólne właściwości bloków i przejść	Uwaga o konieczności restartu	1.15.8.0
8	SQL do bazy systemowej	Dodano przykład SQL z odwołaniem do tablicy *Selected*	1.15.8.0
9	JavaScript	Odwołania do kontrolek formularza	1.15.8.0
10	Akcja systemowa Blokuj dokument	Korekta, możliwość użycia akcji	1.15.8.0
11	Akcja systemowa Odblokuj dokument	Korekta, możliwość użycia akcji	1.15.8.0
12	Tworzenie i edycja profilu użytkownika	Obsługa dwu ról systemowych	1.15.8.0
13	Włączanie weryfikacji uprawnień	Zmiana sposobu włączania kontroli uprawnień	1.15.8.0

Wpisy oznaczone gwiazdką (*) oznaczają zmiany wprowadzone po wydaniu wersji.

2. Wstęp

O TYM PODRĘCZNIKU

Leksykon nAxiom to kompendium wiedzy o platformie nAxiom. W tym podręczniku szczegółowo opisano funkcje i narzędzia przeznaczone do budowania aplikacji. Jest ona przeznaczona przede wszystkim dla użytkowników tworzących aplikacje w nAxiom (konsultantów) oraz administratorów odpowiedzialnych za konfigurowanie środowiska platformy.

Układ materiału jest zgodny z organizacją menu głównego aplikacji *AdminSPA*, dodatki zawierają podstawowe informacje dotyczące pracy użytkownika końcowego w aplikacji *FrontSPA* oraz czynności wykonywanych przez użytkownika z rolą *Administrator* związanych z tworzeniem profili użytkowników oraz nadawaniem im uprawnień do określonych funkcjonalności platformy.

Oprócz Leksykonu nAxiom dostępnych jest szereg innych publikacji na temat nAxiom. I tak, użytkownicy, którzy dopiero zaczynają pracę z nAxiom, mogą znaleźć opis procedury tworzenia prostej aplikacji w publikacji **Pierwsze kroki**. Zagadnienia związane z instalacją platformy opisano natomiast w **Przewodniku instalacji**. Ponadto dostępnych jest kilkanaście artykułów poświęconych wybranym funkcjonalnościom i zagadnieniom dotyczącym korzystania z nAxiom.

ARCHITEKTURA NAXIOM

Platforma nAxiom to rozwiązanie *low-code* (z ang. *prawie bez kodu*, w użyciu tłumaczenie *niskokodowe*), które umożliwia tworzenie i publikowanie aplikacji przez osoby niebędące zawodowymi informatykami. Pod względem technicznym jest to działająca w przeglądarce aplikacja internetowa (webowa), która przechowuje dane dotyczące budowanych aplikacji i dane biznesowe w bazie danych SQL Server.

nAxiom oferuje gotowe elementy do wykorzystania w aplikacjach, między innymi obsługę uwierzytelniania i kont użytkowników, obsługę struktury organizacyjnej, łączniki do zewnętrznych źródeł danych, funkcje rozpoznawania tekstu na zeskanowanych dokumentach (OCR), czy obsługę wielu wersji językowych. Ponadto dostępnych jest szereg narzędzi umożliwiających definiowanie między innymi tabel, formularzy, list, diagramów przepływu pracy i innych. Niektóre z tych narzędzi umożliwiają pracę w trybie graficznym.

nAxiom to aplikacja zbudowana w architekturze mikroserwisów, obejmująca kilkanaście serwisów dedykowanych do realizacji określonych zadań. Można je podzielić na dwie główne grupy: przeznaczone do obsługi *front-endu* i oparte na platformie *Angular* oraz przeznaczone do obsługi *back-endu* i oparte na platformie *.NET*.

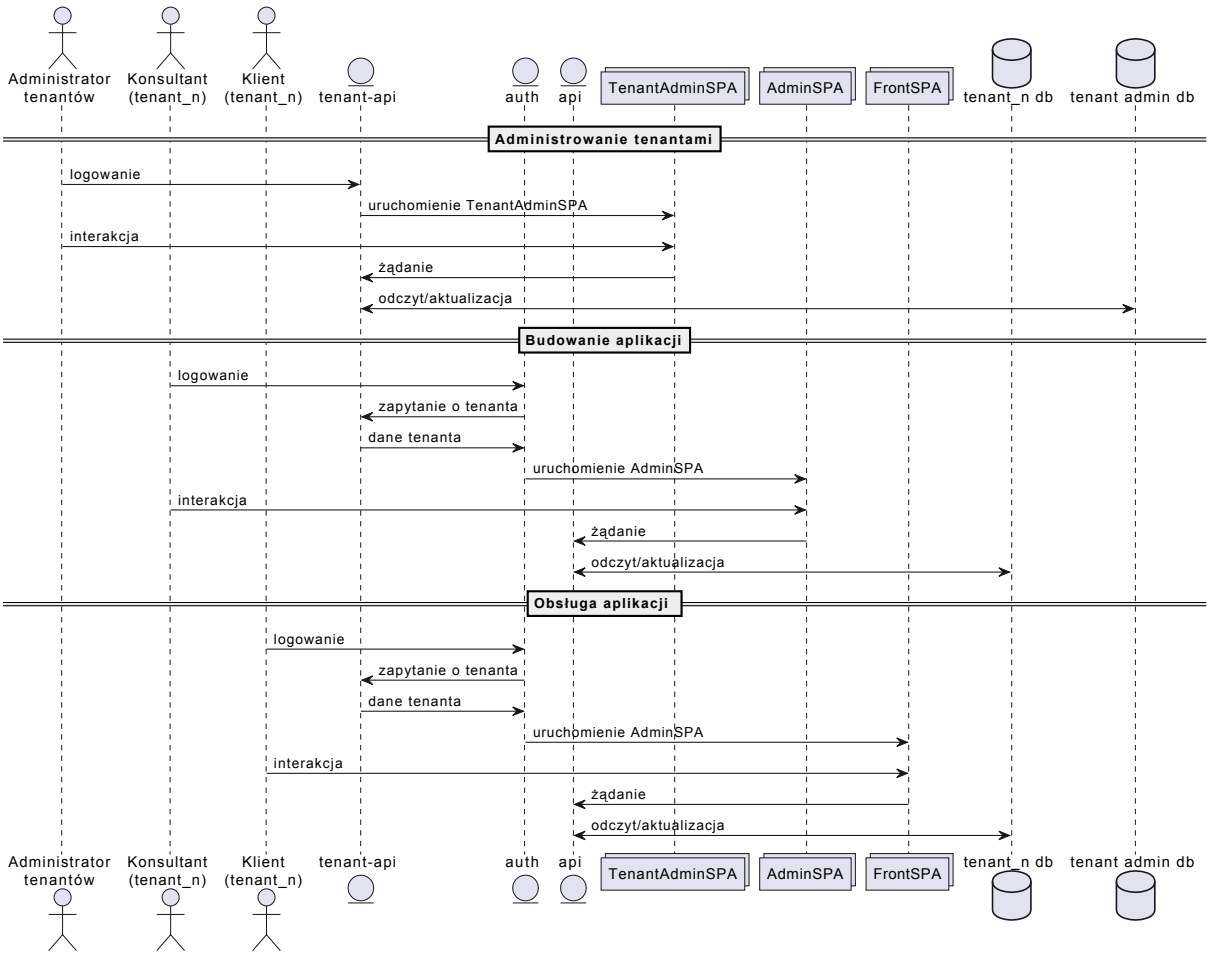
Platforma nAxiom składa się z następujących serwisów (w nawiasie podano typ serwisu):

- **auth** (dotnet): uwierzytelnianie użytkowników
- **front** (angular): obsługa aplikacji biznesowych przez użytkowników końcowych
- **admin** (angular): tworzenie aplikacji biznesowych przez konsultantów

- **tenant-admin** (angular): zarządzanie tenantami w środowisku *multi tenant*
- **api** (dotnet): obsługa żądań do bazy danych wysyłanych z serwisu *admin* i *front*
- **tenant-api** (dotnet): obsługa żądań do bazy danych wysyłanych z serwisu *tenant-admin*
- **task-service** (dotnet): obsługa zadań cyklicznych, wysyłki e-mail, śledzenia terminów
- **workflow** (angular): definiowanie diagramów procesów biznesowych przez konsultantów
- **doc-api** (dotnet): obsługa pomocy kontekstowej i funkcji generowania dokumentacji aplikacji biznesowych
- **public-api** (dotnet): dostęp do publicznego interfejsu API platformy nAxiom
- **mobile-api** (dotnet): obsługa aplikacji mobilnej nAxiom
- **syncfusion-api** (dotnet): obsługa generowania, szyfrowania i podglądu plików PDF
- **telerik-reports** (dotnet): obsługa definiowania, generowania i podglądu raportów Telerik
- **crystal** (dotnet): obsługa definiowania, generowania i podglądu raportów Crystal Reports
- **ocr** (angular): obsługa definiowania szablonów dokumentów dla funkcji OCR
- **ocr-api** (dotnet): obsługa żadań dotyczących rozpoznawania dokumentów przez funkcję OCR

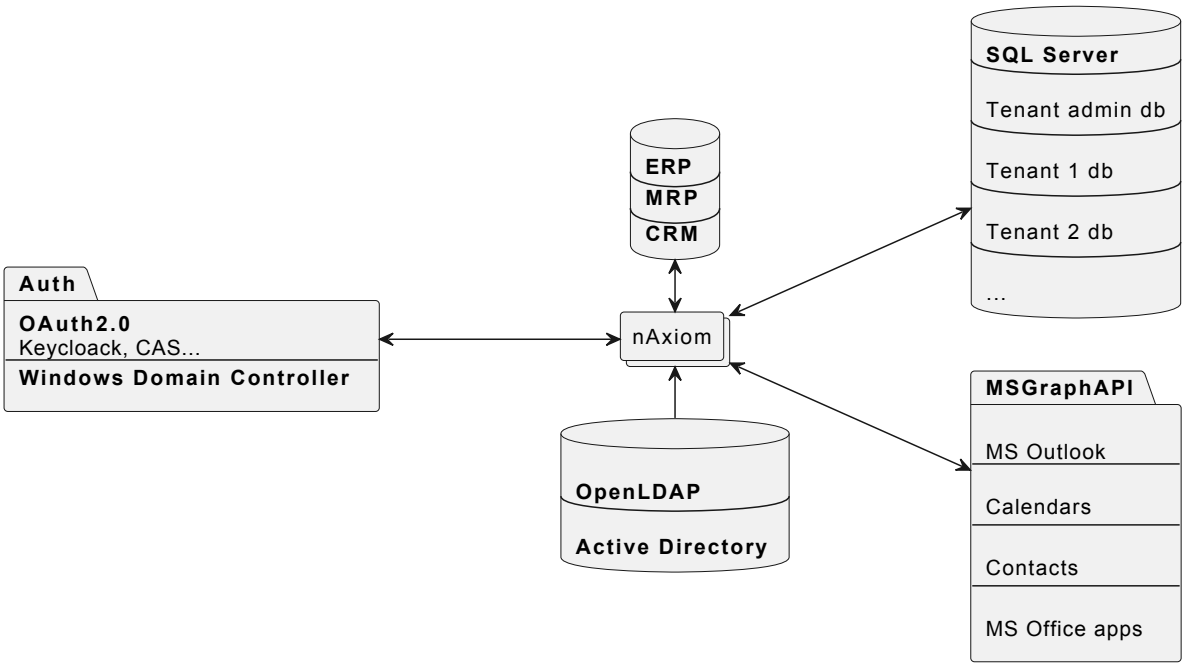
Niezbędnym uzupełnieniem środowiska nAxiom jest serwer bazy danych MS SQL Server. Współpraca nAxiom z bazą danych jest realizowana przy użyciu platformy *Entity Framework*.

Poniżej zamieszczono diagram prezentujący w uproszczeniu komunikację między poszczególnymi komponentami dla różnych scenariuszy.



Na diagramie ze względu na czytelność ograniczono się tylko do jednego tenanta, czyli organizacji korzystającej z własnych aplikacji i osobnej bazy danych. Platforma nAxiom umożliwia konfigurowanie środowisk z wieloma tenantami.

Ponadto, nAxiom można integrować z różnymi systemami zewnętrznymi, takimi jak systemy ERP, serwery uwierzytelniania i usługi katalogowe, co przedstawia poniższy diagram.



Platforma nAxiom jest aplikacją WWW przeznaczoną do uruchamiania na serwerze IIS w systemach Windows. Możliwe jest wdrożenie nAxiom w systemie Linux na podstawie obrazów Docker. W takim przypadku używany jest serwer WWW nginx.

2.1. BUDOWA APLIKACJI W NAXIOM

Podsumowanie:
W nAxiom można budować skalowalne, bezpieczne aplikacje umożliwiające obsługę dużych grup użytkowników. Naturalnymi środowiskami zastosowań nAxiom są przedsiębiorstwa, instytucje, urzędy, placówki edukacyjne i inne organizacje. W szczególności platforma oferuje wsparcie dla procesów biznesowych i ich automatyzacji. Aplikacje mogą być udostępniane jako portale, w których użytkownicy wykonują różne czynności odpowiednio do posiadanych uprawnień.
Przykładowe dziedziny zastosowań aplikacji zbudowanych w nAxiom to:
- obiegi dokumentów
- obsługa sprzedaży
- obsługa zgłoszeń
- wspomaganie produkcji
- zarządzanie łańcuchem dostaw
- zarządzanie projektami

Do budowy aplikacji służy w nAxiom mikroserwis *AdminSPA*, do którego mogą logować się użytkownicy z rola systemową *konsultant*. Gotowe aplikacje są publikowane w mikroserwisie *FrontSPA*, gdzie mogą z nich korzystać użytkownicy końcowi (rola systemowa *klient*).

Aplikacje w nAxiom są definiowane jako kontenery, w których agreguje się osobne funkcjonalności. Budowanie aplikacji w nAxiom polega zatem na definiowaniu kolejnych funkcjonalności. Takie podejście zapewnia większą elastyczność w rozbudowie aplikacji oraz określaniu praw dostępu do funkcjonalności w ramach aplikacji.

Przystępując do definiowania funkcjonalności, najpierw należy określić dane, które będą przetwarzane w aplikacji oraz zdefiniować tabelę bazy danych do przechowywania tych danych. Ta tabela to *model danych*.

Następnie należy uruchomić automatycznego kreatora, który na podstawie tej tabeli utworzy makietę przyszłej aplikacji. Ta makieta składa się między innymi z następujących składników:

- **Dokument biznesowy**: inaczej *definicja dokumentu*, to zbiór informacji dotyczących przetwarzania danych z konkretnego modelu danych. Definicja dokumentu określa między innymi konfigurację audytu danych dokumentu, szablonów uprawnień ACL i grup interesariuszy, zdarzeń w przetwarzaniu dokumentu i wyzwalanych przez nie akcji.
- **Diagram procesu**: model procesu biznesowego ilustrujący przejścia pomiędzy różnymi statusami dokumentu.
- **Widok listy**: obiekt interfejsu użytkownika umożliwiający przeglądanie danych (dokumentów) z modelu danych.
- **Widok formularza**: obiekt interfejsu użytkownika umożliwiający wprowadzanie i edycję danych (dokumentów) oraz zawierający elementy sterujące przepływem działań w aplikacji.

Kolejny etap to rozbudowa wygenerowanej automatycznie makiety funkcjonalności zgodnie z potrzebami klienta. W edytorze diagramów procesów (*workflow*) definiuje się przebieg przetwarzania, używając różnych bloków (zadanie ręczne, blok decyzyjny, akceptacja i inne) i łącząc je przejściami. Definiowane na tych przejściach przyciski są automatycznie generowane na formularzu, co zapewnia, że przetwarzanie przebiega zgodnie z diagramem.

Publikacja gotowych aplikacji w nAxiom polega na zdefiniowaniu pozycji menu użytkownika, powiązaniu tej pozycji z widokiem listy lub formularza i włączeniu jej w uprawnieniach roli biznesowej. Wykonanie tych czynności spowoduje, że użytkownik z tą rolą biznesową, będzie mógł korzystać z aplikacji po zalogowaniu się w nAxiom.

Jednakże wymaga to wcześniejszego przygotowania środowiska biznesowego danej organizacji, to jest zdefiniowania ról biznesowych, zdefiniowania profili użytkowników i przypisania im ról oraz przypisania uprawnień PBA do korzystania z aplikacji. W nAxiom definiuje się standardowo cztery kategorie użytkowników, którym przypisuje się odpowiednie, predefiniowane role systemowe. Użytkownicy z rolą `Klient` i `Klient`

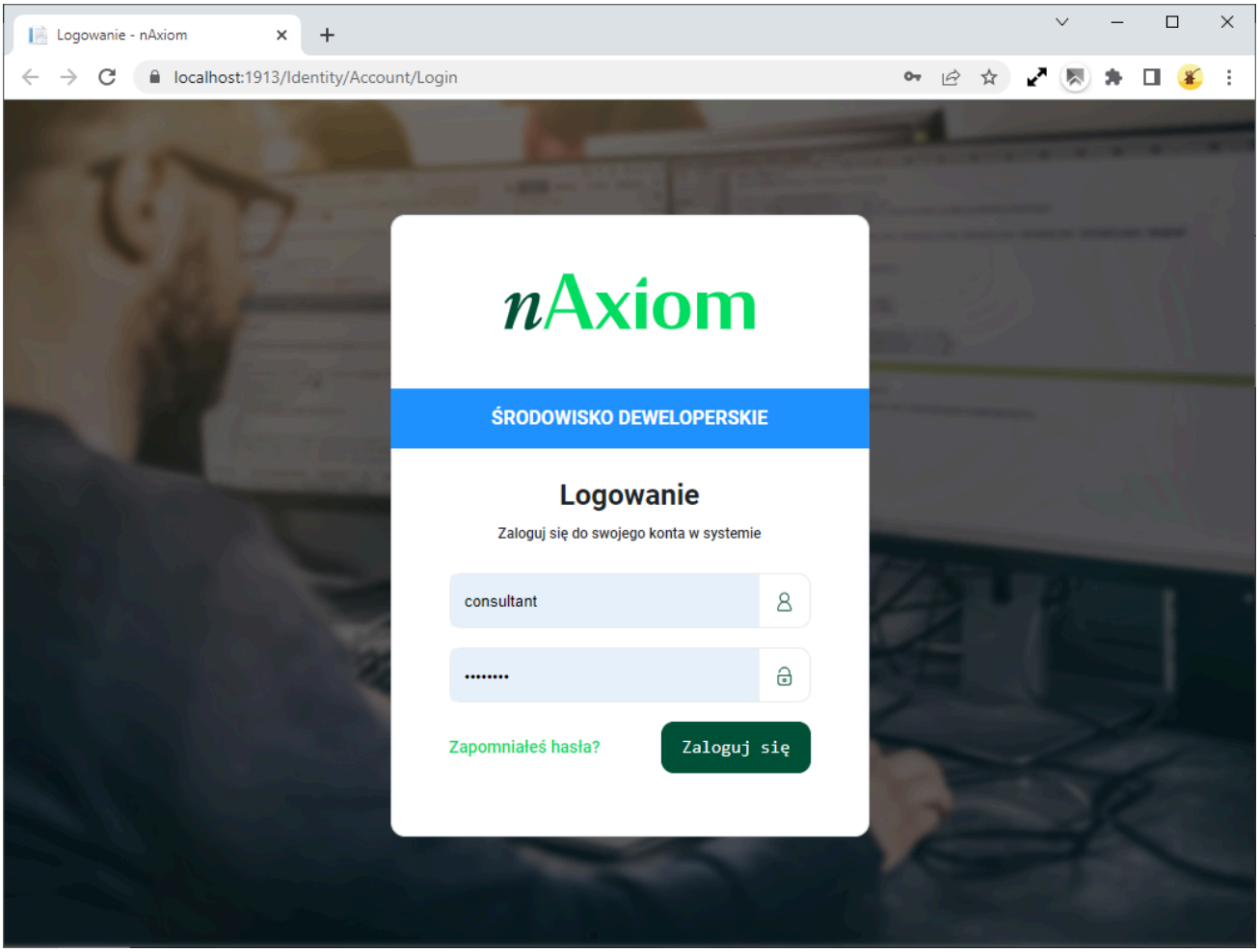
`uproszczony` mają po zalogowaniu dostęp do *FrontSPA*, gdzie mogą korzystać z udostępnionych im aplikacji. Użytkownicy z rolą `Konsultant` mają dostęp do aplikacji *AdminSPA*, w której mogą budować aplikacje biznesowe.

Proces początkowego konfigurowania środowiska oraz tworzenia prostej aplikacji opisano szczegółowo w dokumencie **Pierwsze kroki**.

Konsultanci tworzący aplikacje biznesowe powinni dobrze znać język SQL, do bardziej zaawansowanych rozwiązań przydatna będzie także znajomość języków JavaScript i C# oraz znajomość zasad korzystania z interfejsów API.

2.2. LOGOWANIE: KLIENT, KLIENT UPROSZCZONY

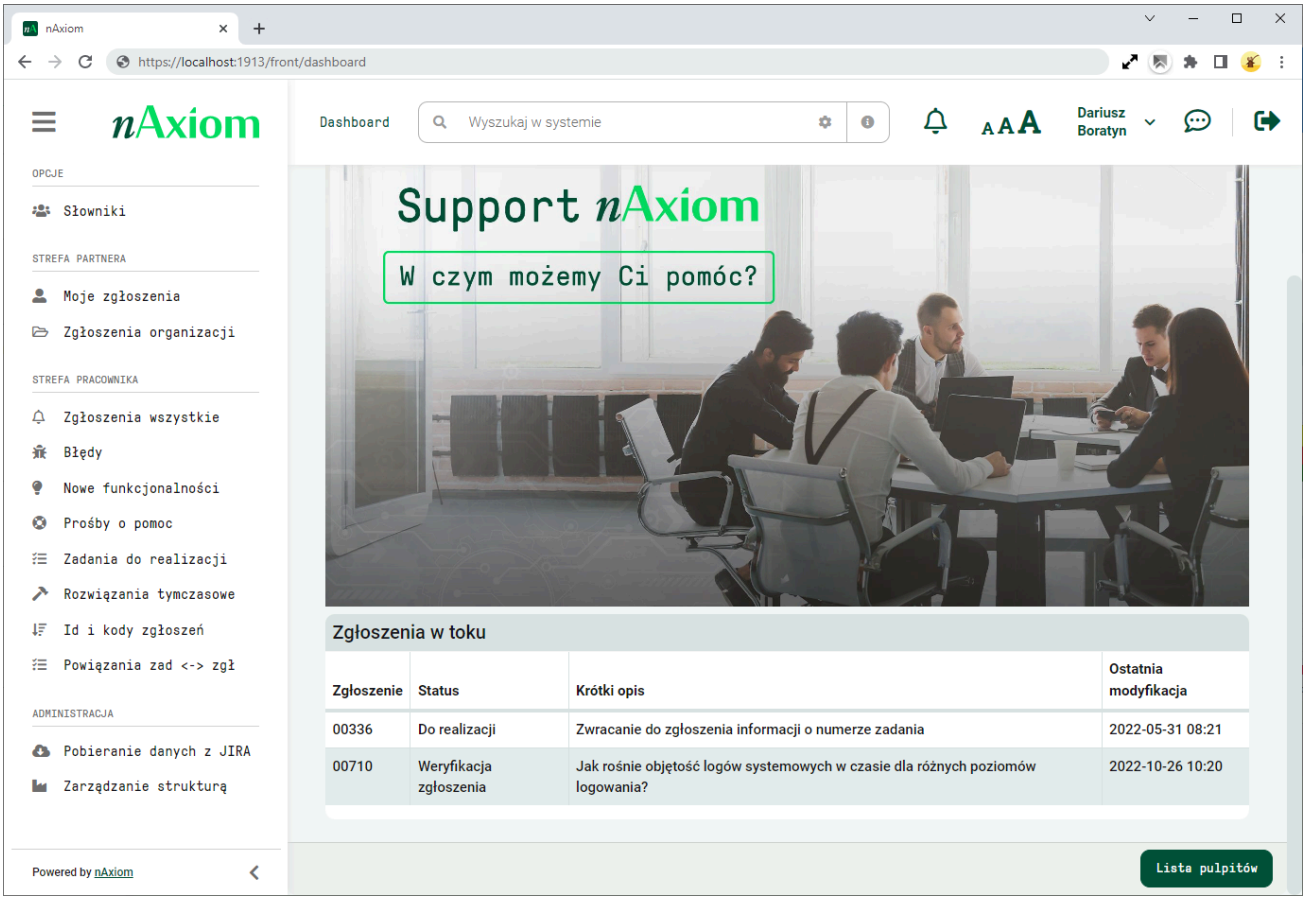
W celu zalogowania do nAxiom należy w pasku adresu przeglądarki wpisać adres platformy nAxiom w danej organizacji określony podczas instalacji platformy za pomocą parametrów *Host*, *Adres IP* i *Port*, patrz Przewodnik instalacji. Następnie, w wyświetlonym oknie logowania należy wpisać dane logowania. W zależności od roli systemowej przypisanej logującemu się użytkownikowi, po zalogowaniu zostanie wyświetlona strona aplikacji *FrontSPA* (role *Klient* i *Klient uproszczony*) lub *AdminSPA* (rola *Konsultant*).



Strona logowania do nAxiom (na ilustracjach w tym podręczniku jest zwykle widoczny adres

`https://localhost:nr_portu`)

Na ilustracji poniżej przedstawiono przykładową witrynę nAxiom służącą do obsługi zgłoszeń pomocy technicznej. W centralnej części (`Pulpit`/`Dashboard`) są wyświetlane tzw. *widżety*, to jest elementy interfejsu, które mogą prezentować bieżące dane, wykresy, wskaźniki itp. W menu z lewej strony są dostępne polecenia wyświetlające listy i formularze.



Portal nAxiom

2.3. LOGOWANIE: KONSULTANT

Podsumowanie:
Użytkownik z rolą *konsultant* jest po zalogowaniu kierowany do aplikacji *AdminSPA* (*https://url_witryny_naxiom/admin/*), która służy do budowania aplikacji biznesowych.

WSKAZÓWKA
Domyślnie w nAxiom jest dostępny jeden profil konsultanta. Aby utworzyć kolejne, należy przypisać użytkownikowi rolę systemową *Konsultant* oraz nadać mu uprawnienia PBA do konfiguracji systemu i budowania aplikacji. Dodatkowo, o ile to konieczne, inny konsultant z uprawnieniami PBA do definicji aplikacji musi nadać mu uprawnienia do poszczególnych aplikacji w AdminSPA. Szczegółowy opis tej procedury znajduje się w rozdziale Tworzenie nowego konsultanta .

Po pomyślnym zalogowaniu w przeglądarce wyświetlana jest strona główna aplikacji *AdminSPA* (*back-office*) witryny nAxiom, na której można wyróżnić następujące obszary:

- Nagłówek: zawiera następujące elementy (od lewej):
  - logo nAxiom: kliknięcie powoduje przeładowanie strony
  - oznaczenie rodzaju środowiska: wraz z kolorem paska nagłówka informuje o rodzaju bieżącego środowiska; może to być środowisko deweloperskie/niebieski, testowe/żółty lub produkcyjne (brak opisu tekstowego, kolor biały)
  - lista rozwijana wyboru aplikacji: pozwala ograniczyć wybór obiektów tylko do tych, które przypisano wybranej aplikacji biznesowej
  - lista rozwijana wyboru trybu pracy: pozwala przełączać wartości zmiennych środowiskowych na wartości zdefiniowane dla innego środowiska niż bieżące, np. w środowisku deweloperskim użyć wartości zmiennych dla środowiska testowego lub odwrotnie
  - imię i nazwisko zalogowanego użytkownika: kliknięcie rozwija menu z poleceniem `Wyloguj` oraz numerem bieżącej wersji nAxiom.
- Pasek skrótów zawiera (poniżej nagłówka) zawiera (lewej):
  - ścieżkę nawigacji do bieżącego okna,
  - przycisk wyświetla okno, w którym można testowo uruchomić dowolne zapytanie SQL,
  - przycisk wyświetla okno logów (`Administracja` > `Logi środowiska i bieżącego tenanta`),
  - przycisk odświeża zawartość bieżącego okna.
- Menu główne: lista poleceń podzielona na osiem sekcji:
  - `ULUBIONE`
  - `DANE`
  - `PROCESY BIZNESOWE`
  - `INTERFEJS UŻYTKOWNIKA`
  - `NARZEDZIA`
  - `INTEGRACJE`
  - `ADMINISTRACJA`
  - `POMOC` Menu główne można zwinąć, klikając ikonę `<` u dołu menu.
- Kokpit: obszar roboczy; po wybraniu polecenia menu w kokpicie są wyświetlane strony, na których tworzy się obiekty aplikacji.

INFO
Aplikacja *AdminSPA* jest dostępna także w języku angielskim. Wyświetlana wersja językowa zależy od języka ustawionego w profilu zalogowanego użytkownika.

Powiązane tematy:

-
- Zmienne środowiskowe
- Wprowadzenie
- Profile użytkowników

3. DANE

Aplikacje tworzone w nAxiom służą do przetwarzania danych biznesowych przechowywanych w tabelach w systemowej bazie danych oraz w innych źródłach danych. Do definiowania źródeł danych dla aplikacji nAxiom służą polecenia `Kreator modeli danych`, `Źródła danych` i `Słowniki` w sekcji `DANE` w menu głównym *AdminSPA*. Dodatkowo w tej sekcji znalazły się polecenie `Repozytoria załączników` i `Kategorie załączników`. Polecenia tej sekcji opisano w dalszej części tego rozdziału.

3.1. KREATOR MODELI DANYCH

Podsumowanie:
Modele danych to w nAxiom tabele w bazie danych SQL Server wskazywanej podczas instalacji danej witryny nAxiom. W tych tabelach przechowywane są dane biznesowe przetwarzane w aplikacjach biznesowych nAxiom.

☛ *AdminSPA* > `DANE` > `Kreator modeli danych`

Podstawową funkcją kreatora jest definiowanie tabel na potrzeby budowanych aplikacji. Ponadto można w nim edytować definicje tabel, tworzyć funkcje tabelaryczne oraz definiować powiązania systemowych funkcji tabelarycznych i procedur składowanych z modułami aplikacji. Bardzo ważną funkcją dostępną z poziomu Kreatora jest **generowanie elementów aplikacji**. Proces budowania aplikacji w nAxiom zaczyna się zawsze od zdefiniowania standardowego modelu danych, a następnie wygenerowania na jego podstawie elementów aplikacji, które są później modyfikowane.

Dodatkowo w kreatorze są dostępne funkcje pomocnicze, takie jak aktualizacja funkcji tabelarycznych generowanych automatycznie dla widoków list, wykonanie zapytania SQL, czy podgląd rekordów w tabelach modelu danych, a także możliwość edycji tabel systemowych służących do synchronizacji użytkowników i jednostek organizacyjnych z serwerem usług katalogowych.

Powiązane tematy:

- Tabela dokumentu biznesowego — kolumny systemowe
- PROCESY BIZNESOWE

3.1.1. Okno Kreator modeli danych

Okno `Kreator modeli danych` dzieli się na dwa panele: „drzewko" modeli danych z lewej i panel edycji modeli danych z prawej. Panel z lewej strony zawiera domyślnie pięć pozycji: `Modele danych`, `Funkcje tabelaryczne`, `Funkcje skalarne`, `Procedury składowane` i `Widoki`. Gałąź `Modele danych` może zawierać pozycje podrzędne: `Standardowe`, `Uproszczone`, `Tabele pomocnicze` i `Tabele systemowe`. Rozwinięcie dowolnej pozycji (kliknięciem symbolu trójkąta z lewej strony nagłówka) spowoduje wyświetlenie listy tabel zdefiniowanych przez użytkownika. Kliknięcie trójkąta z lewej strony nazwy tabeli spowoduje rozwinięcie listy kolumn i zależności tabeli. W górnej części panelu znajduje się pole wyszukiwania, które pozwala znaleźć tabelę po wpisaniu fragmentu jej nazwy.

UWAGA

Pozycja `Tabele systemowe` może zawierać tylko dwie tabele systemowe: *core.SyncOUInstances* i *core.SyncUserProfiles*, których edycja jest wymagana w przypadku synchronizacji struktury organizacyjnej i listy użytkowników z systemem zewnętrznym. Aby wyświetlić te tabele w kreatorze, należy ustawić wartość *true* dla parametru *CoreTablesModificationEnabled* w ustawieniach tenanta na karcie `Configuration` w aplikacji *TenantsAdminSPA*, a następnie ponownie uruchomić serwis *api*. Więcej informacji zawiera artykuł **Obsługa wielu tenantów w nAxiom**.

W oknie `Kreator modeli danych` możesz:

- dodać model danych; kliknij przycisk `Nowy model danych` .
- zaktualizować funkcje tabelaryczne generowane na podstawie tabel modeli standardowych jako źródła danych dla widoków list; kliknij przycisk `Aktualizacja widoków`. Na wyświetlonej liście zaznacz pola wyboru obok tabel i kliknij przycisk `Aktualizuj zaznaczone`. Ta funkcja jest przeznaczona do zbiorowego odświeżenia funkcji tabelarycznych, np. z powodu zmian w nowej wersji nAxiom,

które wymagałyby takiego odświeżenia.

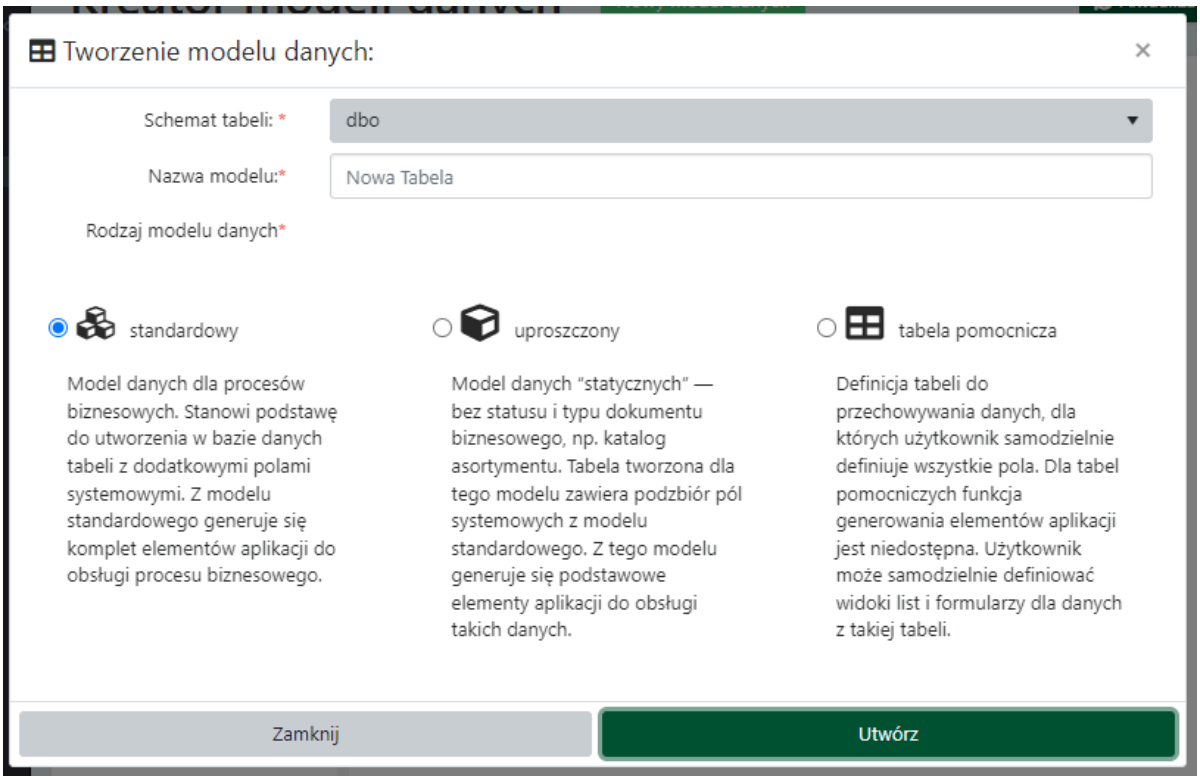
- wykonać SQL; kliknij przycisk `Wykonaj SQL` z prawej strony u góry, w wyświetlonym oknie wpisz zapytanie i kliknij przycisk `Wykonaj zapytanie`.

> W zapytaniach uruchamianych w tym oknie obsługiwana jest składnia smart numbers; zmienne smart numbers w klauzulach *SELECT* i *WHERE* należy ująć w apostrofy, a aby przekazać nazwę tabeli w klauzuli *FROM*, należy skorzystać z *AppInlineVariables*.

- edytować model danych; dwukrotnie kliknij tabelę na panelu z lewej strony lub kliknij tabelę i kliknij ikonę edycji z prawej strony nazwy; w kreatorze można otworzyć kilka definicji tabel w osobnych kartach, patrz opis ; na karcie modelu danych wyświetlane są przyciski, które pozwalają:
    - wygenerować elementy aplikacji dla bieżącego modelu (standardowego lub uproszczonego); kliknij przycisk `Generuj elementy aplikacji`; patrz opis .
    - zmienić nazwę bieżącego modelu; kliknij `Zmień nazwę modelu`.
    - usunąć bieżący model; kliknij `Usuń model`.
    - wyświetlić próbkę rekordów z bieżącego modelu; kliknij `Podgląd zapisanych rekordów`.
- dodać funkcję tabelaryczną; kliknij prawym przyciskiem myszy pozycję `Funkcje tabelaryczne` i wybierz polecenie `Dodaj funkcję`; patrz opis .
- powiązać zdefiniowane w bazie danych funkcje tabelaryczne, funkcje skalarne, procedury składowane i widoki z modułami aplikacji na potrzeby eksportu aplikacji. W tym celu kliknij wybrany obiekt prawym przyciskiem myszy i wybierz polecenie `Przypisz moduł` z menu kontekstowego. Następnie w wyświetlonym oknie zaznacz moduły aplikacji, z którymi chcesz powiązać dany widok lub procedurę.

3.1.2. Tworzenie nowego modelu danych

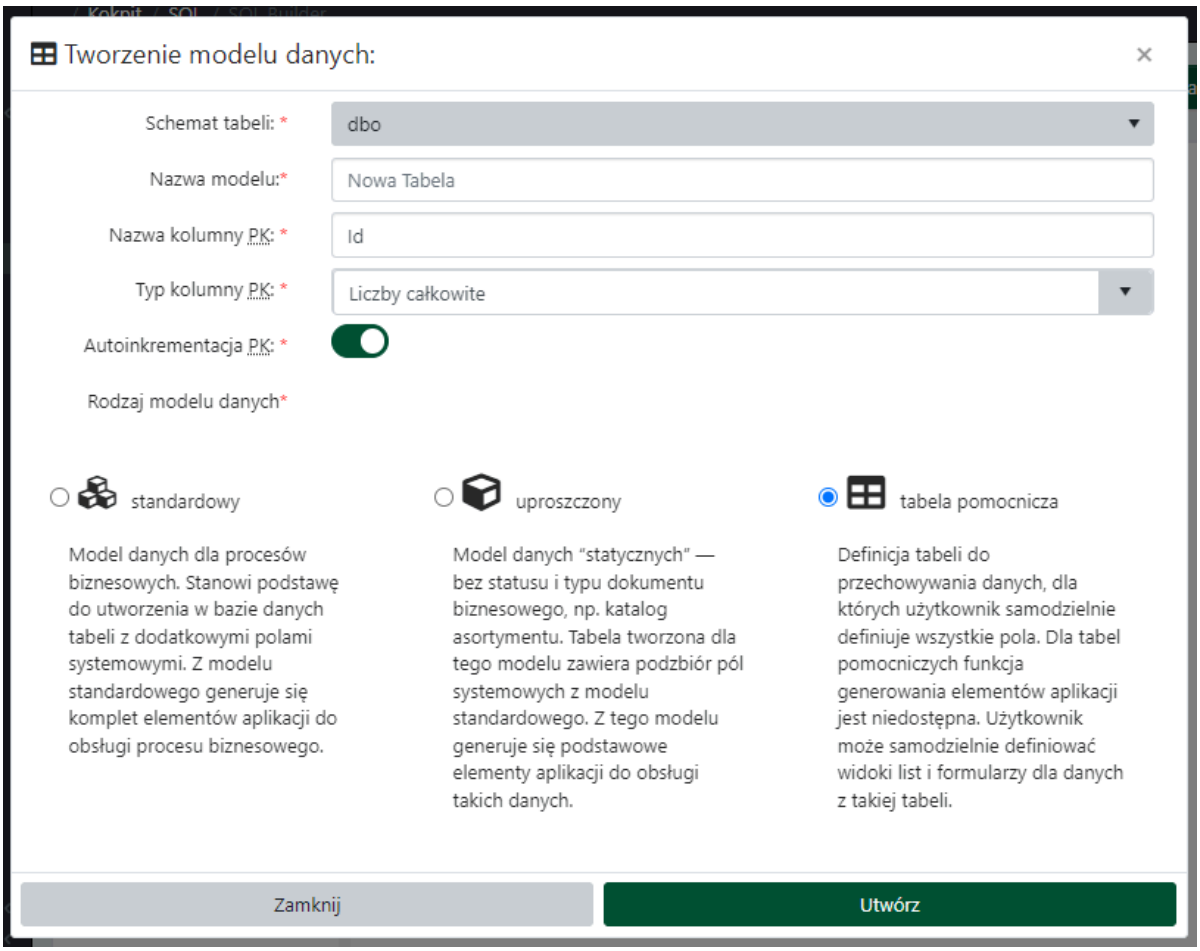
1. W nagłówku okna `Kreator modeli danych` kliknij przycisk `Nowy model danych`. Zostanie wyświetlone okno dialogowe `Tworzenie modelu danych`.



Tworzenie modelu danych

W tym oknie określ następujące ustawienia:

- `Schemat tabeli`: wybierz z listy schemat, do którego zostanie przypisana tabela; wartość domyślna to *dbo*.
- `Nazwa modelu`: wpisz nazwę tabeli; nazwa tabeli nie może zawierać spacji.
- `Rodzaj modelu danych`: wybierz rodzaj tworzonej tabeli; dostępne wartości:
  - `standardowy`: modele przechowywane w tabelach, które oprócz pól zdefiniowanych przez użytkownika mają także zestaw pól systemowych umożliwiających automatyczne generowanie elementów aplikacji, takich jak definicja dokumentu biznesowego, typ dokumentu biznesowego, licznik autonumeracji, formularz, widok listy i zalążek diagramu procesu biznesowego; patrz także PROCESY BIZNESOWE
  - `uproszczony`: model korzystający z tabeli z ograniczonym zestawem pól systemowych, dla których możliwe jest automatyczne wygenerowanie definicji dokumentu biznesowego, licznika autonumeracji, formularza i widoku listy
  - `tabela pomocnicza`: model korzystający ze zwykłej tabeli w bazie danych zawierającej pola zdefiniowane przez użytkownika; zgodnie z nazwą tabele te pełnią rolę pomocniczych kontenerów danych.

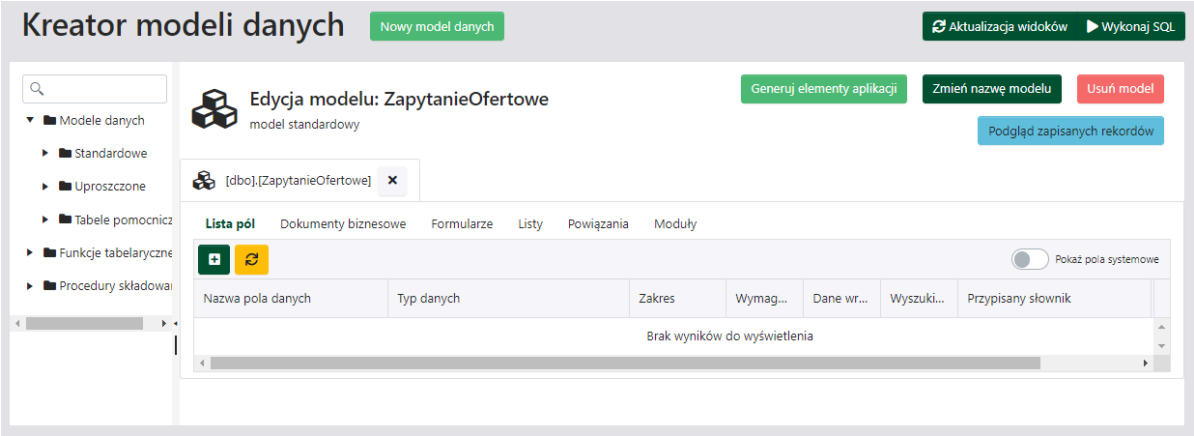


Tworzenie modelu danych, tabela pomocnicza

Dla tabeli pomocniczej należy dodatkowo wypełnić pola:

- `Nazwa kolumny PK`: nazwa kolumny klucza głównego; ta kolumna zostanie utworzona automatycznie; nazwa domyślna to `Id`.
- `Typ kolumny PK`: typ danych kolumny klucza głównego; dostępne wartości to *Liczby całkowite* (domyślnie), *Tekst nvarchar*, *Data i czas*, *Tekst varchar*, *Unikalny identyfikator*.
- `Autoinkrementacja PK`: domyślnie włączony przełącznik dostępny dla kolumny klucza głównego typu *liczba całkowita*, powoduje automatyczne generowanie wartości klucza głównego.

2. Kliknij przycisk `Utwórz`.
   Nowa tabela zostanie wyświetlona w centralnym panelu okna `Kreator modeli danych`. Następny krok to zdefiniowanie pól (kolumn) tabeli.



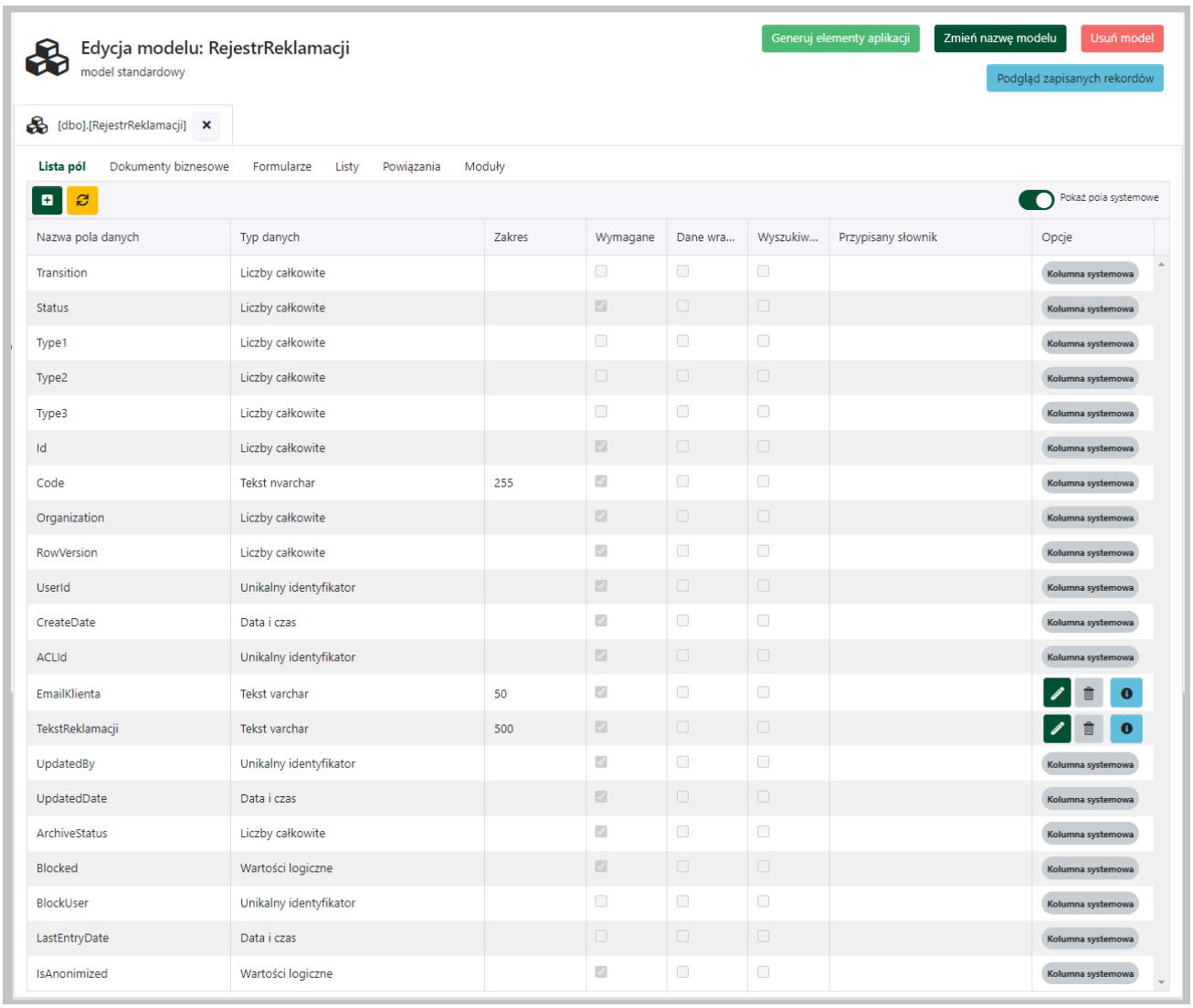
Nowy model danych

3.1.3. Definiowanie pól modelu danych

1. W oknie `Kreator modeli danych` wyświetl definicję modelu i kliknij zielony przycisk z symbolem `+`.
   Zostanie wyświetlone okno dialogowe `Nowe pole modelu danych`.
2. Wypełnij pola zgodnie z opisem w kolejnej sekcji i kliknij przycisk `Zapisz` u dołu.
   Pole zostanie dodane do tabeli.
3. Ponownie kliknij przycisk z symbolem `+` i dodaj kolejne pole, a potem następne.

Zdefiniowane pola zostaną dodane do tabeli i będą widoczne na liście z lewej strony po kliknięciu symbolu `+` z lewej strony nazwy tabeli.

> **WSKAZÓWKA**
> oprócz pól zdefiniowanych przez użytkownika, tabela ma także pola systemowe, aby je wyświetlić, należy włączyć przełącznik `Pokaż pola systemowe` z prawej strony nad listą pól tabeli w widoku projektowania; patrz sekcja Tabela dokumentu biznesowego — kolumny systemowe .



Pola systemowe, na panelu z lewej strony rozwinięta lista pól i zależności tabeli

3.1.4. Nowe pole modelu danych

> **WSKAZÓWKA**
> To okno dialogowe jest wyświetlane w Kreatorze modelu danych, w Designerze formularzy (po kliknięciu przycisku z symbolem + z lewej strony właściwości `Pole bazy danych`) oraz w oknie edytora SQL po wybraniu z menu kontekstowego nazwy tabeli polecenia `Dodaj kolumnę`.

Aby dodać nowe pole tabeli, wypełnij następujące pola

- `Tabela`: pole tylko do odczytu z nazwą tabeli, w której dodawane jest nowe pole; ta informacja jest przydatna w designerze formularzy.
- `Nazwa pola danych`: wpisz nazwę pola (kolumny); nazwa pola musi zaczynać się od wielkiej litery i może składać się tylko ze znaków alfanumerycznych oraz ze znaków podkreślenia.

- `Etykieta`: wpisz treść etykiety (maks. 255 dowolnych znaków) dla danego pola modelu danych; wartość atrybutu jest zapisywana w tabeli *core.SqlBuilderColumnsInfo*; wartość tego atrybutu może być tłumaczona i jest wykorzystywana w komunikatach o błędach wyświetlanych dla reguł walidacji reguł walidacji .

- `Typ danych`: wybierz typ danych które będą zapisywane w tym polu; dostępne wartości:
    - `Liczby całkowite` (*int*): liczba całkowita z zakresu od -9 223 372 036 854 775 808 do 9 223 372 036 854 775 807.
    - `Liczby zmiennoprzecinkowe` (*float*): liczba zmiennoprzecinkowa z zakresu od -1,79E+38 do 3,40E+38
    - `Liczby dziesiętne` (*decimal*): liczba dziesiętna o stałej precyzji z zakresu od $-10^{38}+1$ do $10^{38}-1$; parametr `Zakres` jest określany za pomocą dwu liczb: *m* określa maksymalną liczbę wszystkich cyfr w wartości, a *n* liczbę cyfr na prawo od separatora dziesiętnego; *n* może przyjmować wartości od *0* (wartość domyślna) do *m*.
    - `Tekst nvarchar(n)` (*nvarchar(n)*): łańcuch znaków Unicode o zmiennej długości; parametr `Zakres` określa maksymalną liczbę znaków, wartości zawierające więcej znaków nie zostaną zapisane w bazie danych.
    - `Wartości logiczne` (*bit*): liczba całkowita, która może przyjmować tylko wartości *0*, *1* i *NULL*; używana do reprezentacji danych logicznych.
    - `Data` (*date*): wartość daty z zakresu od 1 stycznia 0001 do 31 grudnia 9999.
    - `Data i czas` (*datetime*): wartość daty i godziny z zakresu od 1 stycznia 1753 do 31 grudnia 9999 z dokładnością do 3,33 ms.
    - `Czas` (*time*): wartość godziny z zakresu od 00:00:00.0000000 do 23:59:59.9999999.
    - `Unikalny identyfikator` (*uniqueidentifier*): globalnie niepowtarzalny identyfikator (GUID); ciąg 32 znaków heksadecymalnych.
    - `Tekst varchar(n)` (*varchar(n)*): łańcuch znaków o zmiennej długości; parametr `Zakres` określa maksymalną liczbę znaków, wartości zawierające więcej znaków nie zostaną zapisane w bazie danych.
- `Zakres`: określ rozmiar pola wybranego typu; ustawienie dostępne dla typów tekstowych i dla liczb dziesiętnych (w tym przypadku określ rozmiar pola i po przecinku precyzję, to jest liczbę miejsc po przecinku).

- `Wartość domyślna`: wartość, która będzie domyślnie wstawiana w tym polu w nowym rekordzie; określanie wartości domyślnej odbywa się tak samo jak w programie MS SQL Server Management Studio lub w języku SQL przy użyciu funkcji *DEFAULT*; patrz dokumentacja Microsoft.

UWAGA

- Wartości typu **tekstowego** oraz **daty i czasu** muszą zostać ujęte w **apostrofy**, np. *'2023-03-08'*, **'Nie dotyczy'* itp.
- W wartościach **liczbowych** jako separator dziesiętny musi zostać użyta **kropka**, np. *123.4*.
- Możliwe jest użycie funkcji systemowych SQL Server, np. *GETDATE()* lub *NEWID()*.
- Po wejściu w edycję pola ze zdefiniowaną wartością domyślną ta wartość będzie wyświetlana w nawiasach. Są one dodawane przez SQL Server podczas zapisu w bazie danych.
  Patrz także dodatkowe uwagi na końcu tej sekcji.

- `Wymagane`: (domyślnie zaznaczone) uniemożliwia zapis rekordu w tabeli, jeśli w danym polu rekordu nie zostanie podana wartość.
- `Dane wrażliwe`: wskazuje, że wartości z danego pola zawierają dane osobowe, które powinny być przetwarzane zgodnie z rozporządzeniem RODO.
- `Dostęp chroniony`: powoduje, że wartość z tego pola jest zwracana tylko wtedy, gdy użytkownik, który próbuje pobrać dane ma uprawnienie ACL R do kontrolki formularza powiązanej z tym polem, lub gdy widoczność kontrolki nie została wyłączona ustawieniami dostępu. Użytkownik musi mieć także dostęp do sekcji formularza, w której znajduje się kontrolka oraz do danej instancji dokumentu.
- `Wyszukiwanie`: dodaje wartości z tego pola do indeksu wyszukiwania; umożliwia wyszukiwanie rekordów tabeli według tych wartości.
- `Przypisany słownik`: dla pola typu *int* to ustawienie pozwala wybrać zdefiniowany słownik; dla pola z przypisanym słownikiem wartości ustawia się, wybierając je z listy.

Atrybut `Wartość domyślna` ma na celu umożliwienie dodania nowego pola z włączoną flagą `Wymagane` do tabel, które już zawierają dane. Po dodaniu pola w istniejących rekordach zostanie ono wypełnione wartością domyślną. To ustawienie nie wpłynie na wartości domyślne definiowane w kontrolkach na formularzu, ani w kolumnach listy (karta `Edycja inline`). Dla nowych instancji dokumentów będą używane wartości domyślne zdefiniowane w interfejsie użytkownika.

UWAGA
W przypadku definiowania wartości domyślnej dla pola tabeli należy skonfigurować również wartość domyślną w odpowiadającej mu kontrolce formularza (kolumnie listy z włączonym dodawaniem rekordów inline) lub ustawić dla kontrolki atrybut `Wymagane`, aby nie dopuścić do sytuacji, kiedy po zapisie w formularzu rekordu z pustymi polami w tych polach pojawią się wartości.

Powiązane tematy:

- Słowniki

3.1.5. Tabela dokumentu biznesowego — kolumny systemowe

Podsumowanie:
Podczas definiowania modelu standardowego w kreatorze modeli danych (patrz Kreator modeli danych) tworzona jest w bazie danych tabela z kolumnami zdefiniowanymi przez użytkownika oraz zestawem kolumn systemowych, w których zapisywane są dodatkowe informacje dotyczące przetwarzania danego rekordu. Poniższa tabela zawiera krótki opis tych kolumn.

Kolumny systemowe w tabeli dokumentu biznesowego

Nazwa	Typ	Atrybuty SQL	Opis
Transition	int	NULL	Identyfikator ostatnio wykonanego przejścia biznesowego (ze statusu X na status Y); identyfikatory przejść są zapisywane w tabeli systemowej *core.BusinessTransitions*
Status	int	NOT NULL	Identyfikator statusu dokumentu biznesowego wg tabeli *core.BusinessStatuses*; patrz także Definicje statusów
Type1/2/3	int	NULL	identyfikator typu biznesowego dokumentu; pozwala definiować podkategorie danego dokumentu biznesowego; typy dokumentów biznesowych są zapisywane w tabeli *core.BusinessTypes*; patrz także Typy dokumentów
Id	int	IDENTITY(1,1) NOT NULL PRIMARY KEY	Identyfikator rekordu w tabeli, kolumna klucza głównego każdej tabeli dokumentu biznesowego
Code	nvarchar(255)	NOT NULL	Numer dokumentu zgodnie ze zdefiniowanym schematem autonumeracji, domyślnie 4-cyfrowy nr kolejny łamany przez rok; patrz także Autonumeracja dokumentów
RowVersion	int	NOT NULL	Numer wersji rekordu, każdy zapis dokumentu zwiększa licznik
UserId	uniqueidentifier	NOT NULL	Identyfikator użytkownika tworzącego dokument z tabeli *core.UserProfiles*
CreateDate	datetime	NOT NULL	Data utworzenia rekordu/dokumentu

Kolumny systemowe w tabeli dokumentu biznesowego

Nazwa	Typ	Atrybuty SQL	Opis
UpdatedBy	uniqueidentifier	NOT NULL	Identyfikator użytkownika, który zaktualizował dokument jako ostatni, z tabeli *core.UserProfiles*
UpdatedDate	datetime	NOT NULL	Data ostatniej aktualizacji dokumentu
ACLId	uniqueidentifier	NOT NULL	Identyfikator używany po włączeniu obsługi uprawnień na dokumencie; patrz także Uprawnienia ACL do dokumentów
Blocked	bit	NOT NULL	Flaga informująca, że dany rekord jest zablokowany przez innego użytkownika; patrz także Definicja dokumentu, Blokady
BlockUser	uniqueidentifier	NULL	Identyfikator użytkownika blokującego dokument
LastEntryDate	datetime	NULL	Moment zablokowania dokumentu, od którego jest liczony czas blokady
ArchiveStatus	int	NOT NULL	Status archiwizacji dokumentu
IsAnonimized	bit	NOT NULL	Flaga określająca, czy dla jakichkolwiek danych w tym rekordzie stosowano akcję anonimizacji danych; dotyczy tylko kolumn z zaznaczonym polem wyboru `Dane wrażliwe`; jeśli dane były anonimizowane, flaga ma wartość *1*; patrz także Sekcja danych RODO

WSKAZÓWKA

Tabela w uproszczonym modelu danych, tzw. *tabela katalogu*, różni się od tabeli w modelu standardowym tym, że nie ma kolumn *Transition*, *Status* i *Type 1/2/3*.

3.1.6. Edytowanie modelu danych

Podsumowanie:
Aby przejść do edycji modelu danych, należy wyświetlić jego definicję w kreatorze modeli danych, np. przez dwukrotne kliknięcie jego nazwy.

W TYM TEMACIE

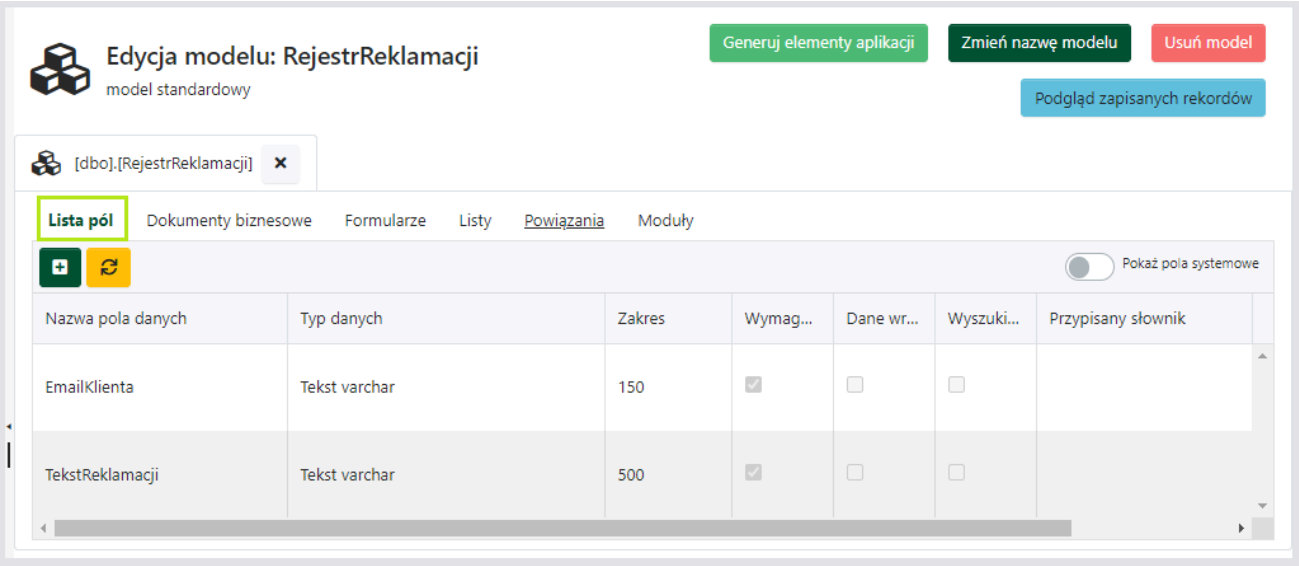
- Karta Lista pól
- Karta Dokumenty biznesowe
- Karta Formularze
- Karta Listy
- Karta Powiązania
- Karta Moduły

Centralny panel w oknie `Kreator modeli danych` to panel edycji modeli danych/tabel. Edytowane tabele są wyświetlane jako karty tego panelu, a w nagłówku wyświetlana jest nazwa i rodzaj aktywnego modelu oraz przyciski:

- `Generuj elementy aplikacji`: przycisk dostępny tylko dla modeli standardowych i uproszczonych; kliknij, aby uruchomić mechanizm automatycznego generowania elementów interfejsu użytkownika (formularz, widok listy) i logiki biznesowej (definicja dokumentu biznesowego, diagram procesu) tworzonej aplikacji; patrz Kreator modeli danych, generowanie elementów aplikacji
- `Zmień nazwę modelu`: kliknij, aby zmienić nazwę tabeli; zostanie wyświetlone okno dialogowe, w którym można będzie podać nową nazwę tabeli.
- `Usuń model`: kliknij, aby usunąć tabelę modelu danych; usunięcie tabeli wymaga dwukrotnego potwierdzenia, ponieważ spowoduje usunięcie danych zapisanych w tej tabeli; nie będzie można usunąć modelu, jeśli wygenerowano na jego podstawie elementy aplikacji.
- `Podgląd zapisanych rekordów`: kliknij, aby wyświetlić rekordy danych zapisane w tabeli (maks. 1000) oraz zapytanie SQL użyte do zwrócenia tych rekordów; w wyświetlonym oknie można modyfikować zapytanie, sprawdzać jego poprawność i wykonywać w celu zwrócenia zestawu rekordów.

Karta tabeli modelu danych dzieli się na kilka kart podrzędnych, na których znajdują się różne informacje dotyczące edytowanego modelu danych.

KARTA LISTA PÓL

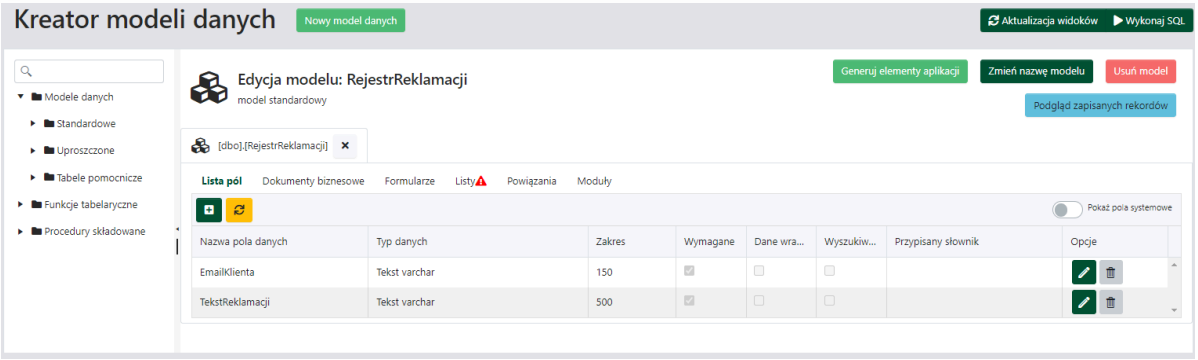


Edycja modelu — Lista pól

Karta `Lista pól` to tabelka, w której wyświetlane są informacje o zdefiniowanych polach (kolumnach) tabeli. Na tej karcie można dodawać, edytować i usuwać pola tabeli. Na karcie znajdują się następujące kontrolki:

- (`Nowa kolumna`): kliknij, aby dodać pole do tabeli. W wyświetlonym oknie dialogowym wypełnij pola zgodnie z opisem w poprzedniej sekcji.
- (`Odśwież`): ikona wyświetlania po zmodyfikowaniu definicji tabeli; kliknij, aby odświeżyć inne obiekty w nAxiom, w których jest wykorzystywana definicja tabeli (np. formularze, raport itp.).

> **WSKAZÓWKA**
> W przypadku niektórych zmian po odświeżeniu modelu, obok nazw kart `Listy` i `Formularze` mogą zostać wyświetlone czerwone trójkąty ostrzegawcze, które wskazują na konieczność odświeżenia definicji danego obiektu w celu dostosowania go do zmian tabeli. Odświeżenia polega na otwarciu definicji i zapisaniu jej.



Sygnalizacja konieczności odświeżenia definicji listy w związku ze zmianami tabeli

- `Pokaż pola systemowe`: (domyślnie wyłączone) kliknij, aby włączyć wyświetlanie na liście pól systemowych, to jest predefiniowanych pól tworzonych automatycznie przez nAxiom w momencie tworzenia tabeli;patrz sekcja Tabela dokumentu biznesowego — kolumny systemowe .

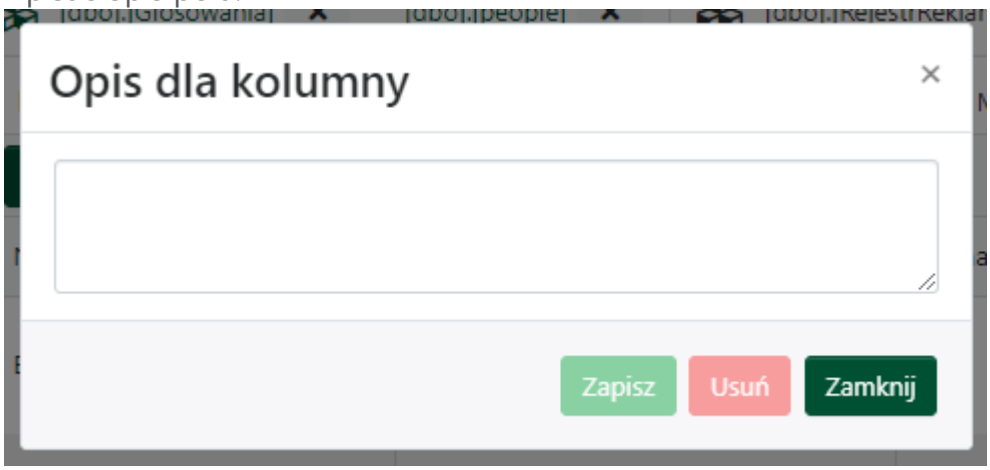
- (`Edycja`): ikona wyświetlana w wierszu definicji pola tabeli; kliknij, aby wyświetlić okno dialogowe `Edycja pola modelu danych`.

  > **UWAGA**
  > Nie można edytować kolumn systemowych.

- (`Usuń`): ikona wyświetlana w wierszu definicji pola tabeli; kliknij, aby usunąć pole z definicji tabeli.

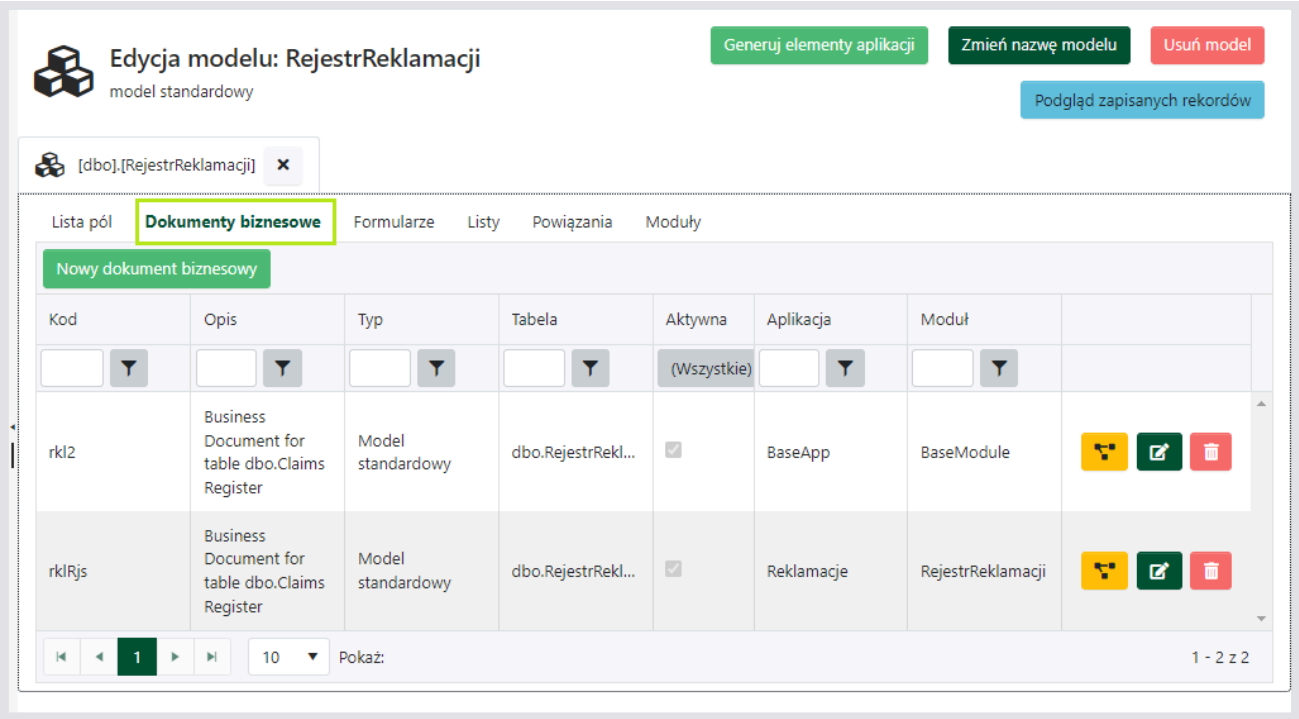
  > **OSTRZEŻENIE**
  > Usunięcie pola spowoduje usunięcie danych zapisanych w tym polu.

- (`Info`): kliknięcie tej ikony powoduje wyświetlenie okna dialogowego, w którym można wpisać opis pola.



Opis kolumny

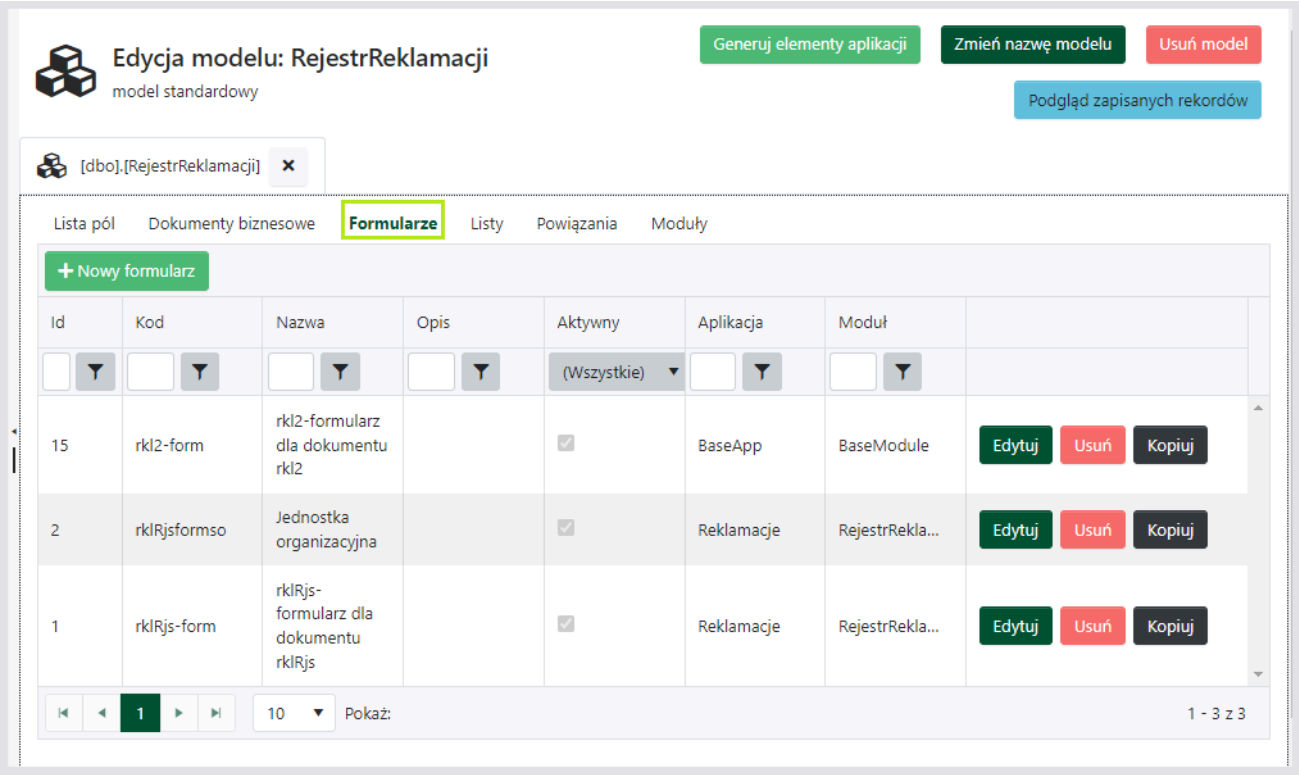
KARTA DOKUMENTY BIZNESOWE



Edycja modelu — Dokumenty biznesowe

Na karcie `Dokumenty biznesowe` jest wyświetlana lista dokumentów biznesowych zdefiniowanych dla tej tabeli. Na tej karcie można dodawać nowe definicje dokumentów biznesowych, edytować je i usuwać. Funkcjonalnie ta karta jest identyczna z oknem wyświetlanym poleceniem `Dokumenty biznesowe` w sekcji `Procesy biznesowe` w menu głównym. Patrz Dokumenty biznesowe (definicje dokumentów) .

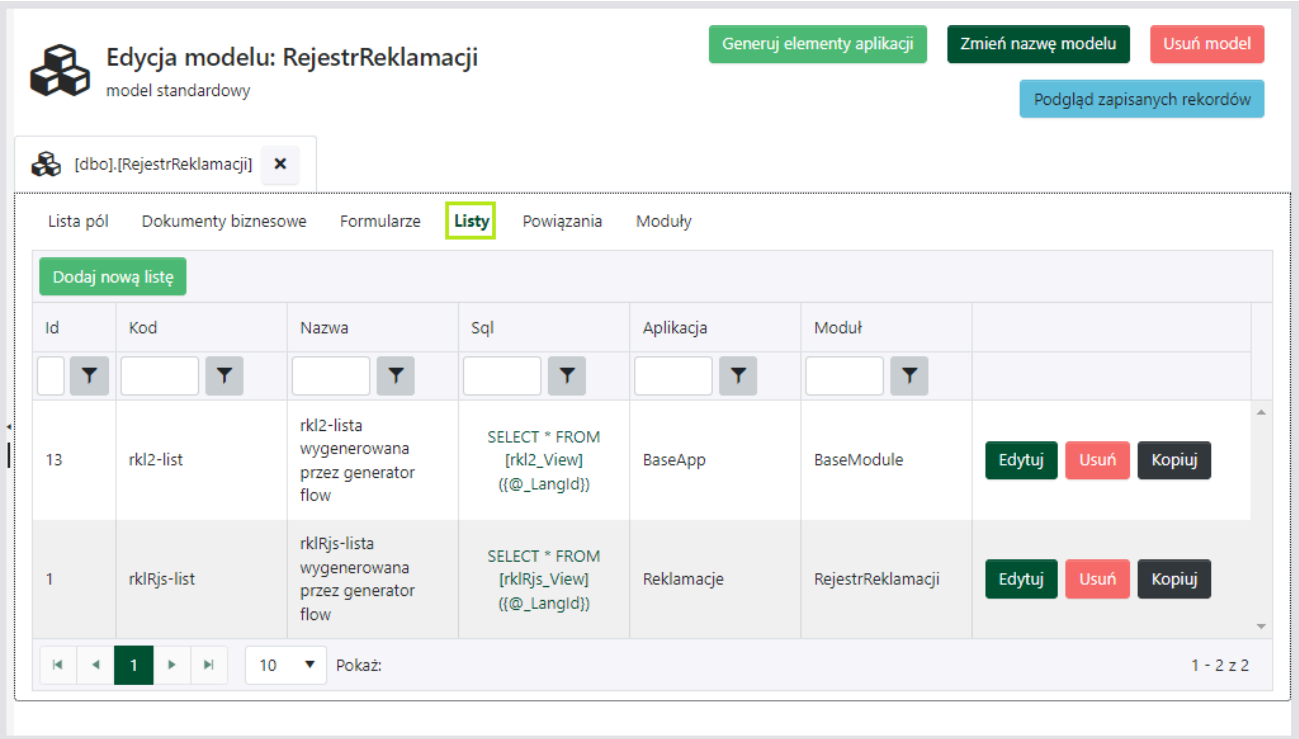
KARTA FORMULARZE



Edycja modelu — Formularze

Na karcie `Formularze` jest wyświetlana lista formularzy zdefiniowanych dla tej tabeli. Na tej karcie można dodawać nowe definicje formularzy, edytować je i usuwać. Funkcjonalnie ta karta jest identyczna z oknem wyświetlanym poleceniem `Formularze` w sekcji `Interfejs użytkownika` w menu głównym. Patrz Formularze .

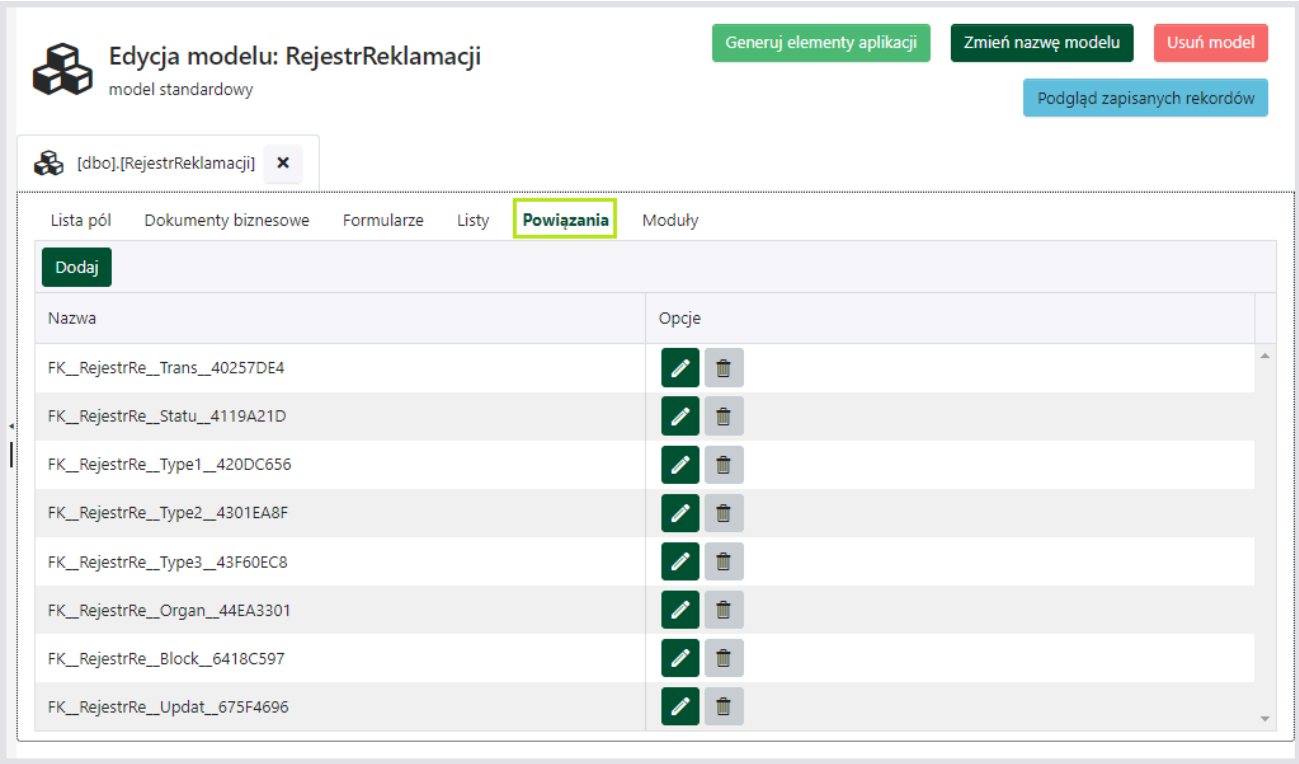
KARTA LISTY



Edycja modelu — Listy

Na karcie `Listy` jest wyświetlana lista widoków list zdefiniowanych dla tej tabeli. Na tej karcie można dodawać nowe definicje widoków list, edytować je i usuwać. Funkcjonalnie ta karta jest identyczna z oknem wyświetlanym poleceniem `Listy` w sekcji `Interfejs użytkownika` w menu głównym. Patrz Listy .

KARTA POWIĄZANIA

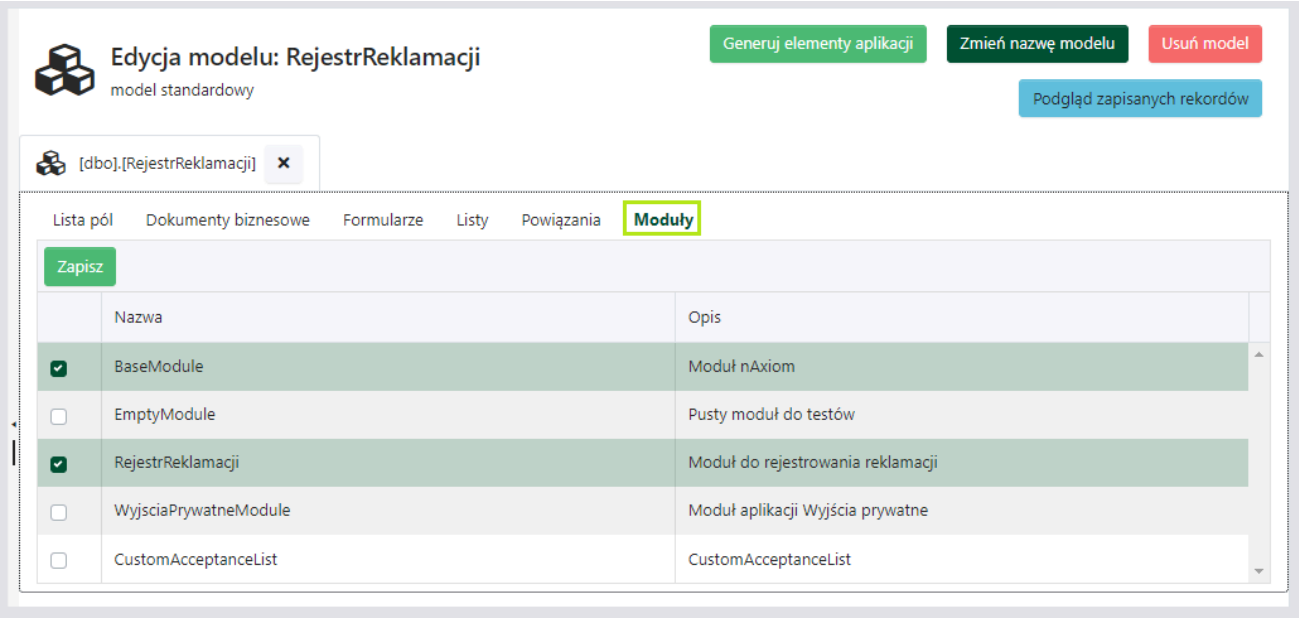


Edycja modelu — Powiązania

Na karcie `Powiązania` jest wyświetlana lista powiązań (relacji) danej tabeli z innymi tabelami. Dla tabel dokumentu biznesowego kilka takich powiązań jest tworzonych automatycznie podczas tworzenia tabeli. Na tej karcie można także definiować własne powiązania, edytować powiązania i usuwać je. Szczegółowy opis zawiera następna sekcja.

- `Dodaj`: kliknij, aby dodać nowe powiązanie tabeli; zostanie wyświetlone okno dialogowe `Edycja relacji`; patrz opis w kolejnej sekcji.
- (`Edycja`): kliknij, aby edytować powiązanie tabeli.
- (`Usuń`): kliknij, aby usunąć powiązanie tabeli.

KARTA MODUŁY



Edycja modelu — Moduły

Na karcie modułów znajduje się lista modułów aplikacji, do których można przypisać daną tabelę. Kliknij pole wyboru obok modułu, do którego chcesz przypisać tabelę i kliknij przycisk `Zapisz`.

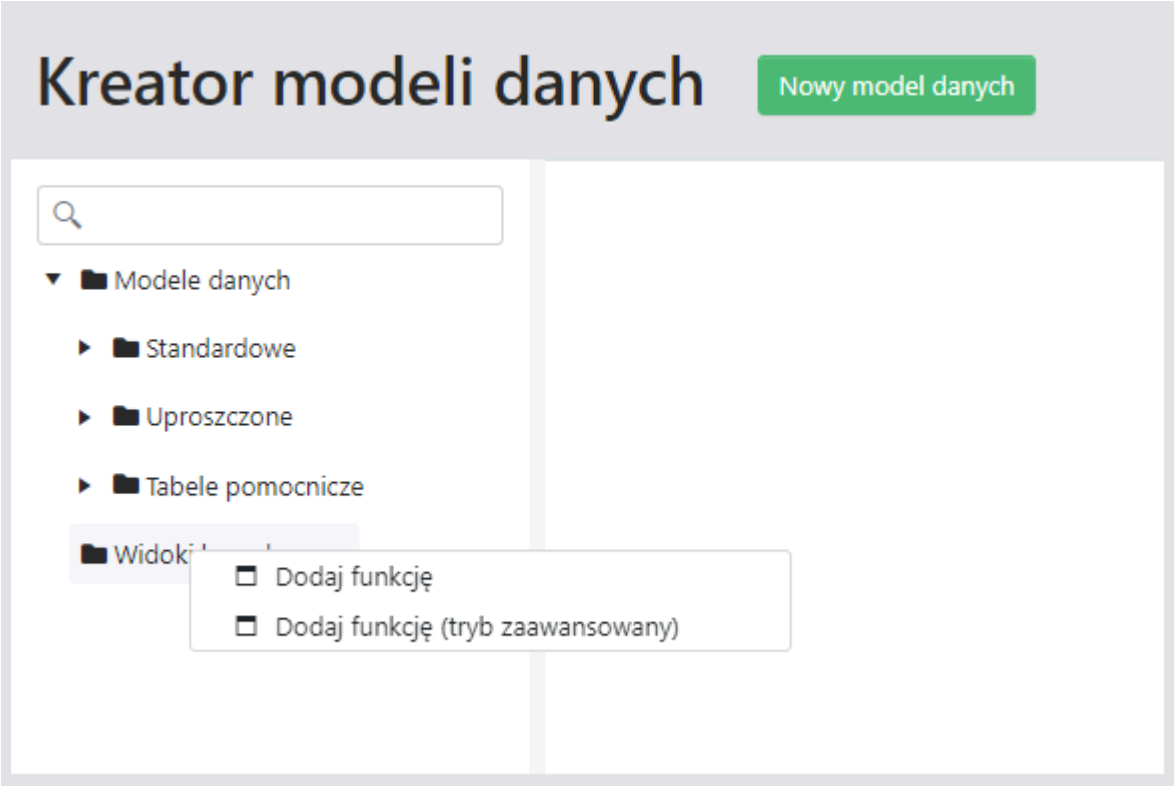
Tabela musi zostać przypisana do modułu, aby można było ją migrować, to jest przenosić między środowiskami nAxiom przy użyciu funkcjonalności *Migrator aplikacji*.

3.1.7. Nowa funkcja tabelaryczna

Podsumowanie:
Funkcja tabelaryczna (ang. *table valued function*) to oparta na zapytaniu SQL funkcja, która zwraca tabelę. Na tabeli zwracanej przez funkcję tabelaryczną można wykonywać takie same operacje, jak na zwykłej tabeli. Funkcji tabelarycznych można używać jako źródeł danych dla różnych obiektów nAxiom.

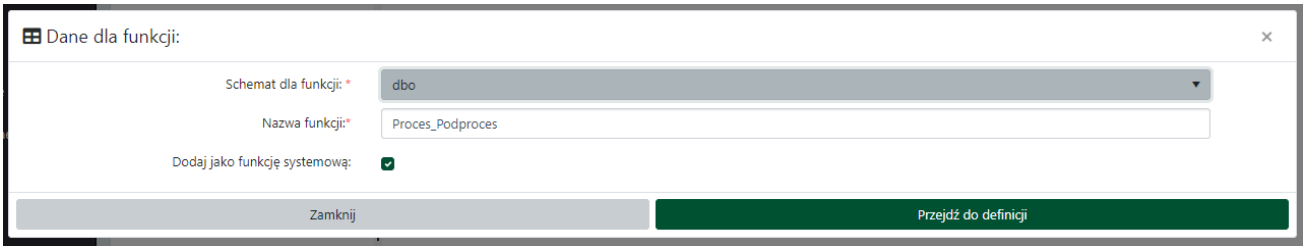
Aby dodać nową funkcję tabelaryczną, wyświetl kreator modeli danych i kliknij prawym przyciskiem myszy pozycję `Widoki bazodanowe` w drzewie modeli danych z lewej strony. Zostanie wyświetlone menu z dwoma pozycjami:

- `Dodaj funkcję`
- `Dodaj funkcję (tryb zaawansowany)`



Nowa funkcja tabelaryczna

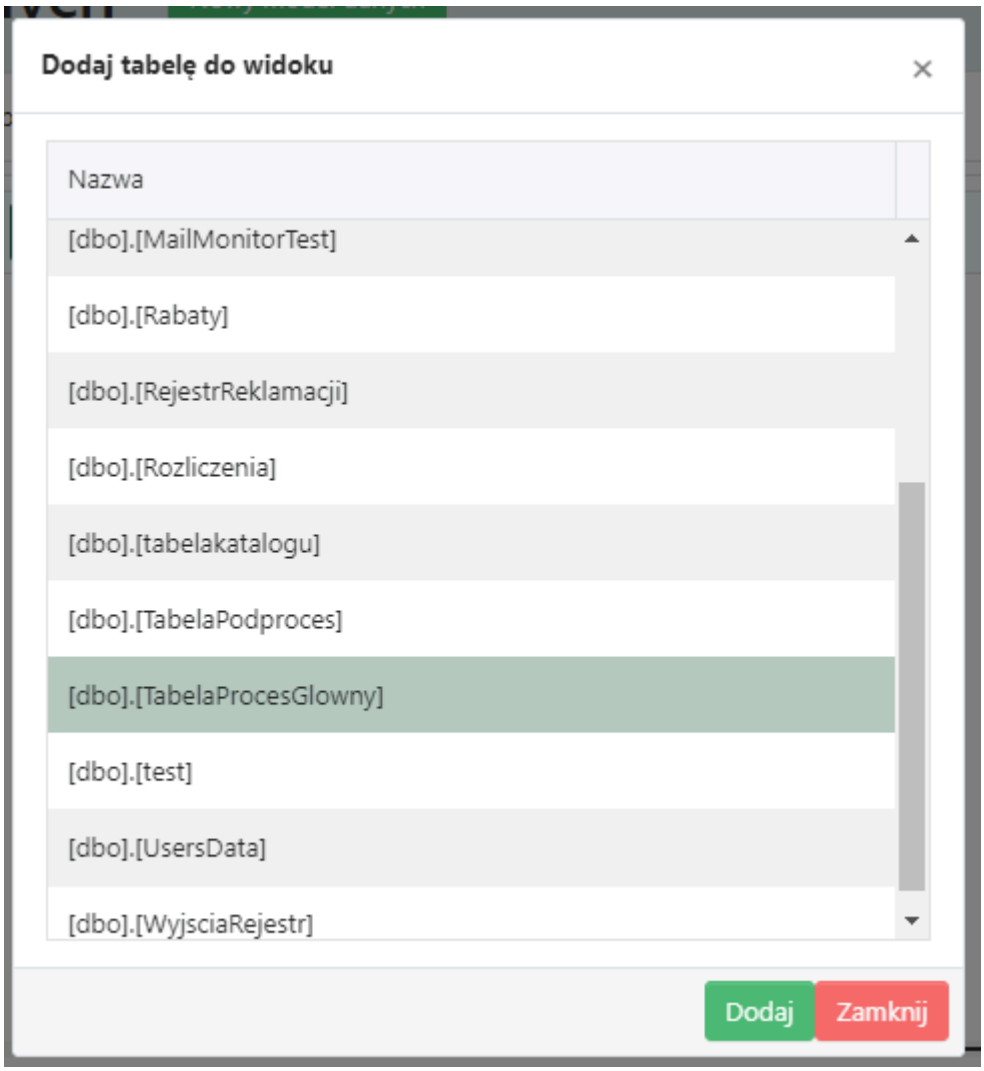
Wybierz pierwsze polecenie. Zostanie wyświetlone okno dialogowe, w którym należy wybrać schemat bazy danych (dla obiektów użytkownika zwykle używa sie schematu *dbo*) i podać nazwę funkcji. Domyślnie zaznaczone pole wyboru `Dodaj jako funkcję systemową` powoduje, że utworzona funkcja będzie widoczna w kreatorze, a ponadto do jej nazwy zostanie dodany przedrostek *_custom_*. W razie anulowania zaznaczenia tego pola funkcja zostanie zdefiniowana, ale nie będzie widoczna w kreatorze. Kliknij przycisk `Przejdź do definicji`.



Dane funkcji

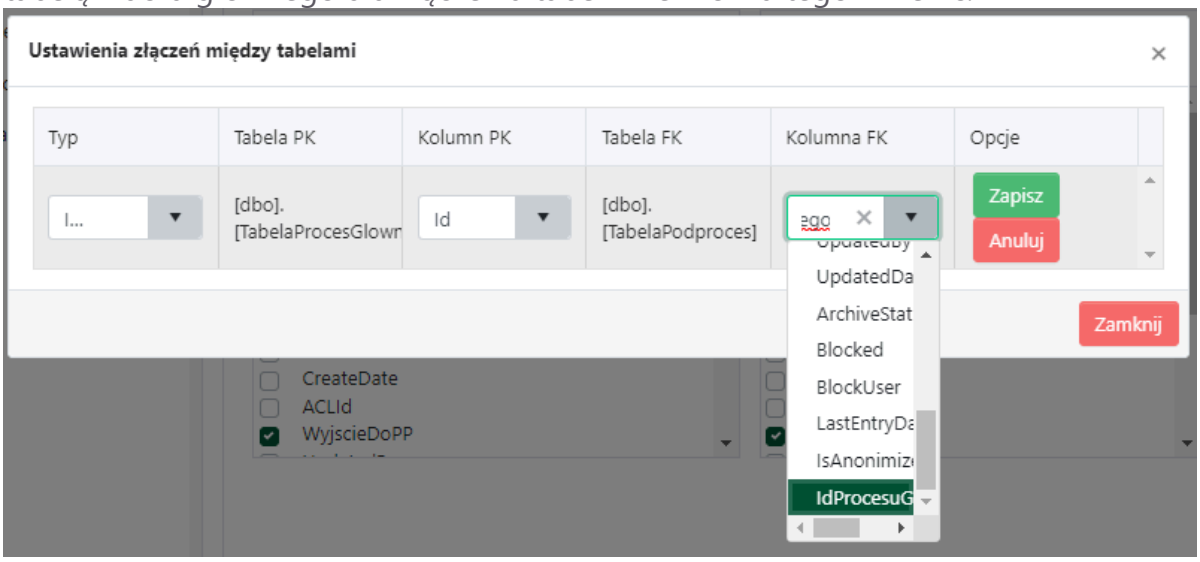
Przejdź do głównego panelu kreatora, aby utworzyć funkcję tabelaryczną.

1. Kliknij przycisk `Dodaj tabelę`: zostanie wyświetlona lista tabel, na podstawie których można utworzyć funkcję tabelaryczną; dostępne są tabele reprezentujące standardowe i uproszczone modele danych. Kliknij kolejno nazwę tabeli i kliknij przycisk `Dodaj`.



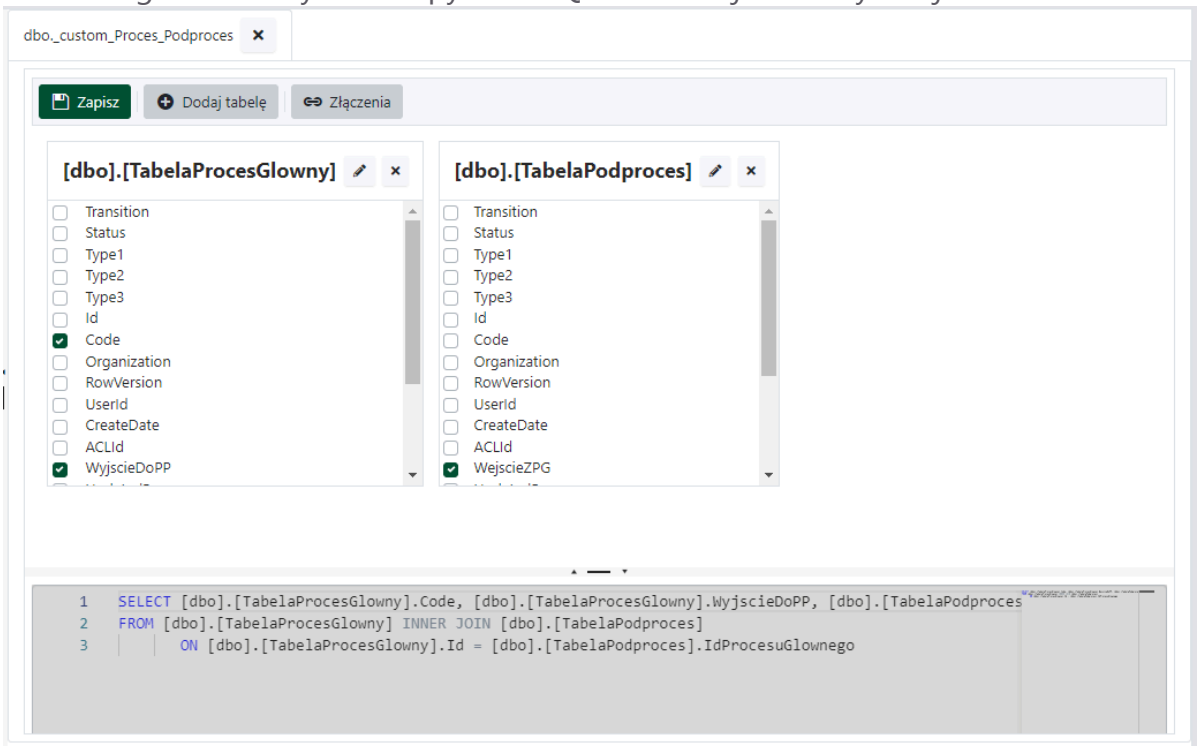
Dodaj tabelę do widoku

2. Kliknij przycisk `Złączenia`: po dodaniu co najmniej dwóch tabel kreator automatycznie tworzy złączenie między nimi, kliknięcie przycisku `Złączenia` pozwala edytować parametry tego złączenia, w tym typ złączenia, kolumnę klucza głównego i kolumnę klucza obcego; tabela dodana jako pierwsza jest domyślnie tabelą klucza głównego dla złączenia tabel i nie można tego zmienić.



Ustawienia złączeń między tabelami

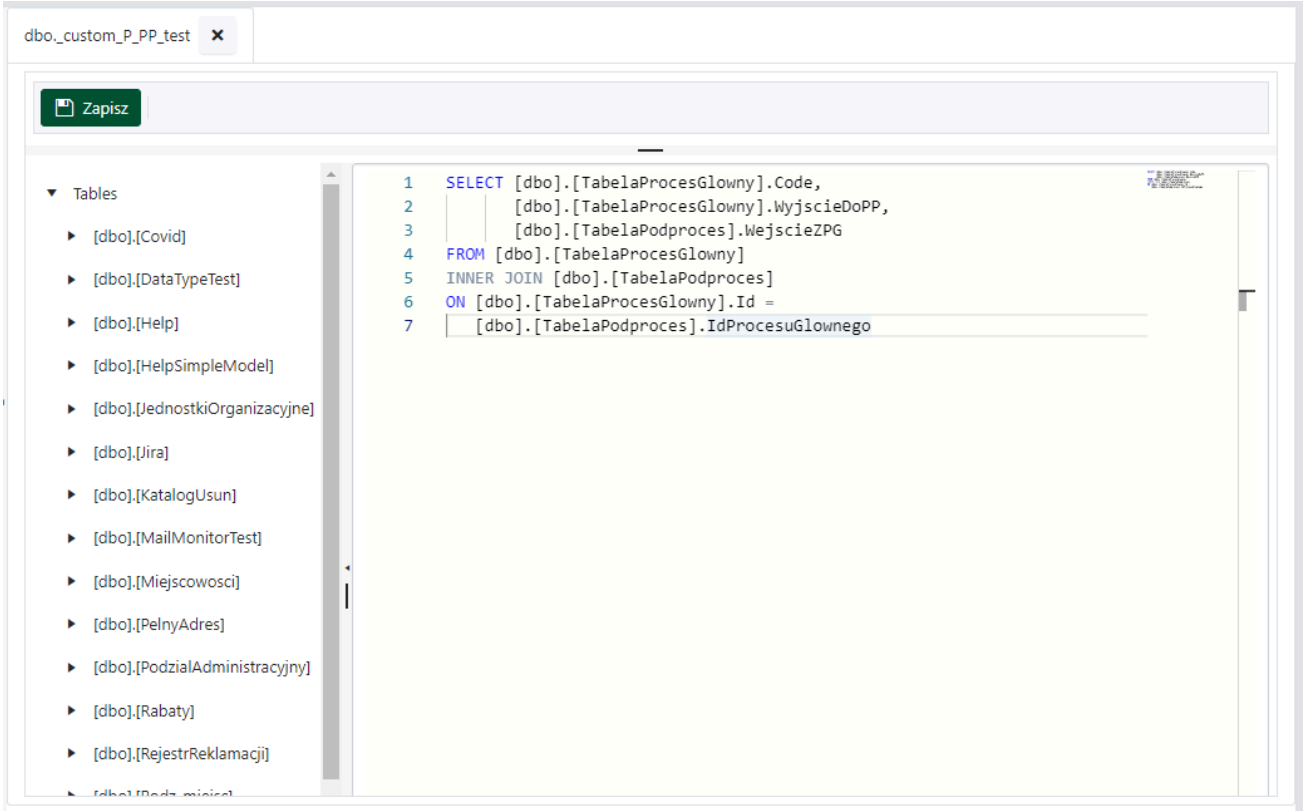
3. Zaznacz w wybranych tabelach pola, które mają być dostępne w funkcji tabelarycznej; nazwy zaznaczonych pól zostaną dopisane do klauzuli *SELECT* tworzonego automatycznie zapytania SQL dla funkcji tabelarycznej



Wybór pól do funkcji tabelarycznej

4. Kliknij przycisk `Zapisz`, aby zapisać funkcję tabelaryczną.

Polecenie `Dodaj funkcję (tryb zaawansowany)` wyświetla na panelu głównym kreatora okno edycji SQL, w którym należy ręcznie wpisać zapytanie dla funkcji tabelarycznej.

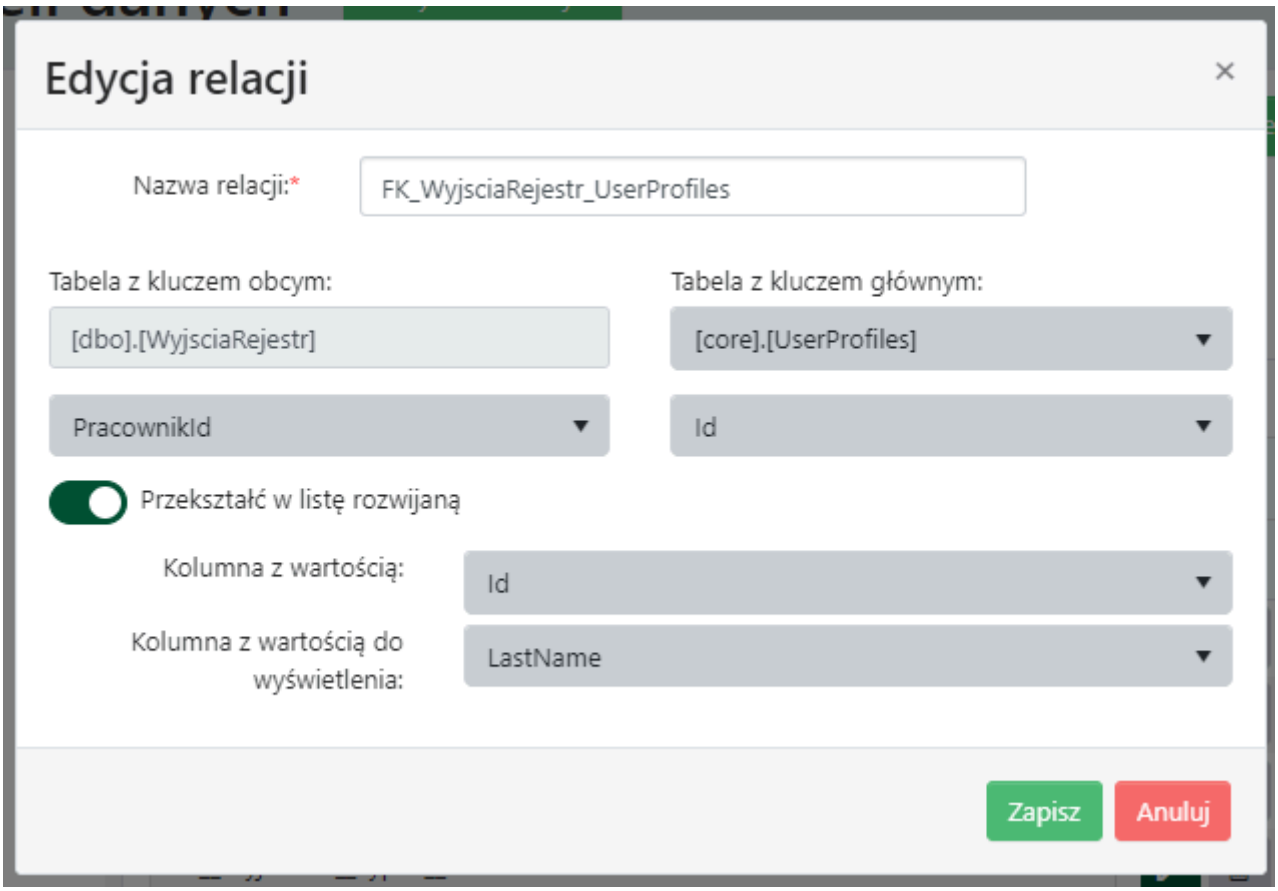


Funkcja tabelaryczna — tryb zaawansowany

3.1.8. Powiązania tabel

Podsumowanie:
Powiązanie dwóch tabel polega na wybraniu odpowiadających sobie kolumn w obu tabelach w celu wskazania związku, w jakim pozostają dane przechowywane w tych tabelach. W oparciu o takie powiązanie nAxiom może automatycznie konfigurować listy rozwijane na formularzach.

☛ *AdminSPA* > `DANE` > `Kreator modeli danych` > *Nazwa tabeli* > `Edycja` > `Powiązania`



Pole *PracownikId* można będzie wypełniać, wybierając nazwisko pracownika z listy rozwijanej

Określ następujące właściwości:

- `Nazwa relacji`: wpisz nazwę powiązania; standardowo nazwy zaczynają się od liter FK wskazujących, że w danej relacji tabela jest tabelą z kluczem obcym (ang. foreign key), tj. *FK_NazwaBieżącejTabeli_NazwaTabeliPowiązanej*.
- `Tabela z kluczem obcym`: pole tylko do odczytu z nazwą bieżącej tabeli, dla której tworzysz powiązanie.
    - lista kolumn tabeli: wybierz z listy kolumnę klucza obcego; ta kolumna zawiera wartości, na podstawie których identyfikowane będą powiązane rekordy w drugiej tabeli.
- `Tabela z kluczem głównym`: wybierz z listy nazwę tabeli, z którą chcesz powiązać bieżącą tabelę.
    - lista kolumn tabeli: wybierz z listy kolumnę tabeli, która zawiera dane klucza głównego.
- `Przekształć w listę rozwijaną`: (domyślnie wyłączone) po włączeniu na generowanym automatycznie formularzu to pole będzie reprezentowane przez listę rozwijaną; wybór wartości z listy spowoduje wpisanie wartości w kolumnie klucza obcego; ustaw następujące parametry dla tej listy:
    - `Kolumna z wartością`: wybierz kolumnę powiązanej tabeli, której wartości

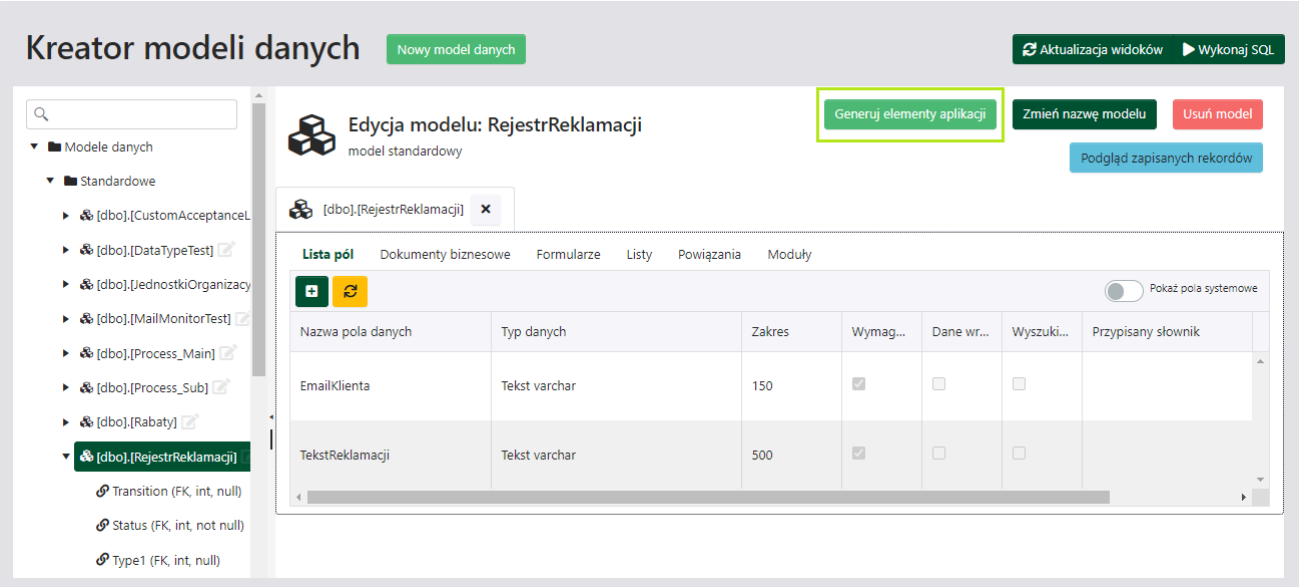
będą wpisywane w kolumnie klucza obcego.
    - `Kolumna z wartością do wyświetlenia`: wybierz kolumnę powiązanej tabeli, której wartości będą wyświetlane na liście rozwijanej.
- `Zapisz`: kliknij, aby zapisać powiązanie.
- `Anuluj`: kliknij, aby porzucić zmiany i zamknąć okno.

Kliknij przycisk `Zapisz` aby zapisać zmiany.

3.1.9. Generowanie elementów aplikacji

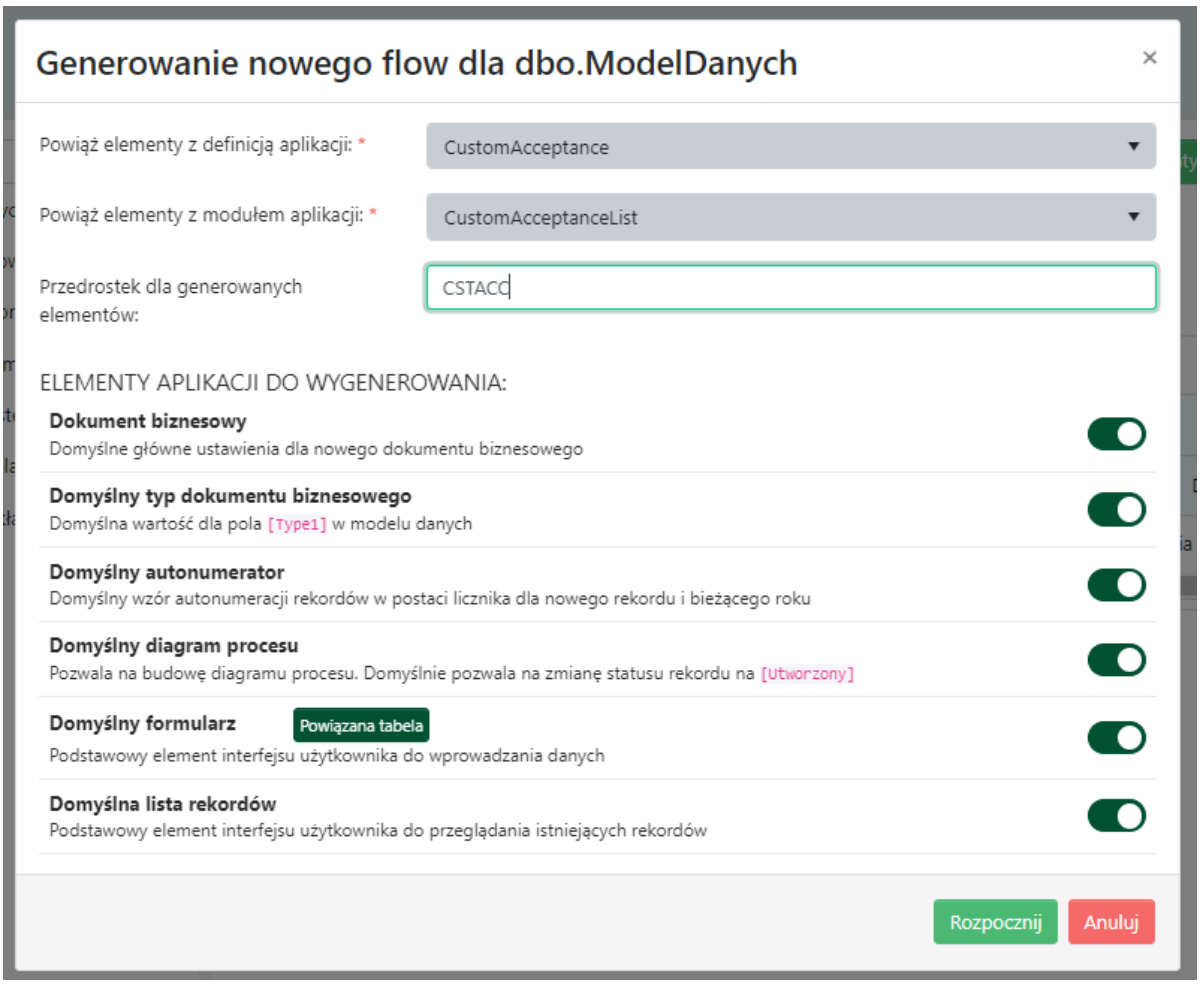
Podsumowanie:
Podstawowe elementy aplikacji nAxiom to: model danych (tabela), definicja dokumentu biznesowego, diagram procesu biznesowego, formularz i lista (widok listy). Po zdefiniowaniu standardowego modelu danych należy wygenerować na jego podstawie pozostałe elementy aplikacji. Służy do tego przycisk `Generuj elementy aplikacji` w nagłówku okna `Kreator modeli danych`.

☛ *AdminSPA* > `DANE` > `Kreator modeli danych` > `Generuj elementy aplikacji`



Przycisk Generuj elementy aplikacji

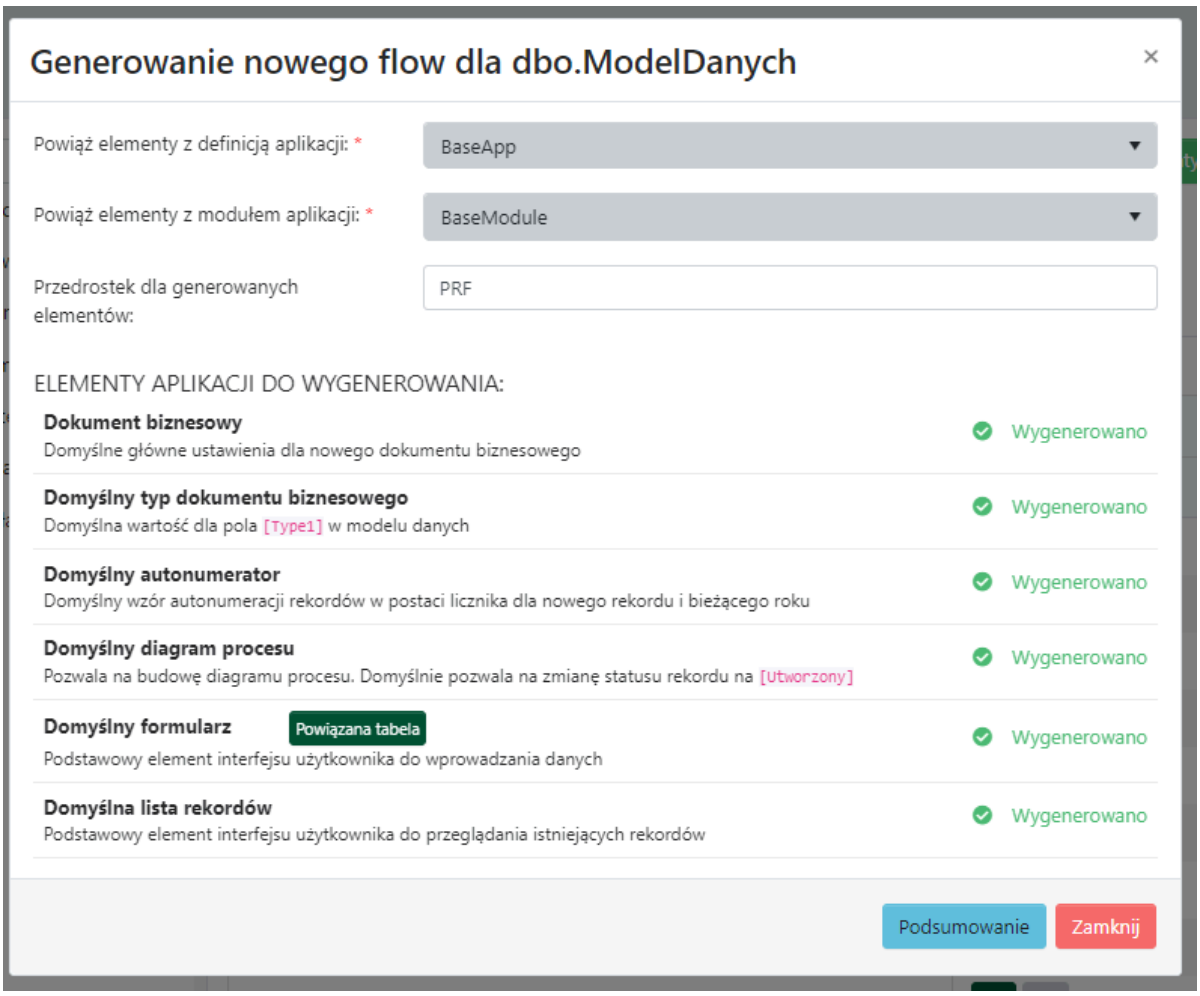
1. Kliknij przycisk `Generuj elementy aplikacji` w nagłówku panelu głównego. Zostanie wyświetlone okno dialogowe `Generowanie nowego flow dla...`.



Generowanie elementów aplikacji

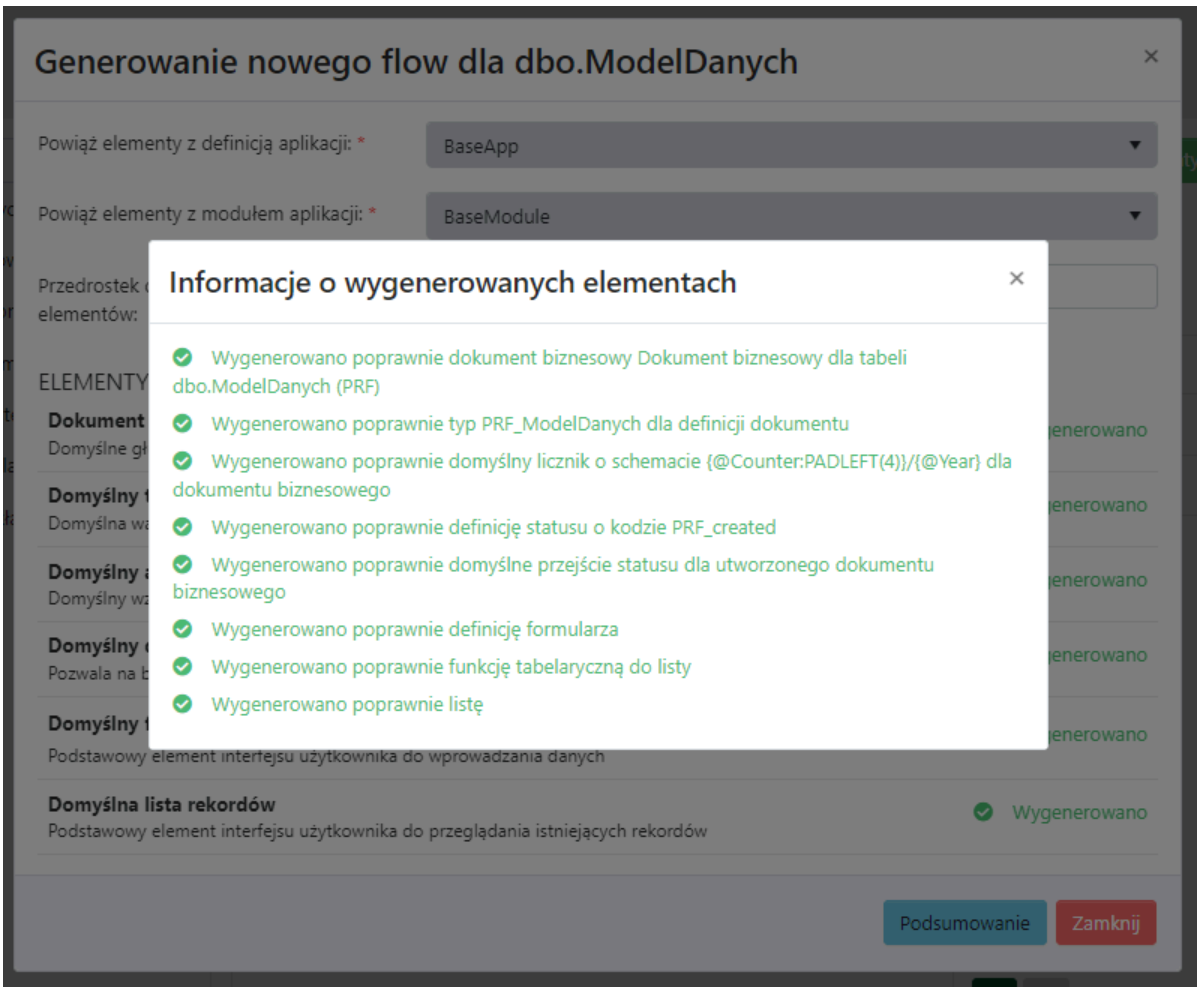
2. Określ ustawienia zgodnie z opisem w sekcji Kreator modeli danych, okno Generowanie elementów aplikacji .

3. Kliknij przycisk `Rozpocznij`.
   nAxiom potwierdzi utworzenie każdego elementu aplikacji napisem `Wygenerowano`.



Powodzenie generowania elementów aplikacji

4. Kliknij przycisk `Informacje`, aby wyświetlić okienko z informacjami o nazwach wygenerowanych elementów.



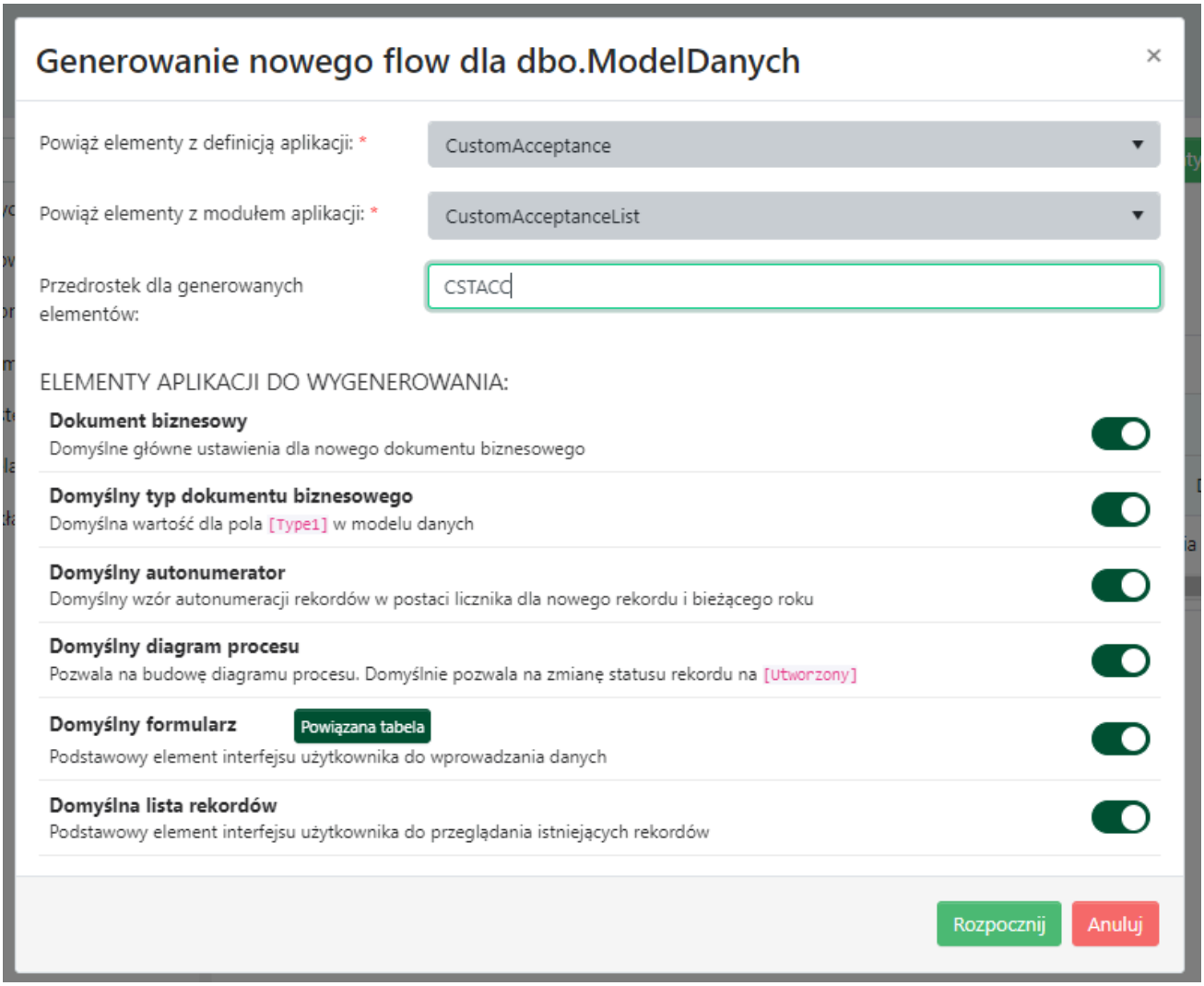
Informacje o wygenerowanych elementach

Utworzone w tym kroku elementy aplikacji można znaleźć w pozycjach menu `INTERFEJS UŻYTKOWNIKA` > `Listy` i `Formularze`, a także w pozycjach sekcji `PROCESY BIZNESOWE`.

Powiązane tematy:

- Dokumenty biznesowe (definicje dokumentów)
- Diagramy procesów
- Formularze
- Listy

3.1.10. Okno Generowanie elementów aplikacji



Generowanie elementów aplikacji

W oknie `Generowanie elementów aplikacji dla...` znajdują się następujące pola i elementy sterujące:

- `Powiąż elementy z definicją aplikacji`: wybierz nazwę aplikacji, w ramach której zostanie utworzony model procesu.
- `Powiąż elementy z modułem aplikacji`: wybierz nazwę modułu aplikacji, w ramach którego zostanie utworzony model procesu.
- `Przedrostek dla generowanych elementów`: wpisz przedrostek (co najmniej 3 znaki), którym zostaną poprzedzone nazwy tworzonych elementów.
- `Dokument biznesowy`: włącz lub wyłącz tworzenie definicji dokumentu biznesowego; wyłączenie spowoduje, że przycisk `Rozpocznij` będzie nieaktywny.
- `Domyślny typ dokumentu biznesowego`: włącz lub wyłącz tworzenie typu dla definicji dokumentu biznesowego; po włączeniu definiuje wartość w kolumnie `Type1`.

- `Domyślny autonumerator`: włącz lub wyłącz tworzenie schematu numerowania tworzonych dokumentów.
- `Domyślny diagram procesu`: włącz lub wyłącz tworzenie domyślnego diagramu procesu, który obejmuje pierwszy krok od nowego dokumentu do statusu `Utworzony`.
- `Domyślny formularz`: włącz lub wyłącz tworzenie formularza z polami bieżącej tabeli.
    - `Powiązana tabela`: wyświetla okno dialogowe w celu wyboru tabeli powiązanej z tabelą dokumentu biznesowego, która będzie źródłem danych sekcji na formularzu. Takie powiązanie należy zdefiniować wcześniej w oknie `Kreator modeli danych` na karcie `Powiązania`.
- `Domyślna lista rekordów`: włącz lub wyłącz tworzenie widoku listy z polami z bieżącej tabeli.
- `Rozpocznij`: kliknij, aby utworzyć nowy model procesu obejmujący wszystkie wybrane elementy.
- `Anuluj`: kliknij, aby zamknąć okno.
- `Informacje`: kliknij, aby wyświetlić okienko z informacjami o nazwach wygenerowanych elementów modelu procesu.

WSKAZÓWKA

Najlepszą praktyką jest pozostawienie wszystkich przełączników w stanie domyślnym (włączonych) i automatyczne utworzenie wszystkich elementów aplikacji. W kolejnych krokach można modyfikować poszczególne elementy odpowiednio do wymagań.

Powiązane tematy:

- Dokumenty biznesowe (definicje dokumentów)
- Typy dokumentów
- Autonumeracja dokumentów
- Definicje statusów
- Diagramy procesów
- Formularze
- Listy

3.2. ŹRÓDŁA DANYCH

Podsumowanie:
Aplikacje tworzone w nAxiom mogą przetwarzać dane nie tylko z tabel definiowanych w kreatorze modeli danych, ale także z innych źródeł. Dane z takich źródeł mogą być dostępne przy użyciu zapytań REST API i SOAP API (akcje zapytań do webservice) oraz przez połączenia z innymi bazami danych SQL Server i Oracle. Do źródeł danych można się odwoływać w akcjach (zapytania do webserwisów, SQL do baz zewnętrznych, SQL między bazami danych) oraz w zapytaniach SQL. W niektórych polach właściwości źródła danych jest obsługiwana składnia *smart numbers*. W połączeniu ze zmiennymi aplikacji, pozwala to na przykład używać różnych źródeł danych w różnych środowiskach.

Na stronie `Źródła danych` jest wyświetlana lista zdefiniowanych źródeł danych. Na tej stronie można dodawać definicje nowych źródeł danych (przycisk `Nowe źródło danych` w lewym górnym rogu) oraz edytować definicje i usuwać je (przyciski `Edytuj` i `Usuń` obok definicji źródła danych).

Okno dialogowe `Nowe źródło danych` / `Edycja źródła danych` zawiera poniższe ustawienia.

- `Kod`: kod źródła danych; może zawierać tylko litery (bez diakrytycznych), cyfry i znaki podkreślenia.
- `Nazwa`: nazwa tworzonego źródła danych.
- `Aplikacja`: nazwa aplikacji, do której zostanie przypisane źródło danych.
- `Moduł`: nazwa modułu aplikacji, do którego zostanie przypisane źródło danych.
- `Typ źródła danych`: rodzaj źródła danych; dostępne wartości:
  - `Api: REST`: źródło danych obsługiwane za pomocą zapytań i odpowiedzi wymienianych z serwerem, który udostępnia interfejs REST API.
  - `Api: Soap`: źródło danych obsługiwane za pomocą zapytań i odpowiedzi wymienianych z serwerem, który udostępnia interfejs API z protokołem SOAP.
  - `Tenant w środowisku nAxiom`: źródło danych wskazuje na interfejs PublicAPI bieżącego środowiska nAxiom i umożliwia przesyłanie żądań między tenantami. Ten typ źródła danych nie ma żadnych parametrów konfiguracyjnych. W żądaniach korzystających ze źródła tego typu używany jest mechanizm automatycznego uwierzytelniania oparty na jednorazowych tokenach TOTP.
  - `Baza danych: Microsoft SQL (SQL Server)`: zewnętrzna (lub

podstawowa[1]) baza danych SQL Server; po zaznaczeniu opcji `Parametry połączenia` należy wypełnić pola `Adres serwera` (nazwa hosta lub adres IP), `Port` (można użyć smart numbers), `Nazwa użytkownika` `Hasło`, `Nazwa bazy danych` (można użyć smart numbers) i `Nazwa instancji` (można użyć smart numbers); alternatywnie można zaznaczyć opcję `Connection string` i wpisać odpowiednie parametry w formie ciągu połączenia.

  - `Baza danych: Oracle`: zewnętrzna baza danych Oracle; po zaznaczeniu opcji `Parametry połączenia` należy wypełnić pola `Adres serwera`, `Port`, `Nazwa użytkownika`, `Hasło` (można użyć smart numbers), `Nazwa bazy danych (SERVICE_NAME)` i `Nazwa instancji (SID)`. ; alternatywnie można zaznaczyć opcję `Connection string` i wpisać odpowiednie parametry w formie ciągu połączenia

- `Typ uwierzytelniania`: rodzaj uwierzytelniania, który ma być stosowany do połączeń z serwerem udostępniającym źródło danych REST lub SOAP API; dostępne wartości:
  - `Brak uwierzytelniania`: (wartość domyślna) połączenia ze źródłem danych nie wymagają uwierzytelniania.
  - `Podstawowa autentykacja (Base64)`: wymaga wypełnienia pól `Nazwa użytkownika` i `Hasło`.
  - `JSON Web Token (JWT)`: wymaga wypełnienia pól `Serwer autoryzacji` (można użyć smart numbers), `Nazwa użytkownika` i `Hasło`.
  - `OAuth 2.0 Password`: wymaga wypełnienia pól `Serwer autoryzacji` (można użyć smart numbers), `Nazwa użytkownika`, `Hasło`, `Scope`, `Client ID` i `Client Secret`.
  - `OAuth 2.0 Client Credentials`: wymaga wypełnienia pól `Serwer autoryzacji` (można użyć smart numbers), `Scope`, `Client ID` i `Client Secret`.
- `Adres serwera`: adres serwera, który udostępnia źródło danych; w tym polu obsługiwana jest składnia smart numbers.
- `Open api url`: adres URL specyfikacji Open API tworzonego źródła danych; po podaniu tego adresu można zaimportować specyfikację źródła danych i automatycznie utworzyć z niej akcje zapytań do webserwsów; w tym polu obsługiwana jest składnia smart numbers.
- `Open api plik`: plik w formacie JSON ze specyfikacją Open API tworzonego źródła danych; po wskazaniu tego pliku można zaimportować specyfikację źródła danych i automatycznie utworzyć z niej akcje zapytań do webserwsów.
- `Opis`: opis źródła danych.

- `Zapisz i importuj akcje`: przycisk dostępny tylko dla źródeł typu REST API; zapisuje definicję źródła danych i importuje specyfikację OPEN API źródła danych, automatycznie generując akcje zapytań do webserwisów, z których można korzystać jak z szablonów żądań do danego serwisu.
- `Zapisz`: kliknij, aby zapisać definicję źródła danych.
- `Anuluj`: kliknij, aby zamknąć to okno dialogowe.

WSKAZÓWKA
Źródła danych są obsługiwane przez składnię smart numbers. Odpowiednia referencja ma postać:

```
{@<kod_aplikacji>.DataSources.<kod_modułu>.<kod_źródła_danych>}
```

1. Źródło danych odwołujące się do podstawowej bazy danych bieżącej witryny może być potrzebne na przykład w celu zdefiniowania repozytorium na pliki statyczne. ↩

Powiązane tematy:
- Akcja zapytań do webservice
- Miejsca użycia smart numbers
- Definiowanie zmiennych aplikacji
- Repozytoria załączników

3.3. REPOZYTORIA ZAŁĄCZNIKÓW

Podsumowanie:
W nAxiom dostępne są funkcjonalności umożliwiające zarządzanie plikami statycznymi i ich przetwarzanie. Takie pliki to między innymi załączniki, odebrane wiadomości e-mail, rozpoznane pliki OCR i dodatkowe biblioteki. Do przechowywania plików służą w nAxiom repozytoria. Repozytoria mogą być dyskowe (foldery) lub bazodanowe (tabele w bazie danych).

☛ *AdminSPA* > `DANE` > `Repozytoria załączników`

W TYM TEMACIE

- Instalacja
- Konfiguracja dla tenantów
- Zarządzenie repozytoriami
- Wykorzystanie repozytoriów w nAxiom
- Tabela core.AttachmentRepositories

INSTALACJA

Rodzaj repozytoriów (dyskowe/bazodanowe) dla pierwszego tenanta wybiera się podczas instalacji witryny nAxiom w kroku `Pierwszy tenant - zasoby systemu`. W przypadku aktualizacji witryny ten krok jest niedostępny.

W instalatorze dostępne są cztery typy zasobów, dla których można zdecydować, czy ich repozytorium ma się znajdować w folderze instalacyjnym, czy w bazie danych. Tym typom zasobów odpowiadają tworzone domyślnie repozytoria. W przypadku rozpoznanych plików OCR repozytorium bazodanowe jest nieobsługiwane, ale można wskazać inny folder repozytorium. To repozytorium jest używane wyłącznie na potrzeby przetwarzania OCR, a jego konfiguracja nie jest dostępna w aplikacji *AdminSPA*. Można ją zmienić w pliku *appsettings.json* serwisu *ocrapi*.

Pozostałe rodzaje zasobów i odpowiadające im repozytoria to:

- *Załączniki dokumentów*: dla załączników tworzone jest *Domyślne repozytorium*, które standardowo znajduje się w folderze *folder-instalacyjny-naxiom/SystemRepositories/tenant-GUID/default*; to repozytorium jest standardowo wybrane w domyślnej kategorii załączników.
- *Wiadomości email*: dla załączników z wiadomości e-mail tworzone jest *Domyślne repozytorium e-mail*, które standardowo znajduje się w folderze *folder-instalacyjny-naxiom/SystemRepositories/tenant-GUID/email*; w tym repozytorium domyślnie są zapisywane również załączniki dodawane do szablonów wiadomości e-mail.
- *Dodatkowe biblioteki*: dla tej kategorii tworzone jest *Domyślne repozytorium systemowe*, które standardowo znajduje się w folderze *folder-instalacyjny-naxiom/SystemRepositories/tenant-GUID/system*; w tym repozytorium domyślnie zapisywane są własne pliki logo wybierane w kreatorze stylów.

Foldery repozytoriów są faktycznie tworzone na dysku dopiero w momencie zapisu pliku w repozytorium. W przypadku repozytoriów bazodanowych można określić bazę danych (może to być inna baza niż podstawowa baza nAxiom) i tabelę w bazie, w której będą

przechowywane pliki statyczne. Na podstawie podanych parametrów połączenia z bazą danych instalator utworzy źródło danych. Jeśli taka tabela nie istnieje, zostanie utworzona.

W aplikacji AdminSPA jest dostępne jeszcze jedno repozytorium systemowe *Default attachment repository in the tenant database*. Jest to repozytorium bazodanowe, które znajduje się w tabeli *NaxiomDefaultRepoForAttachmentFiles* w bazie danych tenanta. Schemat tabeli odpowiada schematowi tabel użytkownika wybranemu podczas instalacji (domyślnie *dbo*), ale można to zmienić w aplikacji TenantsAdminSPA. Zmiana nazwy lub schematu tabeli spowoduje utworzenie nowej, pustej tabeli (wymagane kliknięcie przycisku `Sprawdź dostęp`). Wcześniej używana tabela nie zostanie usunięta.

KONFIGURACJA DLA TENANTÓW

W trybie obsługi wielu tenantów domyślna lokalizacja repozytoriów dyskowych każdego tenanta to folder o nazwie będącej identyfikatorem GUID tego tenanta. W ogólności ścieżka do folderu repozytoriów tenanta ma postać: *folder-instalacyjny-naxiom/SystemRepositories/GUID-tenanta*.

Administrator tenantów może wskazać lokalizację inną niż *folder-instalacyjny-naxiom/SystemRepositories* poprzez ustawienie `Directory location for relative paths of file repositories` (*TenantAdminSPA* > `Global settings`).

Ponadto dla każdego tenanta można indywidualnie określić nazwę jego podkatalogu, podając inną niż domyślna (GUID) wartość klucza *SubDirName* w sekcji *FileStorageConfiguration* w ustawieniach konfiguracyjnych tenanta (*TenantAdminSPA* > `Edit tenant` > `Configuration` > `Settings`).

Administrator tenantów może także zezwolić poszczególnym tenantom na swobodne określanie lokalizacji repozytoriów za pomocą drugiego parametru w sekcji *FileStorageConfiguration*, *AllowUseRepoFullDir*. Określa on, czy konsultanci danego tenanta mogą definiować repozytoria, używając ścieżek względnych, czy bezwzględnych.

Dla pierwszego tenanta wartość tego klucza to *true* ze względu na konieczność zapewnienia zgodności z konfiguracjami sprzed wprowadzenia obsługi wielu tenantów. Dla kolejnych tenantów parametr ten ma domyślnie wartość *false*, co oznacza, że podawane przez nich nazwy repozytoriów będą traktowane jako ścieżki względne w lokalizacji określonej ustawieniem `Directory location for relative paths of file repositories`.

Aby zmienić domyślny schemat tabeli z systemowym repozytorium bazodanowym, należy wpisać ją w konfiguracji tenanta jako wartość parametru *DefaultRepoForAttachmentFilesSchema*. Jeśli ten parametr nie jest widoczny, należy zapisać bieżącą konfigurację i ponownie uruchomić witrynę.

ZARZĄDZENIE REPOZYTORIAMI

Po wybraniu pozycji menu `Repozytoria załączników` (*AdminSPA* > `DANE`) wyświetlana jest lista repozytoriów zdefiniowanych w witrynie. Standardowo dostępne są repozytoria wymienione powyżej. Konsultant może tworzyć nowe repozytoria oraz edytować i usuwać repozytoria istniejące.

> **UWAGA**
> Usunięcie repozytorium powoduje usunięcie definicji repozytorium z tabeli *core.AttachmentRepositories*. Samo repozytorium (folder lub tabela w bazie danych) i jego zawartość pozostają nienaruszone.

W przypadku tworzenia nowego lub edycji istniejącego repozytorium należy określić następujące właściwości:

- `Kod`: dowolny ciąg znaków.
- `Nazwa`: nazwa repozytorium; jest wyświetlana na listach wyboru w miejscach, w których nalezy wybrać repozytorium, na przykład w definicji kategorii załączników.
- `Typ repozytorium`: dostępne wartości:

- `Folder lokalny`: należy wskazać folder — nazwę lub ścieżkę, odpowiednio do konfiguracji w *TenantsAdminSPA* — w którym będą przechowywane pliki statyczne.
- `Baza danych MS SQL Server`: w tym przypadku należy podać dwa dodatkowe parametry:
    - `Źródło danych`: zdefiniowane osobno źródło danych odwołujące się do bazy MS SQL Server; może to być baza używana przez bieżącą witrynę.
    - `Zapisuj do tabeli`: nazwa tabeli, w której będą zapisywane pliki statyczne; z reguły będzie to nazwa nowej tabeli, która zostanie utworzono po kliknięciu przycisku `Sprawdź połączenie`. Tabela repozytorium ma dwie kolumny *Id* (*uniqueidentifier*) i *Content* (*varbinary(MAX)*).

    > **UWAGA**
    > Jeśli nie zostanie podany schemat tabeli, tabela zostanie utworzona w schemacie domyślnym *dbo*. Aby utworzyć tabelę w schemacie *core*, należy poprzedzić nazwę schematem, na przykład *core.MojeRepozytorium*.

> **WSKAZÓWKA**
> Typ `Baza danych tenanta` jest zastrzeżony tylko dla repozytorium systemowego i nie jest dostępny dla innych repozytoriów.

- `Folder lokalny`: należy wskazać folder — nazwę lub ścieżkę, odpowiednio do konfiguracji w *TenantsAdminSPA* — w którym będą przechowywane pliki statyczne.
- `Baza danych MS SQL Server`: w tym przypadku należy podać dwa dodatkowe parametry:
    - `Źródło danych`: zdefiniowane osobno źródło danych odwołujące się do bazy MS SQL Server; może to być baza używana przez bieżącą witrynę.
    - `Zapisuj do tabeli`: nazwa tabeli, w której będą zapisywane pliki statyczne; z reguły będzie to nazwa nowej tabeli, która zostanie utworzono po kliknięciu przycisku `Sprawdź połączenie`. Tabela repozytorium ma dwie kolumny *Id* (*uniqueidentifier*) i *Content* (*varbinary(MAX)*).

      > **UWAGA**
      > Jeśli nie zostanie podany schemat tabeli, tabela zostanie utworzona w schemacie domyślnym *dbo*. Aby utworzyć tabelę w schemacie *core*, należy poprzedzić nazwę schematem, na przykład *core.MojeRepozytorium*.

- `Baza danych tenanta`: repozytorium w tabeli bazy danych bieżącego tenanta; należy podać nazwę tabeli do zapisywania plików.
- `Aktywne`: określa, czy repozytorium jest aktywne; dezaktywowanie repozytorium, które jest używane w danej witrynie, uniemożliwi zapis i odczyt plików w tym repozytorium.

WYKORZYSTANIE REPOZYTORIÓW W NAXIOM

Repozytoria plików statycznych mają w nAxiom następujące przeznaczenie:

- przechowywanie załączników dodawanych do dokumentów, w tym generowanych przez różne akcje,
- przechowywanie załączników przesyłanych w przychodzących wiadomościach e-mail,
- przechowywanie załączników dodawanych w definicji szablonu e-mail,
- przechowywanie indywidualnych plików logo konfigurowanych w kreatorze stylów.

W przypadku załączników do dokumentów repozytorium nie określa się indywidualnie dla załącznika tylko poprzez kategorię załączników. W momencie generowania elementów aplikacji system powołuje kategorię domyślną, która jest powiązana z repozytorium domyślnym. Jest to repozytorium o nazwie *Domyślne repozytorium*, standardowo lokalizowane w folderze *default* z flagą *IsDefault* o wartosci *1*. Konsultant może zmienić repozytorium dla kategorii domyślnej w istniejących definicjach dokumentów, ale dla nowej definicji w procesie generowania elementów aplikacji zawsze kategoria domyślna będzie powiązana z repozytorium domyślnym.

Ponadto w ustawieniach systemu wskazuje się trzy repozytoria:

- `Ustawienia systemu` > `Ogólne` > `System - Repozytorium systemowe`: miejsce zapisu własnych plików logo konfigurowanych w kreatorze stylów (standardowo *Domyślne repozytorium systemowe*).
- `Ustawienia systemu` > `Email` > `Repozytorium zapisu plików dla szablonów e-mail`: miejsce zapisu załączników dodawanych w definicji szablonu e-mail (standardowo *Domyślne repozytorium e-mail*).
- `Ustawienia systemu` > `Email` > `Exchange - Repozytorium zapisu plików dla załączników pobieranych z wiadomości`: miejsce zapisu załączników do wiadomości e-mail odbieranych przez tzw. *mail monitory* (standardowo *Domyślne repozytorium e-mail*).

TABELA CORE.ATTACHMENTREPOSITORIES

Definicje repozytoriów są przechowywane w tabeli *core.AttachmentRepositories*, która składa się z następujących kolumn:

Nazwa kolumny	Typ danych	Zezwól NULL	Opis
Id	int	☐	Kolumna klucza głównego, identyfikator rekordu w tabeli
Code	nvarchar(32)	☐	Kod repozytorium
Name	nvarchar(128)	☐	Nazwa repozytorium
IsActive	bit	☐	Flaga aktywności repozytorium
IsDefault	bit	☐	Flaga repozytorium domyślnej kategorii załączników
RepositoryType	int	☐	Typ repozytorium: 0 - folder, 1 - baza danych
DataSourceId	int	☑	Identyfikator źródła danych dla repozytorium bazodanowego
RelativeDirectory	nvarchar(MAX)	☑	Ścieżka względna do folderu repozytorium
AdvancedSaveOptions	nvarchar(MAX)	☑	Dane w formacie JSON z dodatkowymi parametrami: *SubDirectorySaveType*, *NumberOfFiles* i *LastFolderName*
RowGuid	uniqueidentifier	☐	Unikatowy identyfikator szablonu
AbsoluteDirectory	nvarchar(MAX)	☑	Ścieżka bezwzględna do folderu repozytorium
TableName	nvarchar(128)	☑	Nazwa tabeli repozytorium bazodanowego

Powiązane tematy:

- Źródła danych
- Kategorie załączników
- Ustawienia systemu, sekcja Ogólne
- Ustawienia systemu, sekcja Email
- Szablony e-mail

3.4. KATEGORIE ZAŁĄCZNIKÓW

Podsumowanie:
Kategorie załączników to elementy wspomagające zarządzanie załącznikami dodawanymi do dokumentów. Kategorie definiuje się osobno dla każdej definicji dokumentu. Dla nowej kategorii należy wskazać definicję dokumentu i repozytorium, w którym będą zapisywane załączniki. Ponadto można włączyć lub wyłączyć wersjonowanie załączników oraz opcjonalnie określić dozwolone rozszerzenia nazw plików załączników.

Podczas tworzenia definicji dokumentu automatycznie tworzona jest domyślna kategoria załączników, która ma włączone wersjonowanie załączników i obsługuje takie same formaty plików jak określone w ustawieniach systemowych, w sekcji `Załączniki`.

Informacje o kategoriach załączników są zapisywane w tabeli *core.AttachmentCategories*.

☛ *AdminSPA* > `DANE` > `Kategorie załączników`

Po wybraniu polecenia `Kategorie załączników` jest wyświetlana lista zdefiniowanych kategorii. Domyślnie lista jest pogrupowana według definicji dokumentu. Dla każdej kategorii są dostępne przyciski `Edytuj` i `Usuń`. Przy czym nie można usunąć kategorii:

- zawierającej załączniki,
- wybranej dla sekcji załączników i/lub podglądu załączników na formularzu,
- jedynej kategorii zdefiniowanej dla definicji dokumentu.

Przycisk `Nowa kategoria załączników` wyświetla okno, w którym można zdefiniować nową kategorię, określając następujące właściwości:

- `Nazwa`: nazwa kategorii załączników; domyślnie używana jako etykieta sekcji załączników i podglądu załączników na formularzu.
- `Opis`: opis kategorii.
- `Wersjonowanie załączników`: włączenie spowoduje, że dodanie do dokumentu kolejnego załącznika o takiej samej nazwie pliku spowoduje dodanie go jako nowej wersji istniejącego załacznika; jeśli przełącznik bedzie wyłączony, w takiej sytuacji oryginalny załącznik zostanie nadpisany.
- `Repozytorium`: wybierz z listy repozytorium, w którym będą zapisywane pliki załączników.
- `Definicja dokumentu`: wybierz z listy definicję dokumentu, do której zostanie

przypisana kategoria.

- `Dozwolone rozszerzenia załączników`: (opcjonalnie) wpisz rozdzielane średnikami rozszerzenia nazw plików, które mają być obsługiwane w tej kategorii; jeśli nie zostaną określone, dozwolone będą wszystkie rozszerzenia zdefiniowane w ustawieniach tenanta ; w polu jest obsługiwana notacja smart numbers. Jeśli lista dozwolonych typów załączników będzie pusta, rozszerzenia załączników nie będą weryfikowane.

- `Maksymalny rozmiar pliku (KB)`: (opcjonalnie) określa maksymalny rozmiar pliku załącznika (domyślnie 10240 KB); jeśli nie zostanie określony, obowiązuje limit z ustawień systemowych (51200 KB); w polu jest obsługiwana notacja smart numbers. Ustawieniem nadrzędnym jest konfiguracja serwera IIS

Kategoria załączników jest ustawiana jako właściwość sekcji załączników i podglądu załączników w definicji formularza. Na formularzu można zdefiniować osobne sekcje załączników i podglądu załączników dla różnych kategorii załączników.

Kategoria domyślna ma automatycznie włączaną flagę *IsDefault*. Zmiana ustawienia tej flagi jest możliwa tylko z poziomu bazy danych w tabeli *core.AttachmentCategories*. Należy przy tym uważać, aby nie włączyć tej flagi dla kilku kategorii w obrębie jednego dokumentu biznesowego.

UWAGI DOTYCZĄCE MIGRACJI

Jeśli w importowanej aplikacji kategoria ma przypisane jedno z repozytoriów systemowych, to przypisanie zawsze nadpisze przypisanie w aplikacji docelowej. Jeśli natomiast importowana kategoria ma przypisane repozytorium niestandardowe, przypisanie w aplikacji docelowej nie zmieni się. W przypadku importu nowej kategorii z repozytorium niestandardowym zostanie utworzona kategoria z przypisanym domyślnym repozytorium załączników.

Powiązane tematy:

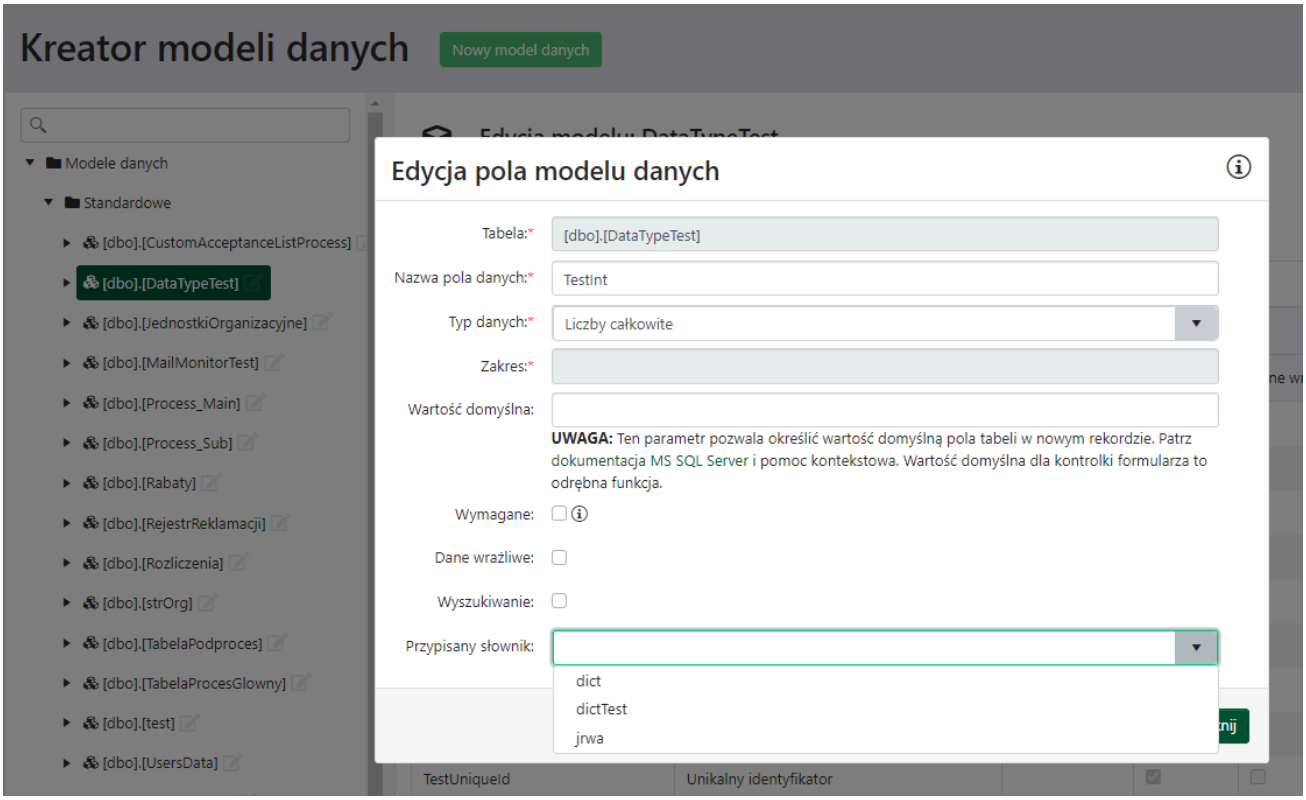
- Ustawienia systemu, sekcja Załączniki
- Repozytoria załączników
- Sekcje formularza - Załączniki
- Sekcje formularza - Załączniki - podgląd

3.5. SŁOWNIKI

Podsumowanie:
Słowniki to źródła danych dla list rozwijanych. Pozycje słownika to zazwyczaj wartości tekstowe, definiowane przez użytkowników końcowych aplikacji biznesowych. W nAxiom można definiować słowniki proste i hierarchiczne. Te ostatnie zawierają pozycje, które są ze sobą powiązane relacją nadrzędna-podrzędne: każda pozycja może mieć pozycje podrzędne.

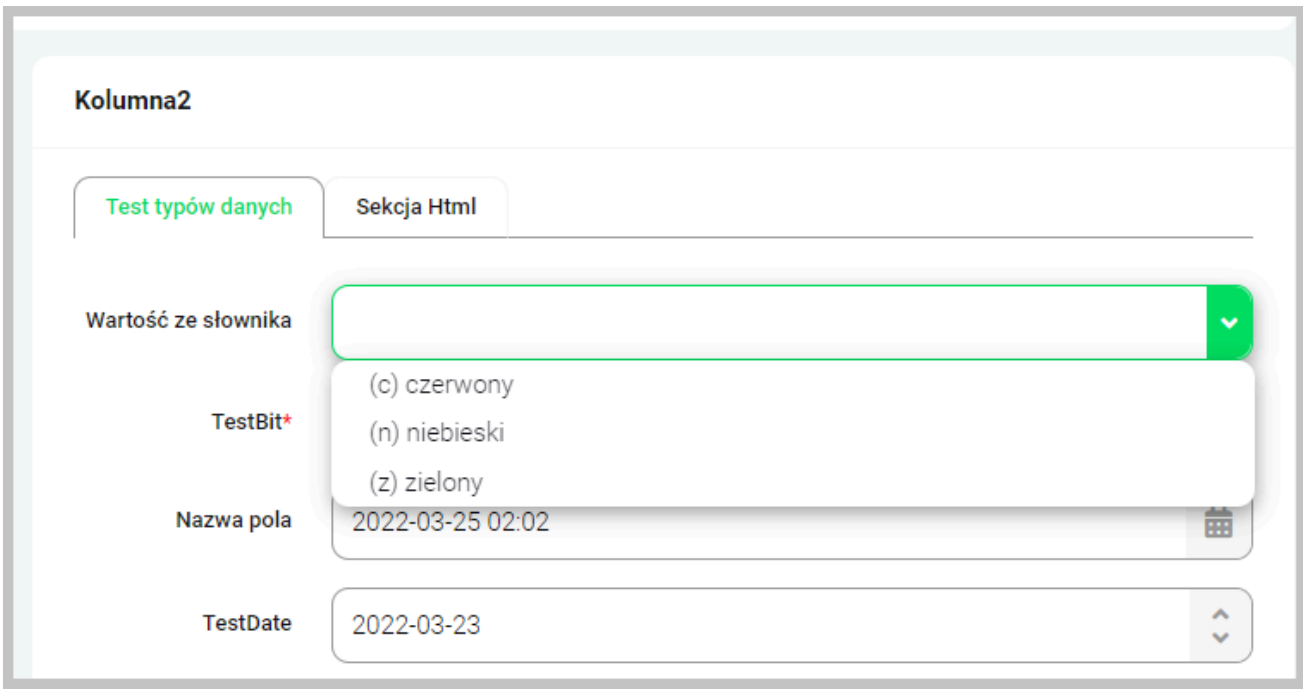
☛ *AdminSPA* > `DANE` > `Słowniki` *FrontSPA* > `Ustawienia systemowe` > `Słowniki`

Słowniki można przypisywać na poziomie modelu danych do kolumn tabeli, zaznaczając pole wyboru `Pozycja słownikowa` (dostępne tylko dla kolumn typu *int*); patrz Kreator modeli danych, nowe pole modelu danych .



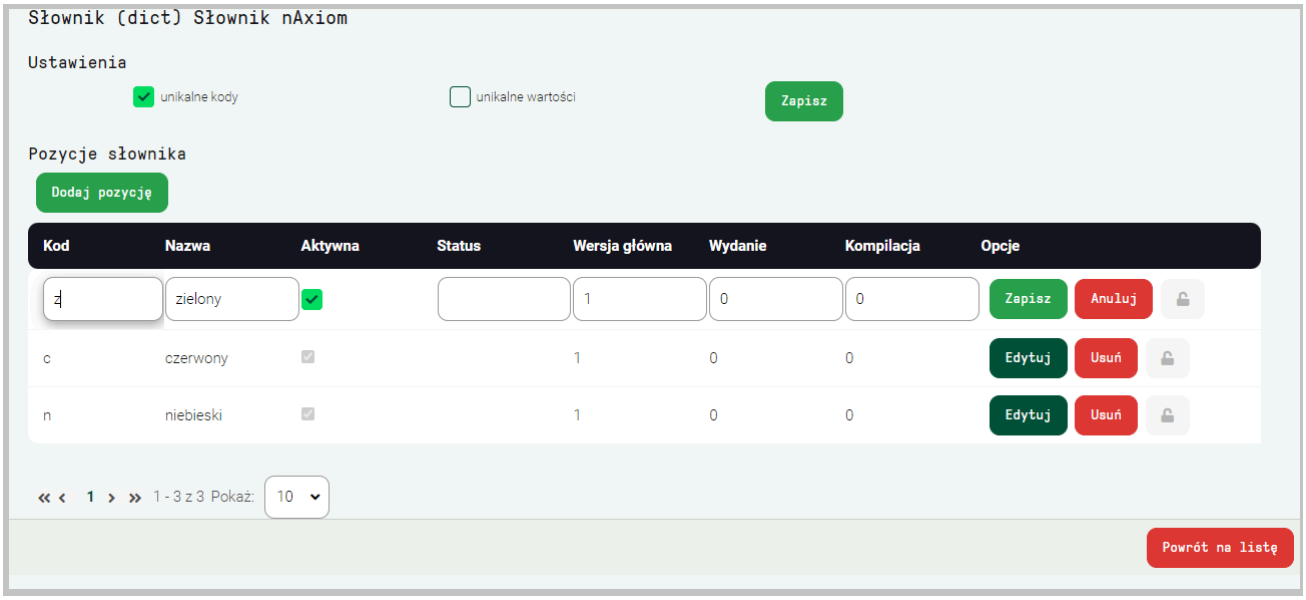
Definicja kolumny tabeli — przypisanie słownika

Po przypisaniu słownika do kolumny tabeli i użyciu polecenia `Generuj elementy aplikacji`, w automatycznie wygenerowanym formularzu do wypełniania pola tej kolumny zostanie zastosowana lista rozwijana typu słownik, na której będą wyświetlane pozycje słownika. (Należy pamiętać, że pozycje słownika wybierane z listy to wartości wyświetlane, wartości zapisywane w tym polu to liczbowe identyfikatory pozycji słownika).



Front-office: lista rozwijana z hasłami ze słownika na formularzu

Słowniki definiuje się po wybraniu polecenia `Słowniki` w sekcji `DANE` menu głównego AdminSPA. Natomiast zarządzanie pozycjami słownika można powierzyć określonym użytkownikom końcowym, przyznając im stosowne uprawnienia PBA (*FrontSPA* > `Ustawienia systemowe` > `Uprawnienia` > `Użytkowanie aplikacji` > `Zarządzanie pozycjami słowników`). Te uprawnienia obejmują: podgląd listy słowników i pozycji słowników oraz tworzenie, aktualizację i usuwanie pozycji słowników.



FrontSPA: edycja pozycji słownika (w menu po lewej zaznaczono pozycję Słowniki)

Definicje słowników prostych są zapisywane w bazie danych w tabeli *core.Dictionaries*, a pozycje słowników w tabeli *core.DictionaryValues*. Odpowiednie tabele dla słowników hierarchicznych to *core.HierarchicalDictionaries* i *core.HierarchicalDictionaryValues*.

Słowniki mogą być migrowane wraz z aplikacjami do innych środowisk, zależy to od ustawienia w definicji słownika. Aby uniknąć sytuacji, w której pozycje słownika w środowisku docelowym są nadpisywane/usuwane, dla każdej pozycji słownika można indywidualnie włączać/wyłączać zabezpieczenie. Służy do tego ikona kłódki widoczna obok każdej pozycji na liście pozycji słownika wyświetlonej w aplikacji *FrontSPA*.

Powiązane tematy:

- Kreator modeli danych, nowe pole modelu danych
- Uprawnienia PBA, obszar Użytkowanie aplikacji

3.5.1. Słowniki proste

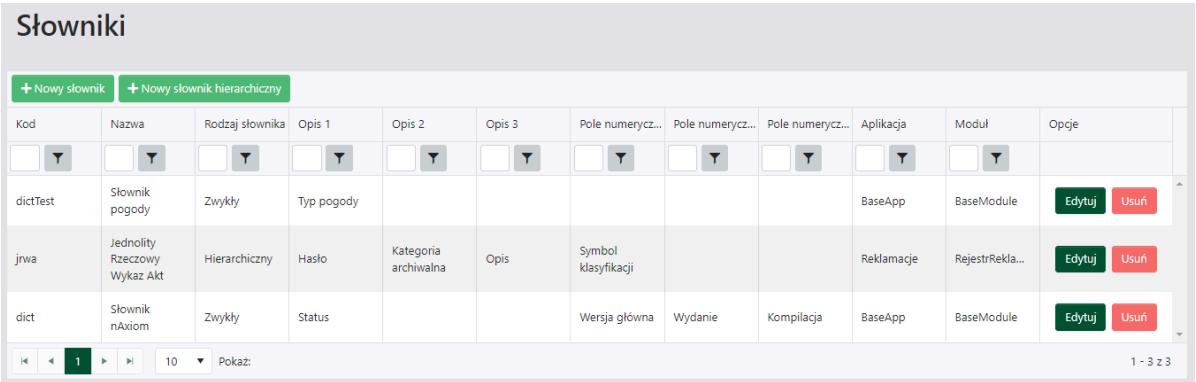
☛ *AdminSPA* > `DANE` > `Słowniki`

W TYM TEMACIE

- Nowa definicja słownika
- Edycja definicji słownika
- Usunięcie słownika
- Okno dialogowe `Nowy słownik/Edycja słownika`

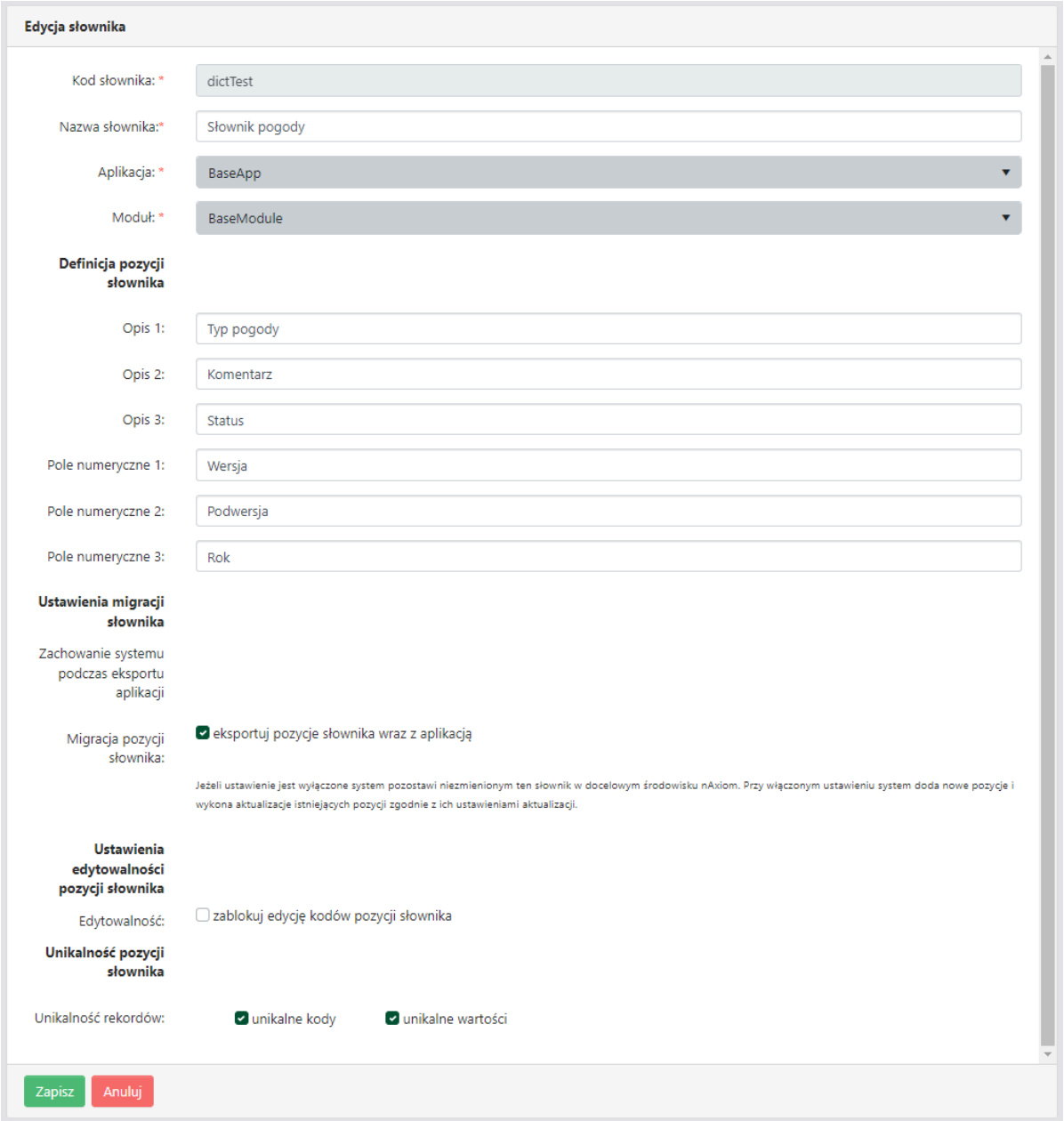
NOWA DEFINICJA SŁOWNIKA

1. Kliknij polecenie `Słowniki` w sekcji `DANE` w menu głównym.
   Zostanie wyświetlone okno `Słowniki`.



Okno `Słowniki`

2. Kliknij przycisk `Nowy słownik` z lewej strony u góry.
   Zostanie wyświetlone okno dialogowe `Nowy słownik`.



Okno dialogowe `Edycja słownika`

3. Wpisz wartości w polach tego okna dialogowego zgodnie z poniższym opisem i kliknij przycisk `Zapisz`.
   Definicja słownika zostanie zapisana.

EDYCJA DEFINICJI SŁOWNIKA

1. Kliknij polecenie `Słowniki` w sekcji `DANE` w menu głównym.
   Zostanie wyświetlone okno `Słowniki`.
2. Kliknij przycisk `Edycja` w wierszu definicji słownika, którą chcesz zmienić.
   Zostanie wyświetlone okno dialogowe `Edycja słownika`.
3. Zmień wartości w polach tego okna dialogowego zgodnie z poniższym opisem i

kliknij przycisk `Zapisz`.
Definicja słownika zostanie zapisana.

USUNIĘCIE SŁOWNIKA

Usunięcie definicji słownika jest możliwe tylko wtedy, kiedy nie jest on przypisany do żadnej kolumny tabeli ani do listy typu słownik na formularzu.

1. Kliknij polecenie `Słowniki` w sekcji `DANE` w menu głównym.
   Zostanie wyświetlone okno `Słowniki`.
2. Kliknij przycisk `Usuń` w wierszu definicji słownika, który chcesz usunąć.
   Zostanie wyświetlony komunikat potwierdzenia.
3. Kliknij przycisk `OK`.
   Słownik zostanie usunięty.

> **WSKAZÓWKA**
> Usunięcie definicji słownika powoduje także usunięcie pozycji tego słownika z tabeli *core.DictionaryValues*.

OKNO DIALOGOWE `NOWY SŁOWNIK/EDYCJA SŁOWNIKA`

`Kod słownika`: wpisz kod słownika; kod słownika jest używany na liście rozwijanej podczas przypisywania słownika do kolumny tabeli w oknie `Kreator modeli danych`; patrz Kreator modeli danych, nowe pole modelu danych .

`Nazwa słownika`: wpisz opisową nazwę słownika.

`Aplikacja`: wybierz aplikację, do której zostanie przypisany słownik (słownik będzie dostępny we wszystkich aplikacjach, ale będzie należał do pakietu wybranej aplikacji i z nią będzie eksportowany).

`Moduł`: wybierz moduł aplikacji.

`Opis 1/2/3`: opcjonalnie możesz zdefiniować maksymalnie trzy dodatkowe kolumny z wartościami tekstowymi na potrzeby opisu pozycji słownika; kolumna opisu będzie wyświetlana tylko wtedy, gdy zostanie zdefiniowana jej nazwa w tym polu.

`Pole numeryczne 1/2/3`: opcjonalnie możesz zdefiniować maksymalnie trzy dodatkowe kolumny z wartościami numerycznymi do opisu pozycji słownika; kolumna numeryczna będzie wyświetlana tylko wtedy, gdy zostanie zdefiniowana jej nazwa w tym polu.

`Eksportuj pozycje słownika wraz z aplikacją`: pozwala określić, czy dany słownik ma być migrowany do środowiska docelowego, co może spowodować zmiany pozycji słownika w środowisku docelowym; aby migrować słownik, ale zabezpieczyć poszczególne pozycje przed nadpisywaniem/usuwaniem, należy włączyć ochronę pozycji ikoną kłódki dostępną na stronie pozycji słownika w aplikacji *FrontSPA*.

`Unikalne kody`: zaznacz to pole wyboru, aby dla słownika nie można było dodać dwóch pozycji o takim samym kodzie („kod" jest systemowym atrybutem pozycji słownika — kolumna *Code* w tabeli *core.DictionaryValues*).

`Unikalne wartości`: zaznacz to pole wyboru, aby dla słownika nie można było dodać dwóch pozycji o takiej samej wartości („wartość" jest systemowym atrybutem pozycji słownika — kolumna *Name* w tabeli *core.DictionaryValues*). button-cancel: |- `Anuluj`: kliknij, aby zamknąć okno i porzucić zmiany.

`Zablokuj edycję kodów pozycji słownika`: po włączeniu blokuje możliwość zmiany kodu pozycji słownika w aplikacji *FrontSPA*; ma to zastosowanie w przypadku, gdy kody konkretnych pozycji słownika są używane w logice aplikacji biznesowej.

`Zapisz`: kliknij, aby zapisać definicję słownika.

`Anuluj`: kliknij, aby porzucić zmiany i zamknąć tryb edycji inline.

Powiązane tematy:

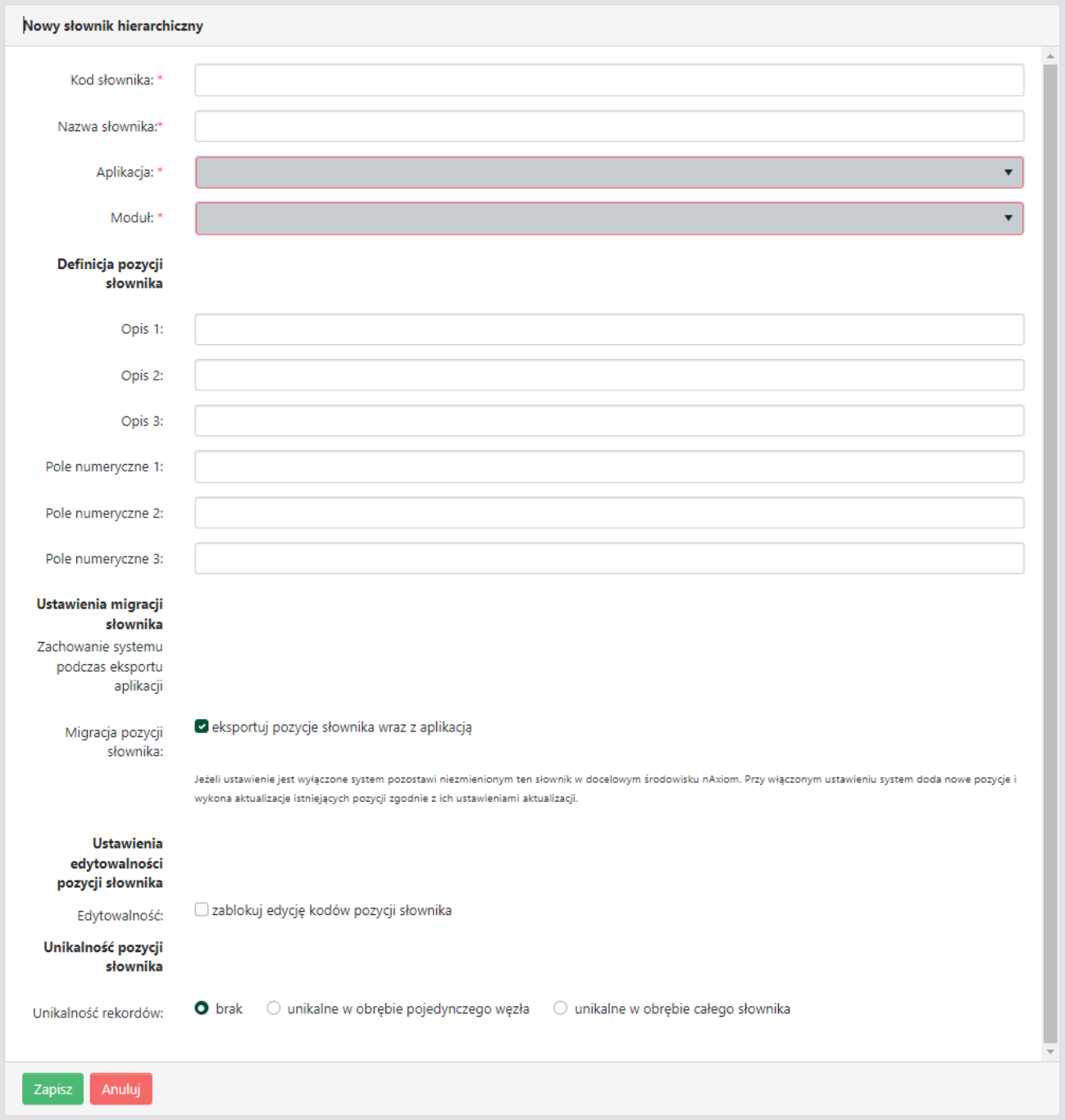
- Lista wyboru (słownik)
- Obsługa słowników

3.5.2. Słowniki hierarchiczne

Podsumowanie:
Słowniki hierarchiczne to listy wartości tekstowych powiązanych ze sobą relacją pozycja nadrzędna — pozycje podrzędne. Taka hierarchia może mieć dowolną liczbę poziomów. Przykładem zastosowania słownika hierarchicznego może być Jednolity Rzeczowy Wykaz Akt, system klasyfikacji akt stosowany w obiegu dokumentów urzędowych. Słowniki hierarchiczne tworzy się, edytuje i usuwa tak samo jak słowniki proste.

☛ *AdminSPA* > `DANE` > `Słowniki`

Aby utworzyć nową definicję słownika hierarchicznego, w oknie `Słowniki` kliknij przycisk `Nowy słownik hierarchiczny`. Zostanie wyświetlone okno dialogowe `Nowy słownik hierarchiczny`.



Okno dialogowe `Nowy słownik hierarchiczny`

Definicja słownika hierarchicznego różni sie od definicji słownika prostego tylko opcjami dotyczącymi powtarzalności pozycji słownika:

`Unikalność rekordów`: określa sposób walidacji pozycji słownika hierarchicznego; dostępne wartości:

- `brak`: (ustawienie domyślne): pozycje słownika mogą się powtarzać;

- `unikalne w obrębie pojedynczego węzła`: pozycje podrzędne określonej pozycji nadrzędnej nie mogą się powtarzać;
- `unikalne w obrębie całego słownika`: pozycje słownika nie mogą się powtarzać.

Powiązane tematy:

- Lista wyboru (słownik hierarchiczny)
- Obsługa słowników

4. INTERFEJS UŻYTKOWNIKA

Podstawowe elementy interfejsu użytkownika w nAxiom to menu użytkownika, formularze i listy. Polecenia menu służą do wyświetlania określonych list i/lub formularzy, na których można przeglądać dane dokumentów biznesowych, rejestrować nowe dokumenty i modyfikować dane dokumentów istniejących. Dodatkowo do tej kategorii należy zaliczyć *widżety* — obiekty, które można wyświetlać na pulpicie startowym aplikacji nAxiom, oraz *kontrolki* — predefiniowane listy rozwijane, które można osadzać w formularzach.

W aplikacji *AdminSPA* polecenia związane z tym obszarem projektowania zgrupowano w sekcji `INTERFEJS UŻYTKOWNIKA`. W kolejnych tematach opisano poszczególne polecenia tego menu.

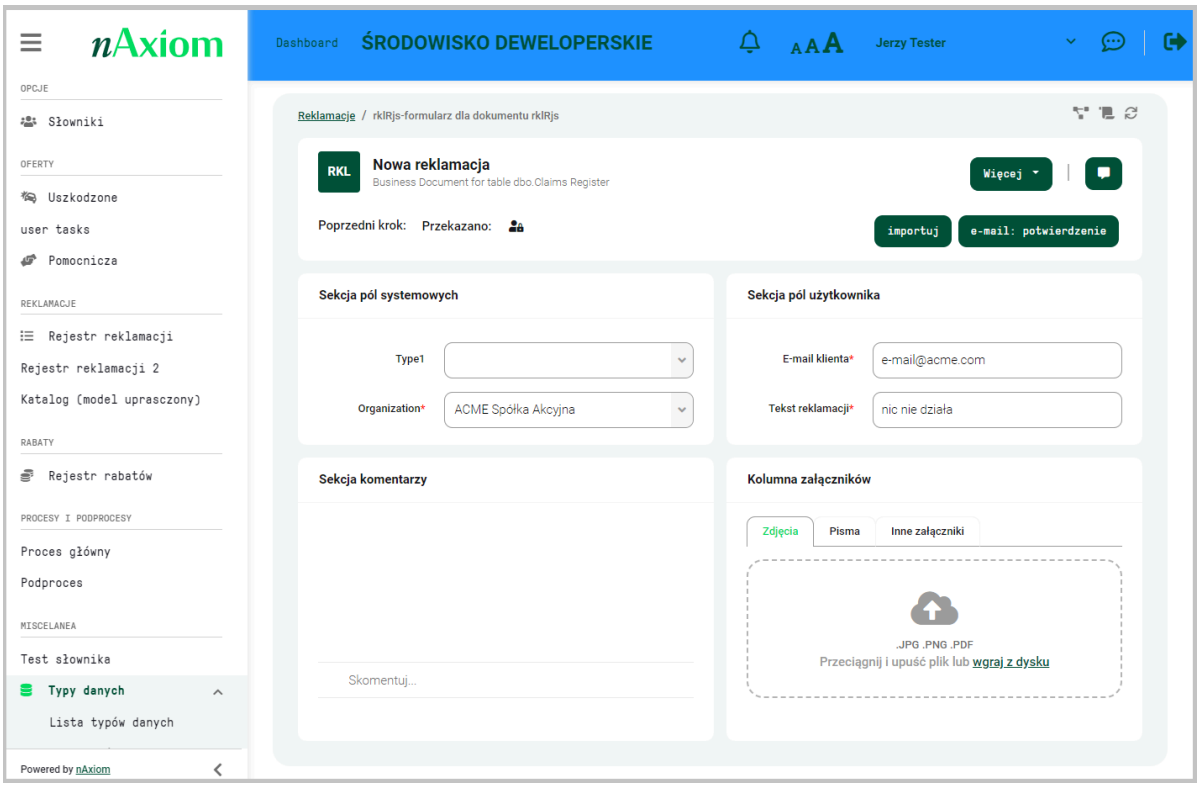
4.1. FORMULARZE

Podsumowanie:
Formularze to obok widoku listy podstawowe elementy interfejsu użytkownika w nAxiom; o ile lista prezentuje zestaw rekordów (dokumentów z tabeli), formularz służy do prezentacji pojedynczych rekordów i ich modyfikacji, a także do prezentacji danych powiązanych z bieżącym dokumentem. Jeśli dokumentem głównym jest zamówienie, w formularzu można wyświetlić dane zamówienia, np. numer, datę, zamawiającego, oraz indywidualne pozycje tego zamówienia; formularze mogą również zawierać elementy sterujące działaniem aplikacji.

☛ *AdminSPA* > `INTERFEJS UŻYTKOWNIKA` > `Formularze`

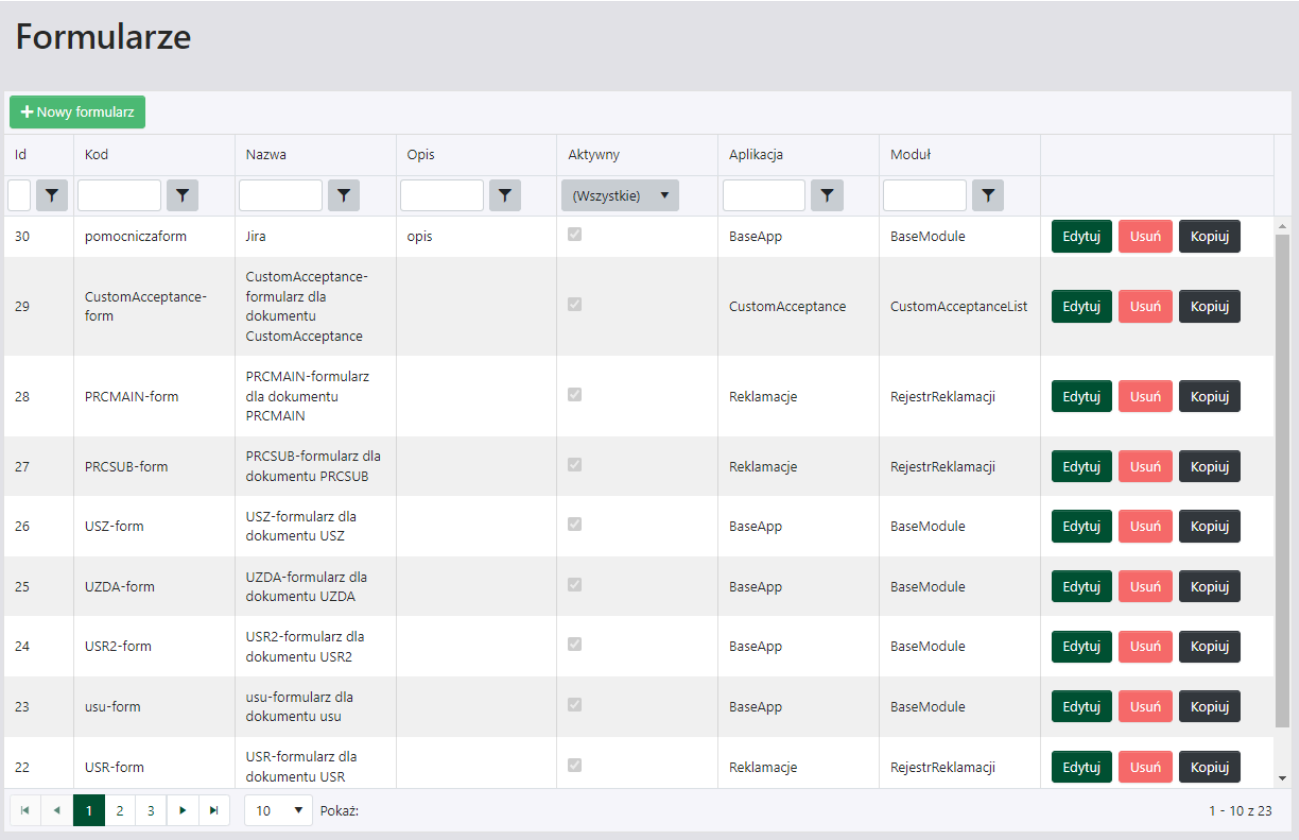
Na poniższej ilustracji przedstawiono przykładowy, prosty formularz widoczny po stronie front-office dla użytkownika aplikacji. W układzie formularza można wyróżnić kilka obszarów. Są to:

- pasek narzędzi (*toolbar*): domyślnie zawiera numer dokumentu, przyciski do zapisu i zmiany statusu.
- `Sekcja pól systemowych`: sekcja grupująca dwa pola systemowe `Type1` i `Organization`.
- `Sekcja pól użytkownika`: sekcja, w której zgrupowano pola powiązane z kolumnami tabeli zdefiniowanymi przez użytkownika.
- `Sekcja komentarzy`: sekcja w której użytkownicy mogą wpisywać komentarze na różnych etapach przetwarzania danych.
- `Kolumna załączników`: obszar, w którym można dodawać załączniki do dokumentu, a także wyświetlać ich podgląd.



Widok formularza w aplikacji FrontSPA

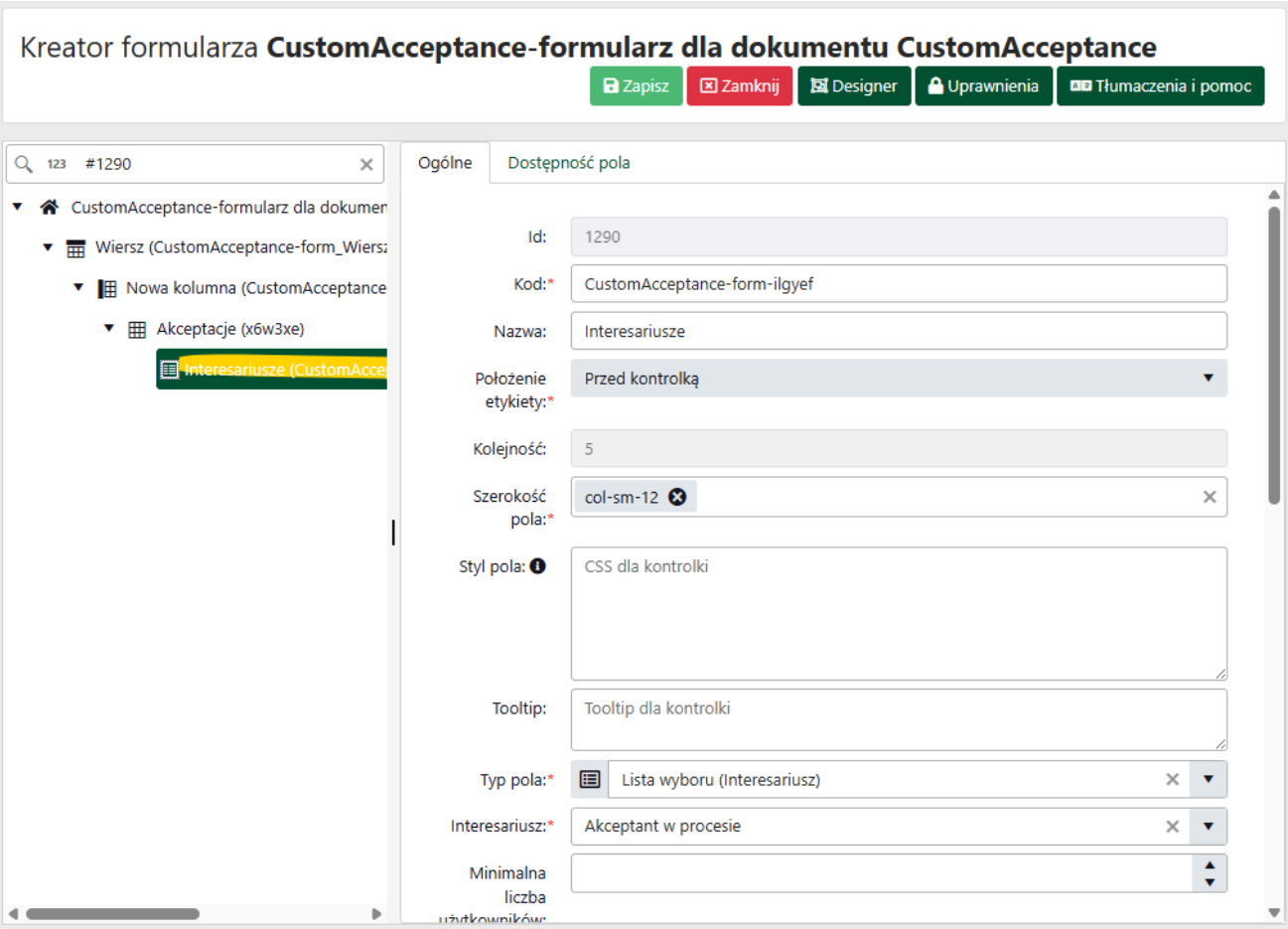
Domyślne tytuły sekcji, ich liczbę i rozmieszczenie można zmienić w trybie projektowania. Aby przejść do modyfikacji formularza, zaloguj się w aplikacji *AdminSPA* (*https:\\adres_platformy_naxiom\admin*) i kliknij polecenie `Formularze` w sekcji `INTERFEJS UŻYTKOWNIKA` w menu głównym. Zostanie wyświetlone okno `Formularze`.



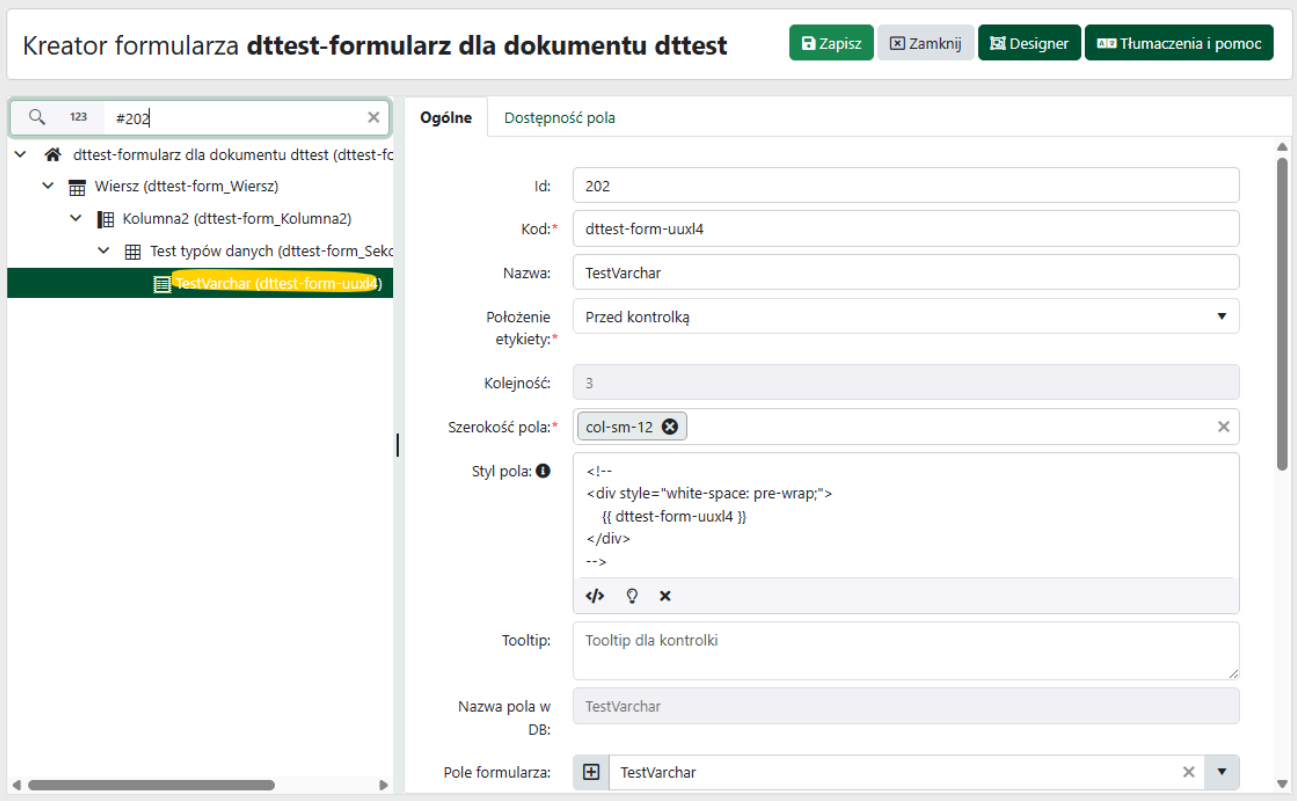
Lista formularzy

Kliknij przycisk `Edycja` obok formularza z listy, aby przejść do edycji tego formularza, lub kliknij przycisk `Nowy formularz`, aby rozpocząć tworzenie formularza. Widok formularza ma dwa tryby edycji. Wyświetlany jako pierwszy *tryb kreatora* jest podobny do trybu edycji widoku listy. Z lewej strony znajduje sie komponent z hierarchiczną listą elementów formularza. Na górze okienko wyszukiwania, w którym można wyszukiwać elementy formularza według nazwy, kodu lub identyfikatora. W tym ostatnim przypadku numer nalezy poprzedzić symbolem # (hash).

Z prawej strony, na kilku kartach wyświetlane są właściwości zaznaczonego elementu. W tym widoku nie ma podglądu formularza.



Kreator formularza



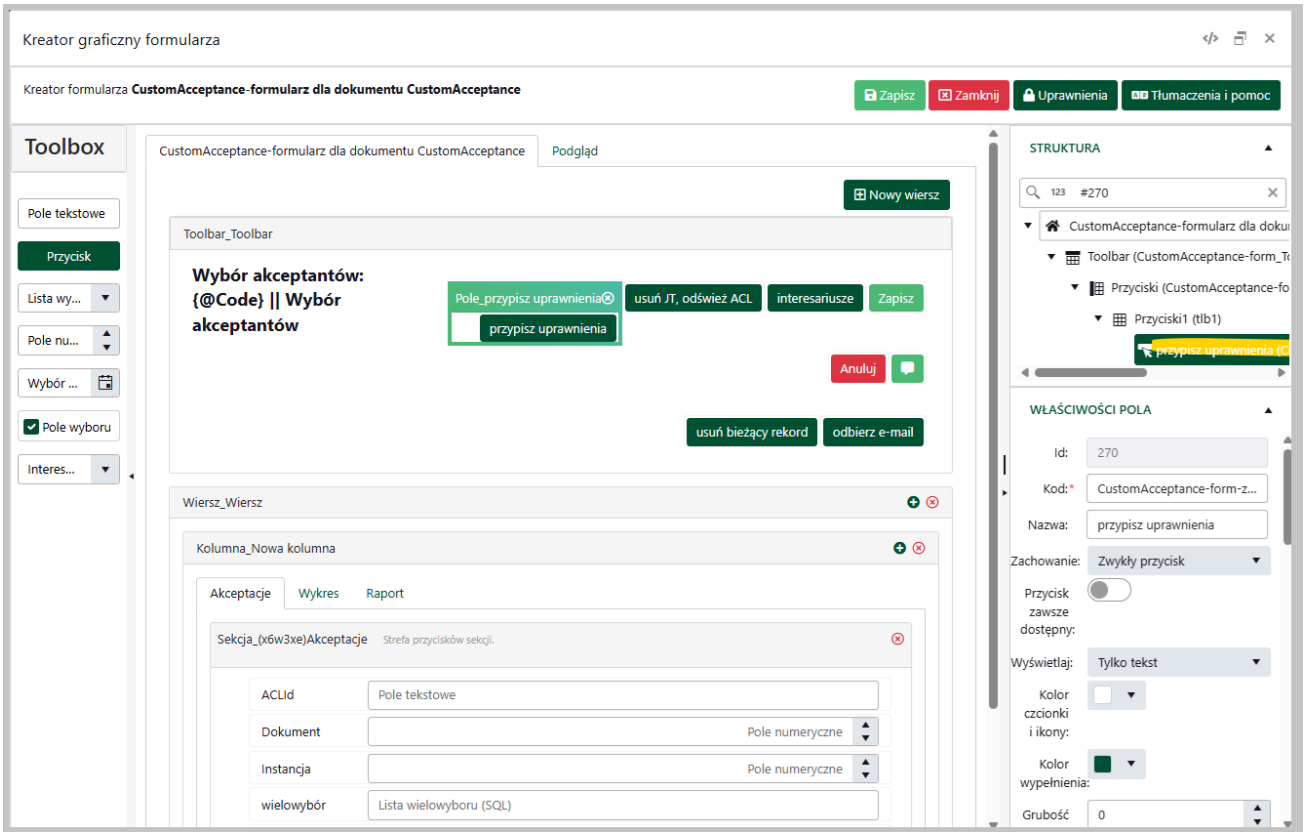
Kreator formularza

Drugi tryb edycji — *tryb designera* — jest wyświetlany po kliknięciu przycisku `Designer` z prawej strony u góry. W tym trybie taki sam komponent struktury elementów formularza znajduje się z prawej strony. Panel centralny ma dwie karty: kartę projektu formularza w widocznym układem kontrolek oraz kartę `Podgląd`, na której można zobaczyć podgląd formularza.

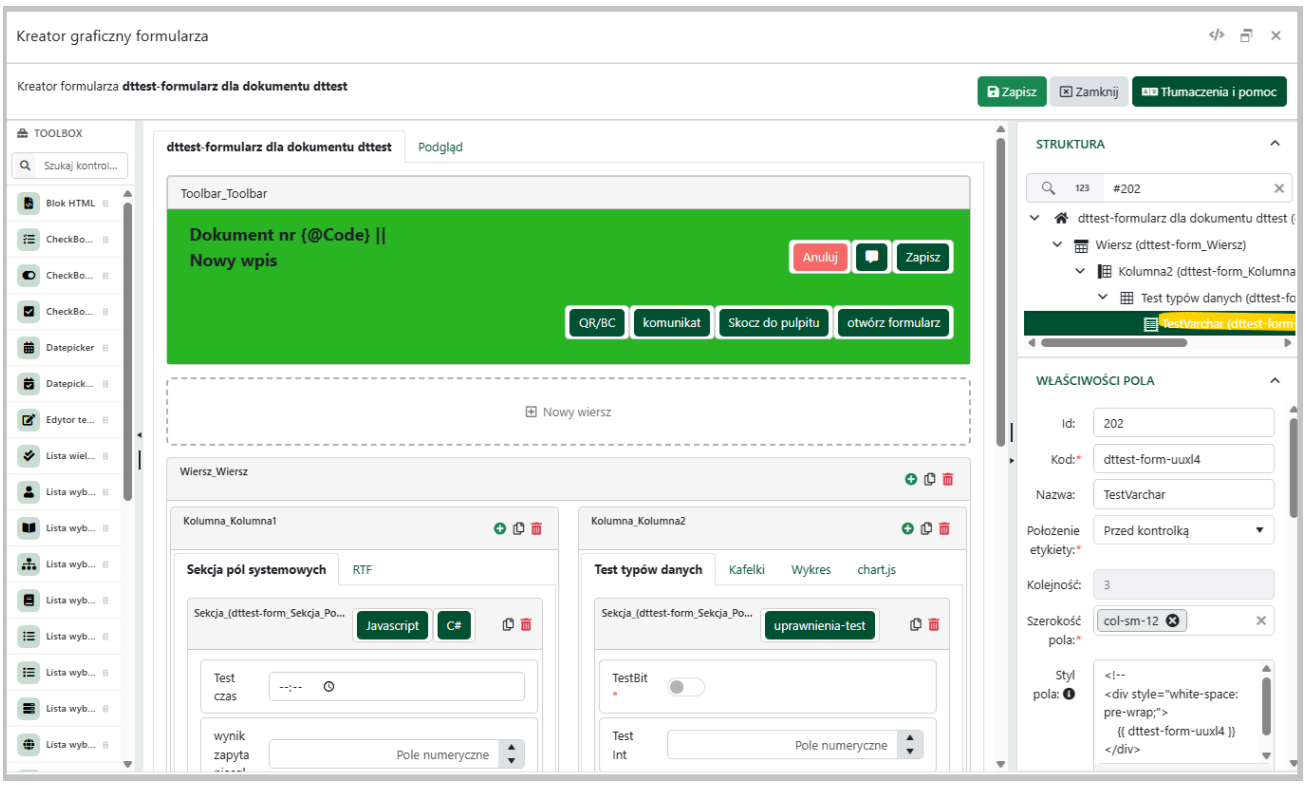
> **WSKAZÓWKA**
> Zmiany formularza są uwzględniane na podglądzie po ich zapisaniu.

Z lewej strony znajduje się panel `Toolbox` z podstawowymi kontrolkami (`Pole tekstowe`, `Przycisk`, `Lista wyboru`, `Pole numeryczne`, `Wybór daty`, `Pole wyboru`, `Interesariusz`). Panel z prawej strony jest dzielony między komponent struktury i właściwości wybranego elementu formularza.

Z lewej strony znajduje się panel `Toolbox` z pełną listą kontrolek, których można użyć na formularzu, oraz polem wyszukiwania. Panel z prawej strony jest dzielony między komponent struktury i właściwości wybranego elementu formularza.



Kreator graficzny formularza — designer



Kreator graficzny formularza — designer

W trybie designera widać jak wygląda wizualny układ formularza. Obejmuje on następujące elementy:

- **Toolbar**: pasek narzędzi z przyciskami (domyślnie `Zapisz` i `Anuluj`); na pasku narzędzi jest także wyświetlana nazwa dokumentu, jego numer i status.
- **Wiersz**: „poziomy” element układu graficznego formularza; zawiera kolumny.
- **Kolumna**: „pionowy” element układu graficznego formularza; zawiera sekcje.
- **Sekcja**: element logiczny, grupujący pola formularza.

> **UWAGA**
> Pola formularza to elementy logiczne powiązane z kolumnami źródła danych formularza. W interfejsie użytkownika pola są reprezentowane przez *kontrolki* — obiekty skonstruowane z myślą o wprowadzaniu danych określonego typu, np. *datepicker* wyświetlający kalendarz w celu wyboru daty, albo lista, która umożliwia wybór predefiniowanej wartości zamiast wpisywania jej. W dalszej części terminy *pole formularza* i *kontrolka* mogą być, w uproszczeniu, stosowane zamiennie.

W obu trybach można modyfikować formularz, korzystając z tych samych narzędzi, jednak w trybie designera dostępne są tylko wybrane właściwości formularza (z karty `Ogólne` w trybie kreatora). Komplet właściwości formularza jest dostępny w trybie kreatora.

W kolejnych tematach tego rozdziału opisano właściwości formularza i jego poszczególnych elementów.

4.1.1. Właściwości formularza

Podsumowanie:
Najważniejsze właściwości formularza to definicja dokumentu biznesowego, z którą formularz jest powiązany oraz źródło danych definiowane zapytaniem SQL. Ponadto dla formularza można zdefiniować szereg akcji, określić warunki ukrywania formularza bądź przełączania go w tryb tylko do odczytu, a także skonfigurować szereg innych ustawień.

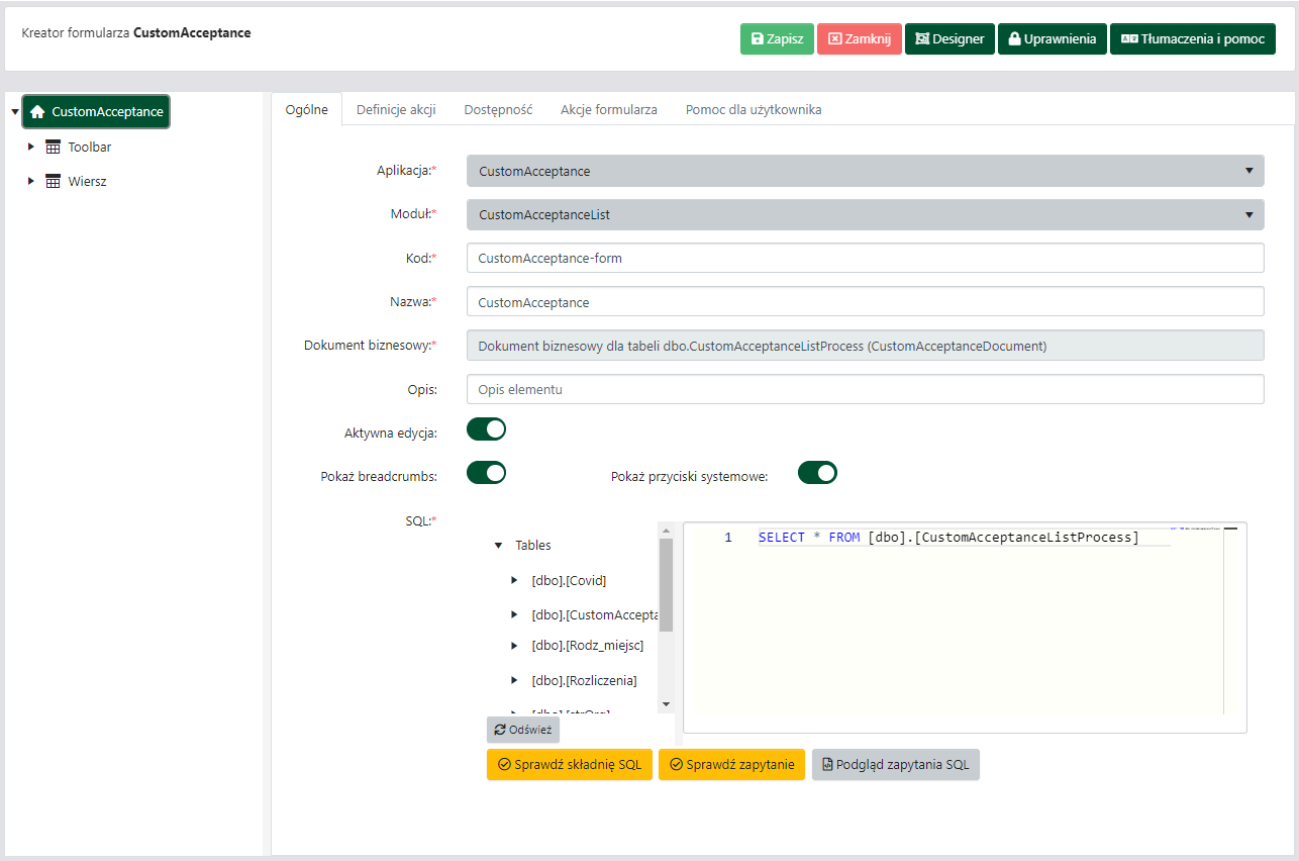
Właściwości formularza zgrupowano na pięciu kartach widocznych po przejściu do edycji formularza w trybie kreatora formularza (w trybie designera dostępne są tylko wybrane właściwości formularza).

☛ *AdminSPA* > `INTERFEJS UŻYTKOWNIKA` > `Formularze` > `Edytuj`

W TYM TEMACIE

- Karta Ogólne
- Karta Pola techniczne
- Karta Definicje akcji
- Karta Dostępność
- Karta Akcje formularza
- Karta Pomoc dla użytkownika

KARTA OGÓLNE



Właściwości formularza — karta `Ogólne`

- `Aplikacja`: nazwa aplikacji, do której ma być przypisany formularz.
- `Moduł`: nazwa modułu, do którego ma być przypisany formularz.
- `Kod`: ciąg znaków jednoznacznie identyfikujący formularz; obsługiwane sa tylko litery i cyfry.
- `Nazwa`: nazwa formularza; nadawana automatycznie w procesie generowania elementów aplikacji; ma postać *{przedrostek}-formularz dla dokumentu {przedrostek}*, gdzie *{przedrostek}* to ciąg znaków podany w polu `Przedrostek dla generowanych elementów` w oknie `Generowanie nowego flow...` (patrz Kreator modeli danych, okno Generowanie elementów aplikacji).
- `Opis`: opis formularza.
- `Aktywna edycja`: pozwala wyłączyć możliwość edytowania wartości w polach na formularzu oraz dezaktywować przyciski; użytkownik może dodawać komentarze do dokumentu oraz dodawać i usuwać załączniki; tak samo działa formularz z ustawionym parametrem `isReadOnly` = 1 na karcie `Dostępność` (parametr `isReadOnly` ma pierwszeństwo przed ustawieniem przełącznika `Aktywna edycja`).
- `Pokaż breadcrumbs`: po włączeniu wyświetla u góry ścieżkę (*nazwa aplikacji* lub *opis aplikacji/id formularza* lub *kod formularza* lub *nazwa formularza*) definiowaną w ustawieniach systemu; patrz Ustawienia systemy, Breadcrumbs .

- `Pokaż przyciski systemowe`: włącza lub wyłacza wyświetlanie w prawym górnym rogu ikon:
    - kontekst formularza,
    - diagram procesu,
    - odśwież formularz.
- `Dokument biznesowy`: definicja dokumentu, z którą jest powiązany formularz; to powiązanie jest niezbędne do prawidłowego działania modelu procesu (*workflow*); w trybie edycji formularza pole tylko do odczytu; podczas tworzenia nowego formularza lista rozwijana, z której można wybrać definicję dokumentu biznesowego, dodatkowo jest dostępny przycisk `Nowy`, który wyświetla okno tworzenia nowej definicji dokumentu biznesowego.
- `SQL`: zapytanie SQL definiujące źródło danych dla formularza; pod oknem z instrukcją SQL znajdują się przyciski:
    - `Sprawdź składnię SQL`: sprawdza poprawność instrukcji SQL pod względem składniowym.
    - `Sprawdź zapytanie`: sprawdza możliwość wykonania zapytania (np. nazwy obiektów bazy danych).
    - `Podgląd zapytania SQL`: otwiera okno dialogowe, w którym można zobaczyć dostępne parametry, ustawić ich przykładową wartość i sprawdzić poprawność zapytania dla tych wartości; patrz Podgląd/wykonanie zapytania SQL .

KARTA POLA TECHNICZNE

Na karcie `Pola techniczne` można dodawać pola niepowiązane z polami modelu danych. Pola techniczne nie są wyświetlane na formularzu. Są one tylko dostępne w kontekście formularza i można się do nich odwoływać poprzez zmienne typu *{@NazwaPola}*.

Dla pola technicznego należy podać następujące atrybuty:

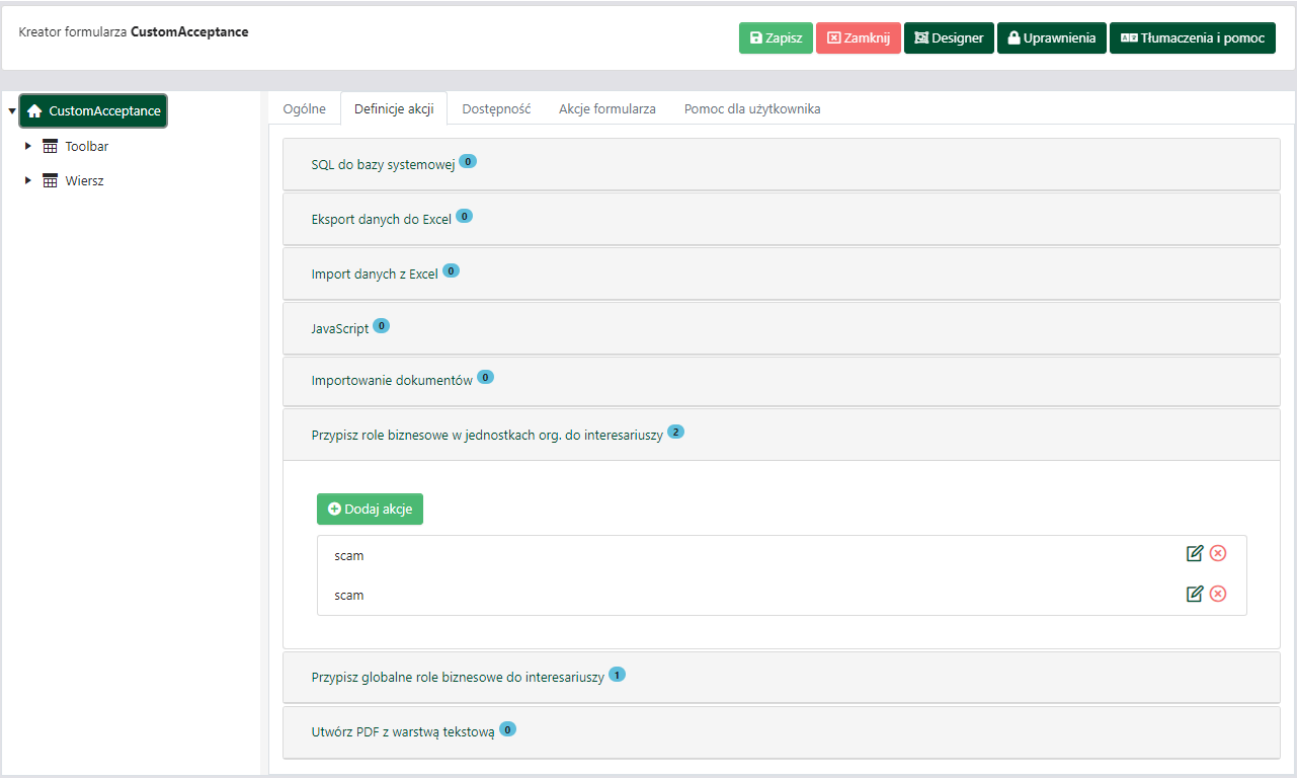
- `Nazwa pola`
- `Typ danych JSON`: dostępne wartosci *string*, *number* i *boolean*
- `Wartość domyślna dla tworzenia rekordu`: (opcjonalnie) wartość domyślna pola dla nowego rekordu
- `Wartość dla istniejącego rekordu`: (opcjonalnie) wartość domyślna pola dla istniejącego rekordu

Wartości w polach technicznych można zmieniać, używając wyrażeń Javascript (Skrypt dla zmiany kontekstu formularza). Odwołania do pól technicznych w tym przypadku są takie same jak dla innych pól formularza: *formContext.nazwaPola*.

WAŻNE

- Wartości pól technicznych są ustawiane dopiero po przeliczaniu dostępności sekcji/pól, dlatego nie należy ich używać w warunkach SQL dotyczących dostępności.
- Należy pamiętać, że użycie wartości pola z atrybutem *Dostęp chroniony* jako parametru w obliczaniu wartości pola technicznego może prowadzić do błędu w sytuacji, gdy wartość pola chronionego nie będzie dostępna w kontekście.

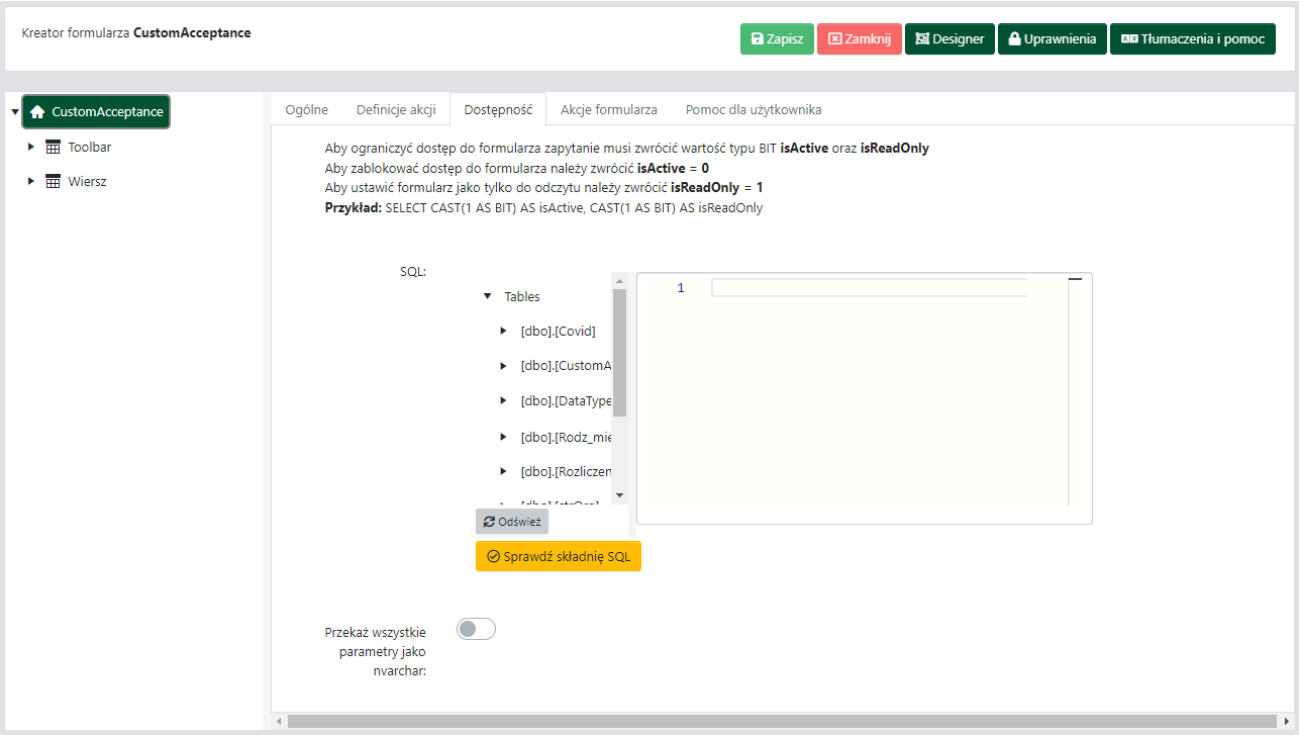
KARTA DEFINICJE AKCJI



Właściwości formularza — karta `Definicje akcji`

Karta `Definicje akcji` zawiera wykaz akcji przypisanych do formularza z podziałem na kategorie akcji. W ramach każdej kategorii można dodawać nowe akcje oraz modyfikować i usuwać akcje już zdefiniowane. Wyzwalaczami akcji na formularzu mogą być przyciski oraz zdarzenia kontrolek formularza: `OnChange`, `OnLeave`, `OnClick`. Sposób definiowania akcji różnych kategorii opisano w temacie Akcje w nAxiom

KARTA DOSTĘPNOŚĆ



Właściwości formularza — karta `Dostępność`

Karta `Dostępność` pozwala ograniczyć dostęp do samego formularza lub do pól i przycisków na formularzu. W tym celu należy skonstruować zapytanie SQL, które zwróci dwa parametry: `isActive` i `isReadOnly`.

- `SQL`: okno do wpisania zapytania SQL definiującego atrybuty dostępności.
- `Sprawdź składnię SQL`: sprawdza poprawności składni zapytania.
- `Przekaż wszystkie parametry jako nvarchar`: przełącznik zapewniający kompatybilność wsteczną z wcześniejszymi wersjami, kiedy wartości zmiennych typu *{@nazwa}* (z kontekstu formularza i systemowe) w zapytaniu SQL były domyślnie przekazywane jako tekst (typ *nvarchar*); od wersji 1.4.1.3 parametry w zapytaniach SQL są przekazywane zgodnie z ich typem natywnym.

Tabelka przedstawia dostępność formularza i jego pól/przycisków w zależności od kombinacji wartości obu parametrów.

isActive	isReadOnly	Zachowanie formularza
1	0	Formularz działa normalnie, można dodawać i edytować dokumenty
1	1	Pola i przyciski na formularzu są nieaktywne, można dodawać komentarze i załączniki; takie samo zachowanie, jak po wyłączeniu przełącznika `Edycja aktywna`.
0	1/0	Próba wyświetlenia formularza (przycisk `Nowy dokument` lub `Edytuj`) powoduje wyświetlenie komunikatu `403 – Nie masz dostępu do tej strony`

WSKAZÓWKA

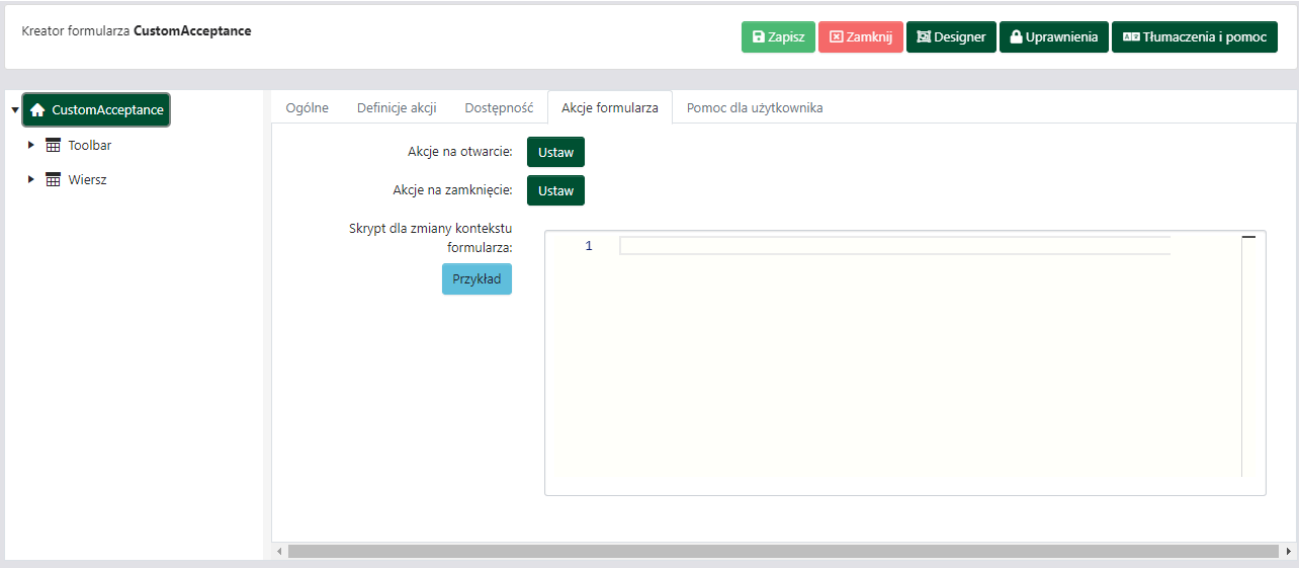
ustawienia dostępności za pomocą zapytania SQL mają pierwszeństwo przed ustawieniem przełącznika `Edycja aktywna` na karcie `Ogólne`.

Poniżej przedstawiono przykład, który będzie ustawiał różne kombinacje obu parametrów w zależności od zalogowanego użytkownika. W przykładzie użyto następujących zmiennych:

- `{@UserId}`: wartość pola `UserId` na formularzu.
- `{@_UserId}`: identyfikator zalogowanego użytkownika.

```
SELECT
CAST(
  CASE
-- Jeśli zaloguje się konkretny użytkownik, dezaktywuj formularz
    WHEN {@_UserId} = 'E8F68D83-7905-4A5E-CDCA-08D9326197D4'
    THEN 0
    ELSE 1
  END
AS BIT) AS isActive,
CAST(
  CASE
-- Jeśli zaloguje się twórca dokumentu, pozwól na modyfikacje,
-- w przeciwnym razie, ustaw tryb tylko do odczytu
    WHEN {@_UserId} = {@UserId}
    THEN 0
    ELSE 1
  END
AS BIT) AS isReadOnly
```

KARTA AKCJE FORMULARZA



Właściwości formularza — karta `Akcje formularza`

Na karcie `Akcje formularza` można dodawać akcje, które będą blokować i odblokowywać możliwość edycji danych w formularzu w celu uniknięcia konfliktów edycji. Ponadto na tej karcie można zdefiniować skrypt do zmiany kontekstu formularza.

- `Akcje na otwarcie/zamknięcie`: kliknięcie przycisku `Ustaw` wyświetla okno wyboru akcji (patrz Przycisk). Po kliknięciu przycisku `Wybierz akcje z listy` wyświetlane są tylko dwie akcje systemowe:
  - `Blokuj dokument`: działa w ten sposob, że otwarcie dokumentu do edycji przez jednego użytkownika powoduje zablokowanie możliwości jego edycji przez innego użytkownika na określony czas (domyślnie 60 s). Po upływie tego czasu odmierzany jest czas (domyślnie 4 min), w którym użytkownik próbujący uzyskać dostęp do edytowanego dokumentu może zdecydować o zignorowaniu blokady. Po upływie tego okresu (łącznie 300 s od zablokowania dokumentu) blokada jest wyłączana.
  - `Odblokuj dokument`: powoduje natychmiastowe przerwanie blokady dokumentu. Informacje dotyczące blokady są zapisywane w kolumnach systemowych tabeli modelu danych `Blocked`, `BlockUser` i `LastEntryDate`. Czasy blokady ustawia się w definicji dokumentu biznesowego na karcie `Blokady`.
- `Skrypt dla zmiany kontekstu formularza`: pole do wpisania skryptu JavaScript służącego do sterowania wartościami i atrybutami pól formularza w zależności od zmian kontekstu formularza; przycisk `Przykład` wyświetla okno z licznymi przykładami skryptów, które przytoczono poniżej.

Przykłady:

Opcje ustawień dostępności pól poprzez skrypt:

Dostęp do pola - odbywa się z wykorzystaniem funkcji `getFieldByCode('unikalny-kod-pola')`

```
const pole1 = getFieldByCode('unikalny-kod-pola');
```

Ustawienia widoczności - widoczność ustawiana poprzez właściwość pola `isActive`

```
pole1.isActive=true;   //pole widoczne
pole1.isActive=false;  //pole niewidoczne
```

Ustawienia edytowalności - edytowalność ustawiana poprzez właściwość pola `isDisabled`

```
pole1.isDisabled=true;   //pole tylko do odczytu
pole1.isDisabled=false;  //pole edytowalne
```

Ustawienia wymagalności - wymagalność ustawiana poprzez właściwość pola `isRequired`

```
pole1.isRequired=true;   //pole wymagane
pole1.isRequired=false;  //pole niewymagane
```

Ustawienia widoczności pola w wersji mobilnej - widoczność ustawiana poprzez właściwość pola `isActiveOnMobile`

```
pole1.isActiveOnMobile=true;   //pole widoczne w wersji mobilnej
pole1.isActiveOnMobile=false;  //pole niewidoczne w wersji mobilnej
```

Ustawienia dostępności pola w przeglądarce nawet gdy niewidoczne - dostępność pola w przeglądarce pomino ewentualnego braku widoczności ustawiana poprzez właściwość pola `sendHidden`

```
pole1.sendHidden=true;   //pole dostępne w przeglądarce
pole1.sendHidden=false;  //pole niedostępne w przeglądarce
```

Dostęp do sekcji - odbywa się z wykorzystaniem funkcji `getSectionByCode('unikalny-kod-sekcji')`

```
const sekcja1 = getSectionByCode('unikalny-kod-sekcji');
```

Ustawienia widoczności sekcji - widoczność ustawiana poprzez właściwość `isActive`

```
sekcja1.isActive=true;   //sekcja widoczna
sekcja1.isActive=false;  //sekcja niewidoczna i nieaktywna
```

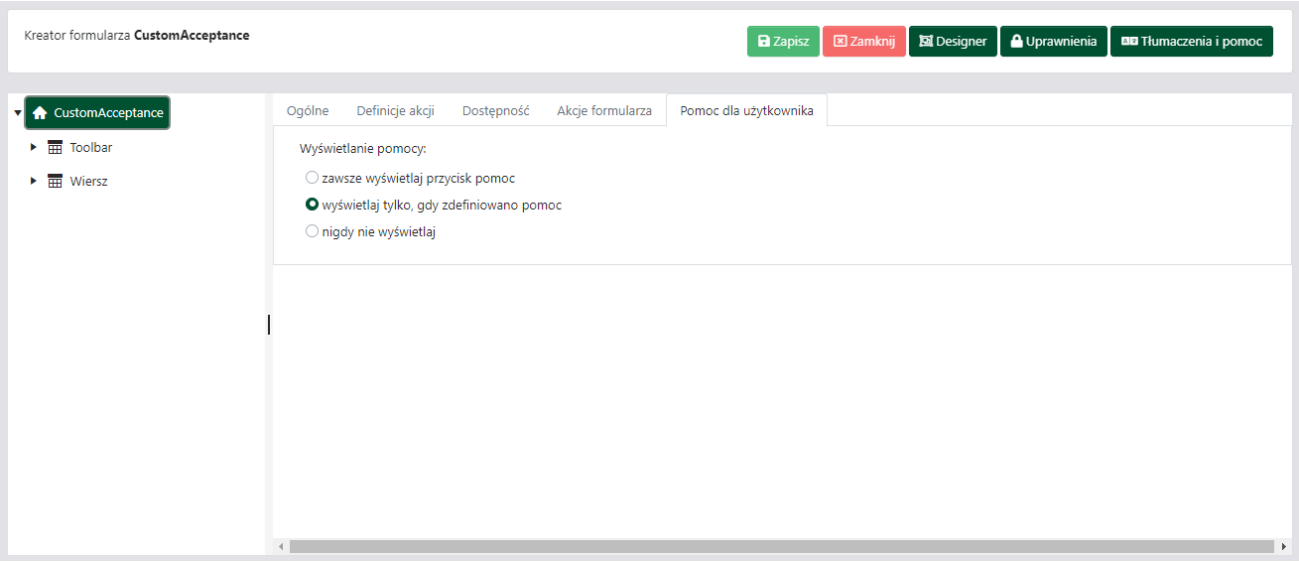
Ukrywanie sekcji - sekcję można ukryć z wykorzystaniem funkcji `setSectionAsHidden('unikalny-kod-sekcji')`

```
setSectionAsHidden('unikalny-kod-sekcji');
// sekcja będzie ukryta i jednocześnie parametry domyślne będą obsługiwane,
// funkcja działa jak kombinacja parametrów "Pole widoczne" = OFF
// i "Pole dostępne w przeglądarce nawet gdy niewidoczne" = ON
```

Ustawienia widoczności i edytowalności na podstawie wartości pola `[Organization]`

```
const pole1 = getFieldByCode('S1P-form-053Z83');

if (formContext.Organization == 1) {
  pole1.isActive=true;
  pole1.isDisabled=true;
}
else {
  pole1.isActive=false;
}
```

KARTA POMOC DLA UŻYTKOWNIKA



Właściwości formularza — karta `Pomoc dla użytkownika`

Karta `Pomoc dla użytkownika` zawiera opcje określające wyświetlanie przycisku pomocy dla formularza:

- `Zawsze wyświetlaj przycisk pomocy`; przycisk pomocy jest wyświetlany na formularzu/liście niezależnie od tego, czy skonfigurowano jego obsługę.
- `Wyświetlaj tylko, gdy zdefiniowano pomoc`: (ustawienie domyślne) przyciski są wyświetlane, jeśli zdefiniowano akcje wywoływane przyciskiem pomocy.
- `Nigdy nie wyświetlaj`: ukrywa przycisk pomocy na liście/formularzu; ustawienie przydatne np. w przypadku formularza do wyświetlania pomocy.

Więcej informacji zawiera rozdział Definiowanie pomocy w aplikacjach biznesowych

4.1.1.1. PODGLĄD ZAPYTANIA SQL

4.1.2. Toolbar

Podsumowanie:
Toolbar to systemowy wiersz formularza, który domyślnie zawiera numer dokumentu oraz przyciski `Zapisz` i `Anuluj`, a także przyciski zdefiniowane w `Workflow`, wyświetlane dynamicznie po zmianie statusu dokumentu. Ponadto, po prawej stronie numeru dokumentu jest wyświetlany bieżący status dokumentu. W wierszu `Toolbar` nie można umieścić żadnych innych kontrolek, oprócz przycisków.

Aby wyświetlić właściwości wiersza `Toolbar` w projekcie formularza, kliknij `INTERFEJS UŻYTKOWNIKA` > `Formularze` > (*NazwaFormularza*) `Edytuj` i kliknij pozycję `Toolbar` w drzewie elementów z lewej strony.

W trybie designera kliknij Toolbar, aby wyświetlić jego właściwości na panelu z prawej strony.

WŁAŚCIWOŚCI WIERSZA `TOOLBAR`

- `Kod`: dowolny ciąg znaków identyfikujący wiersz `Toolbar` na formularzu.
- `Nazwa`: nazwa wiersza `Toolbar`.
- `Tooltip`: dowolny tekst, który będzie wyświetlany po kliknięciu przycisku ℹ (`Info`) w wierszu `Toolbar` po stronie front-office; ten przycisk jest wyświetlany między numerem dokumentu (wartościa pola `Etykieta`) i statusem dokumentu.
- `Kolejność`: (pole tylko do odczytu) numer kolejny kontrolki w bieżącej sekcji; nadawany automatycznie podczas dodawania kontrolki do formularza.
- `Kolor tła`: w tym polu można wpisać nazwę koloru HTML (patrz www.w3schools.com) dla tła danego elementu.
- `Aktywne`: określa, czy Toolbar będzie aktywny na formularzu.
- `Aktywne w wersji mobilnej`: pozwala zdecydować, czy dany element formularza ma być widoczny w wersji mobilnej aplikacji.
- `Dostępność`: kliknięcie przycisku `Otwórz matrycę` otwiera okno dialogowe `Matryca dostępności elementów formularza`, w którym można sterować dostępnością takich elementów formularza jak `Toolbar`, `Wiersz` i `Wiersz z domyślnymi sekcjami`.
- `Przyklej do góry ekranu (tryb desktop)`: po włączeniu powoduje zadokowanie paska narzędzi u góry ekranu, aby był zawsze widoczny podczas przewijania formularza; opcja dotyczy tylko paska narzędzi wyświetlanego w trybie desktop (na większych ekranach).
- `Etykieta`: tekst wyświetlany z lewej strony wiersza `Toolbar`; zwykle służy do wyświetlania nazwy i numeru dokumentu lub samej nazwy w przypadku dodawania dokumentu; kliknięcie przycisku ze znakiem zapytania powoduje wyświetlenie informacji dotyczącej szablonu etykiety; przykład: `Dokument nr {@Code} || Nowy wpis`
  w przypadku pustego formularza (nowy dokument) w wierszu `Toolbar` będzie wyświetlany tekst `Nowy wpis`, w przypadku zapisanego dokumentu wyświetlany będzie na przykład tekst `Dokument nr 0012/2021`, gdzie `0012/2021` to wartość z pola `Code` na formularzu. W tym polu domyślnie zapisywany jest numer kolejnego

dokumentu w formacie zdefiniowanym w menu `Procesy biznesowe` > `Autonumeracja`.

- `Kolor tła akronimu`: określa kolor, na tle którego jest wyswietlany akronim z definicji dokumentu.
- `Wyświetlaj akronim`: decyduje, czy w toolbarze ma być wyświetlany akronim z definicji dokumentu.
- `Wyświetlaj opis`: decyduje, czy w toolbarze ma być wyświetlany opis z definicji dokumentu.
- `Pokazuj osobę`: wyświetla imię i nazwisko użytkownika, który wykonał poprzedni krok w przetwarzaniu dokumentu.
- `Pokazuj rolę`: wyświetla wszystkie role biznesowe użytkownika, który wykonał poprzedni krok w przetwarzaniu dokumentu.
- `Pokazuj datę przejścia`: wyświetla datę wykonania ostatniego kroku w przetwarzaniu dokumentu.
- `Pokazuj termin realizacji`: wyświetla termin realizacji danego kroku, o ile został zdefiniowany na diagramie procesów.

- `Pokazuj właściciela zadania`: wyświetla informacje o właścicielu zadania oraz menu umożliwiające zmianę przypisania dokumentu do właściciela zadania.

> **WSKAZÓWKA**
> Ten przełącznik dotyczy tylko zapisanych rekordów. Na formularzu nowego rekordu żadne elementy GUI dotyczące właściciela zadania nie są wyświetlane.

W elemencie Toolbar można umieścić wiele przycisków do wykonywania różnych akcji w dwóch kontenerach: podstawowym, obok przycisków `Zapisz` i `Anuluj` oraz poniżej tych przycisków. W przypadku, gdy jest za mało miejsca na wyświetlenie wszystkich tych przycisków, zostają one zastąpione tzw. przyciskiem *split-button*. Do przycisku z najmniejszą wartością parametru `Kolejność` w danym kontenerze zostaje dodany z prawej strony przycisk rozwijający listę pozostałych przycisków. Taki przycisk jest widoczny na pierwszej ilustracji powyżej. W ten sam sposób mogą być grupowane przyciski do zmiany statusu dokumentu z lewej strony elementu Toolbar.

4.1.3. Wiersz formularza

> **Podsumowanie:**
> *Wiersz* to element hierarchicznej struktury, którą tworzą kolejno formularz, wiersze, kolumny, sekcje i kontrolki.

Aby utworzyć nowy wiersz na formularzu, kliknij przycisk `Nowy wiersz` na karcie projektu formularza w trybie designera (z prawej strony u góry). W trybie kreatora polecenie `Nowy wiersz` jest dostępne w menu kontekstowym formularza w drzewie (*root*). Aby usunąć wiersz z formularza, kliknij kółko z symbolem `x` z prawej strony paska tytułu wiersza lub wybierz odpowiednie polecenie z menu kontekstowego wiersza w kreatorze formularzy.

Aby utworzyć nowy wiersz na formularzu, kliknij placeholder `Nowy wiersz` na karcie projektu formularza w trybie designera (poniżej nagłówka toolbox). W trybie kreatora polecenie `Nowy wiersz` jest dostępne w menu kontekstowym formularza w drzewie (*root*). Aby usunąć wiersz z formularza, kliknij kółko z symbolem `x` z prawej strony paska tytułu wiersza lub wybierz odpowiednie polecenie z menu kontekstowego wiersza w kreatorze formularzy.

> **UWAGA**
> Wraz z wierszem usuwane są wszystkie elementy, które się w nim znajdują; w razie przypadkowego usunięcia wiersza zamknij formularz bez zapisywania zmian, a następnie otwórz ponownie.

Aby wyświetlić właściwości wiersza, kliknij pasek tytułu wiersza w trybie designera.

Wiersz formularza ma następujące właściwości:

- `Id`: identyfikator wiersza; tylko do odczytu.
- `Kod`: dowolny ciąg znaków identyfikujący wiersz na formularzu.
- `Nazwa`: nazwa wiersza.
- `Opis`: opis wiersza.
- `Kolejność`: pole tylko do odczytu z numerem kolejnym wiersza na formularzu.
- `Kolor tła`: w tym polu można wpisać nazwę koloru HTML (patrz www.w3schools.com) dla tła danego elementu.
- `Aktywne`: służy do dezaktywowania elementu na formularzu; po wyłączeniu tego przełącznika wiersz `Toolbar` będzie niewidoczny.
- `Aktywne w wersji mobilnej`: pozwala zdecydować, czy dany element formularza ma być widoczny w wersji mobilnej aplikacji.

4.1.4. Kolumna formularza

Podsumowanie:
Kolumna to element hierarchicznej struktury, którą tworzą kolejno formularz, wiersze, kolumny, sekcje i kontrolki.

Aby utworzyć nową kolumnę na formularzu, kliknij kółko z symbolem `+` z prawej strony paska tytułu **wiersza** w trybie designera. Aby usunąć kolumnę z formularza, kliknij kółko z symbolem `x` z prawej strony paska tytułu kolumny.

UWAGA
Wraz z kolumną usuwane są wszystkie elementy, które się w niej znajdują; w razie przypadkowego usunięcia kolumny zamknij formularz bez zapisywania zmian, a następnie otwórz ponownie.

Aby wyświetlić właściwości kolumny, kliknij pasek tytułu kolumny w trybie designera.

Kolumna formularza ma następujące właściwości:

- `Id`: identyfikator kolumny; tylko do odczytu.
- `Kod`: dowolny ciąg znaków identyfikujący kolumnę na formularzu.
- `Nazwa`: nazwa kolumny.
- `Opis`: opis kolumny.
- `Kolejność`: pole tylko do odczytu z numerem kolejnym kolumny w wierszu formularza.
- `Typ kolumny`: po kliknięciu w tym polu jest wyświetlane okno dialogowe, w którym można wybrać szerokość kolumny w umownych jednostkach dla różnych rodzajów ekranów.
- `Minimalna wysokość kolumny (px)`: minimalna wysokość kolumny w pikselach w aplikacji *FrontSPA*.
- `Kolor tła`: w tym polu można wpisać nazwę koloru HTML (patrz www.w3schools.com) dla tła danego elementu.
- `Aktywna`: służy do dezaktywowania elementu na formularzu; po wyłączeniu tego przełącznika kolumna będzie niewidoczna.

WSKAZÓWKA

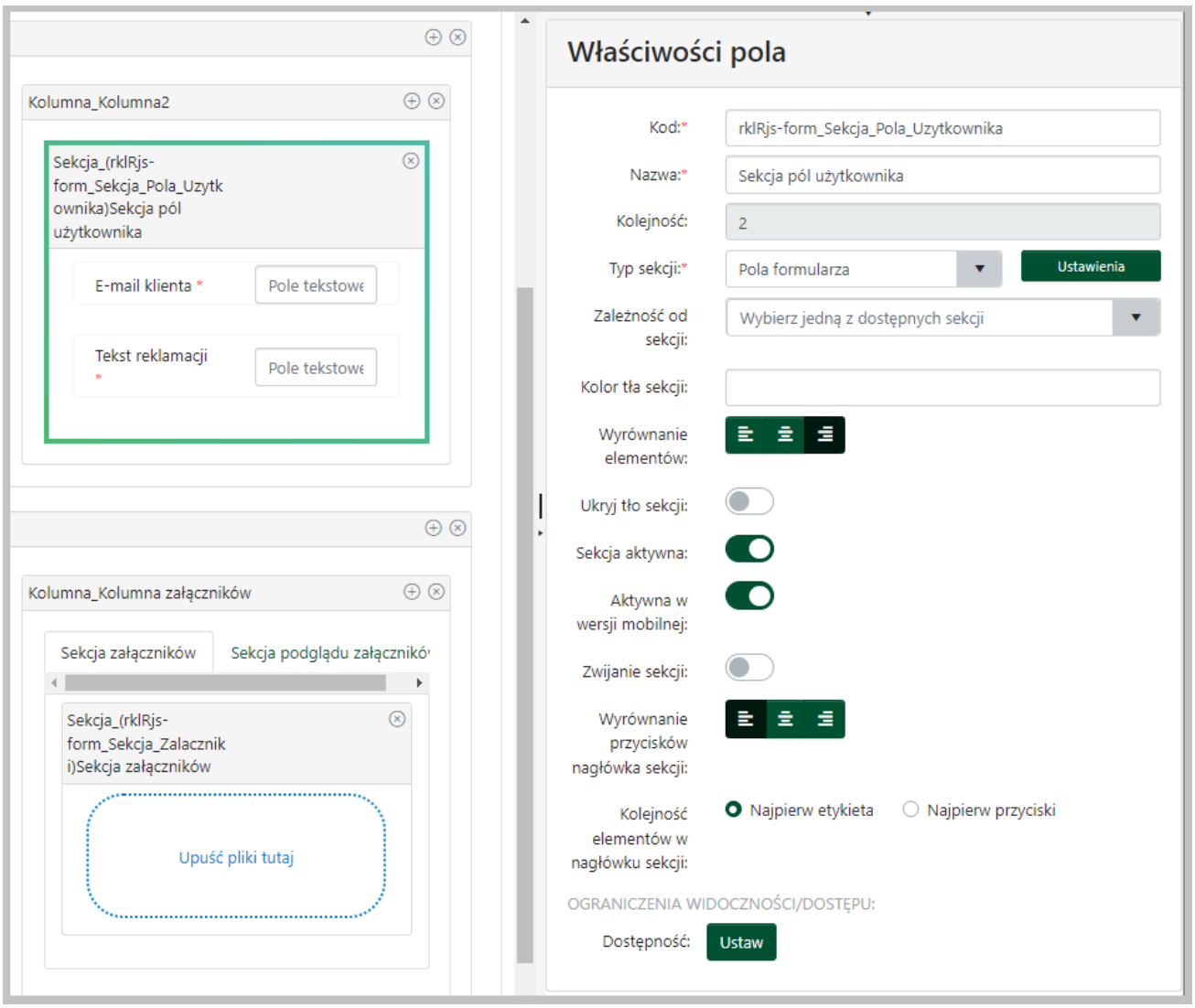
Kolumnę można ukryć za pomocą poniższego kodu Javascript w skrypcie na zmianę kontekstu formularza:

```
const element = document.querySelector('[data-code="<kod-kolumny>"]');
if (element) {
        element.style.display = 'none';
}
```

- `Aktywne w wersji mobilnej`: pozwala zdecydować, czy dany element formularza ma być widoczny w wersji mobilnej aplikacji.

4.1.5. Sekcja formularza

Podsumowanie:
Sekcja to element hierarchicznej struktury formularza, na którą składają się sam formularz, wiersze, kolumny, sekcje i kontrolki. Każdy element nadrzędny może zawierać kilka elementów podrzędnych. W nAxiom wyróżnia się kilka typów sekcji, które opisano w kolejnych rozdziałach. Jeśli w kolumnie formularza zdefiniuje się kilka sekcji, będą one wyświetlane jako osobne karty z widocznymi zakładkami do przełączania.



Właściwości sekcji formularza w trybie designera

Aby utworzyć nową sekcję na formularzu, kliknij kółko z symbolem `+` z prawej strony paska tytułu **kolumny** w trybie designera. Kolejne sekcje w danej kolumnie są dodawane jako karty, które wybiera się klikając ich zakładki. Aby usunąć sekcję z formularza, kliknij kółko z symbolem `x` z prawej strony paska tytułu sekcji.

> **UWAGA**
> Wraz z sekcją usuwane są wszystkie elementy, które się w niej znajdują; w razie przypadkowego usunięcia sekcji zamknij formularz bez zapisywania zmian, a następnie otwórz ponownie.

4.1.5.1. WSPÓLNE WŁAŚCIWOŚCI SEKCJI

Aby wyświetlić właściwości sekcji, w trybie designera kliknij pasek tytułu sekcji z lewej strony. Sekcje formularza współdzielą następujące właściwości:

`Id`: identyfikator sekcji; tylko do odczytu.

`Kod`: dowolny ciąg znaków identyfikujący sekcję na formularzu.

`Nazwa`: nazwa sekcji; tekst wyświetlany w widoku formularza na zakładce karty formularza.

`Kolejność`: pole tylko do odczytu z numerem kolejnym elementu; numery są przypisywane w miarę dodawania elementów.

`Typ sekcji`: pozwala określić rodzaj sekcji formularza z uwzględnieniem rodzaju danych wyświetlanych w tej sekcji oraz sposobu ich prezentacji. Wyboru dokonuje się z listy rozwijanej. Dla niektórych rodzajów obok listy jest wyświetlany przycisk `Ustawienia`, który wyświetla okno dialogowe z ustawieniami dla danego rodzaju sekcji. Ustawienie domyślne to `Pola formularza` — sekcja tego rodzaju zawiera kontrolki powiązane z kolumnami źródła danych formularza; dostępne rodzaje sekcji opisano w kolejnych tematach.

`Zależność od sekcji`: właściwość sekcji typu pola formularza, lista i harmonogram; pozwala wskazać sekcję, do której pól można będzie się odwoływać, używając zmiennej *{@DS_NazwaPola}*, gdzie *NazwaPola* to nazwa pola na sekcji wybranej z tej listy.

`Kolor tła`: w tym polu można wpisać nazwę koloru HTML (patrz www.w3schools.com) dla tła danego elementu.

`Ukryj tło sekcji`: jeśli w danej kolumnie jest tylko jedna sekcja, włączenie tego przełącznika spowoduje, że nie będzie ona wyświetlana jako karta, tylko pola tej sekcji zostaną wyświetlone bezpośrednio na tle kolumny.

`Sekcja aktywna`: służy do dezaktywowania elementu na formularzu; po wyłączeniu tego przełącznika sekcja będzie niewidoczna.

> **WSKAZÓWKA**
> Sekcję można ukryć, korzystając z funkcji wbudowanej Javascript w skrypcie na zmianę kontekstu formularza:
>
> ```
> setSectionAsHidden("<kod-sekcji>");
> ```
>
> Należy przy tym pamiętać o ukryciu kolumny tej sekcji, aby na formularzu nie pozostało puste miejsce.

`Aktywne w wersji mobilnej`: pozwala zdecydować, czy dany element formularza ma być widoczny w wersji mobilnej aplikacji.

`Zwijanie sekcji`: po włączeniu powoduje wyświetlenie na pasku tytułu sekcji z prawej strony trójkąta, którego kliknięcie zwija lub rozwija sekcję.

`Domyślnie zwinięta`: (przełącznik dostępny po włączeniu przełącznika `Zwijanie sekcji`) włączenie powoduje, że po wyświetleniu formularza, sekcja pozostaje zwinięta.

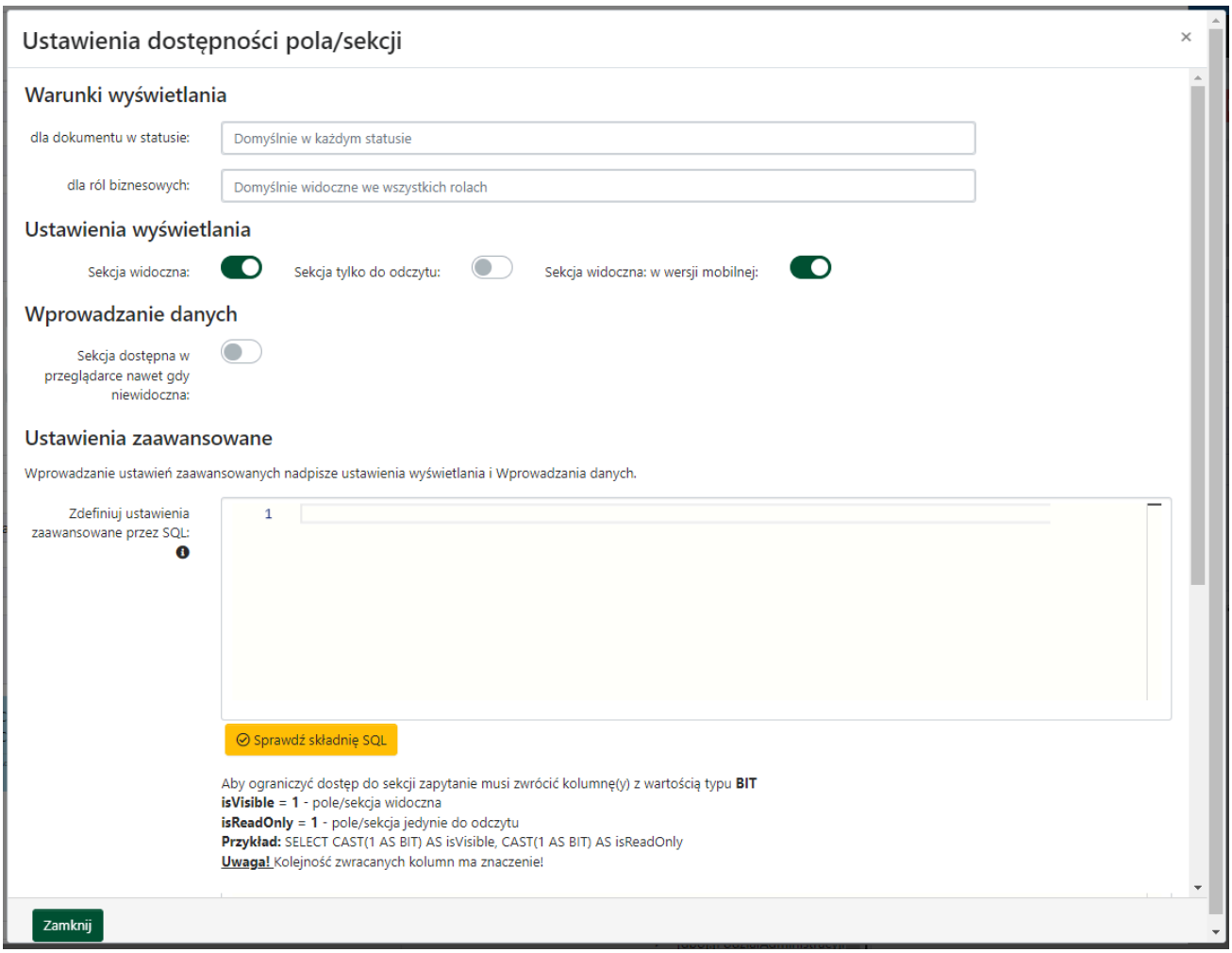
`Wyrównianie przycisków nagłówka sekcji`: kliknij odpowiedni przycisk, aby określić sposób wyrównania przycisków w nagłówku sekcji; dostępne wartości:

- `Wyrównaj do lewej`
- `Wyśrodkuj`
- `Wyrównaj do prawej`

`Kolejność elementów w nagłówku sekcji`: określa sposób rozmieszczenia etykiety z nazwą sekcji względem przycisków dodanych w nagłówku sekcji (od prawej strony); dostępne wartości:

- `Najpierw etykieta`
- `Najpierw przyciski`

`Dostępność`: kliknięcie przycisku `Ustaw` powoduje wyświetlenie okna dialogowego `Ustawienia dostępności pola/sekcji`, w którym można ustawić warunki widoczności, dostępności (możliwości edycji) i wymagalności danego elementu formularza przy użyciu następujących pól i elementów sterujących:



Ustawienia dostępności pola/sekcji

- `dla dokumentu w statusie`: lista wielowyboru; kliknij i wybierz statusy, w których sekcja/kontrolka ma być widoczna; w pozostałych statusach sekcja/kontrolka będzie niewidoczna; brak wybranych statusów (ustawienie domyślne) powoduje, że sekcja/kontrolka jest wyświetlana we wszystkich statusach.
- `dla ról biznesowych`: lista wielowyboru; kliknij i wybierz role, dla których sekcja/kontrolka ma być widoczna; dla pozostałych ról sekcja/kontrolka będzie niewidoczna; brak wybranych ról (ustawienie domyślne) powoduje, że sekcja/kontrolka jest wyświetlana dla wszystkich ról.
- `Sekcja/Pole widoczne`: wyłączenie przełącznika powoduje ukrycie sekcji/kontrolki na formularzu; w przypadku sekcji ten przełącznik jest sprzężony z przełącznikiem `Sekcja aktywna`.
- `Pole wymagane`: (przełącznik niedostępny dla sekcji formularza) pozwala w niektórych przypadkach przesłonić ustawienie `Zezwól Null` dla kolumny tabeli powiązanej z polem formularza. Domyślny stan przełącznika odpowiada ustawieniu dla kolumny (jeśli atrybut `Zezwól Null` ma wartość *false*, to przełącznik `Pole wymagane` jest załączony i odwrotnie). W ogólności służy do zmiany w „obowiązkowe" pola, dla którego powiązana kolumna tabeli zezwala na wartości NULL. W sytuacji odwrotnej, to jest kiedy kolumna nie zezwala na wartości NULL,

wyłączenie tego przełącznika będzie powodowało zapisywanie wartości `''` (pusty string) w polach typu `varchar` i `nvarchar` oraz `0` w polach typu liczbowego i będzie ignorowane w polach innych typów.

- `Sekcja/Pole tylko do odczytu`: domyślnie wyłączone, załączenie powoduje zablokowanie możliwości edycji wartości danej kontrolki, w przypadku sekcji blokuje możliwość edycji wartości we wszystkich polach sekcji.
- `Sekcja/Pole widoczne w wersji mobilnej`: domyślnie włączone, wyłączenie powoduje, że dana sekcja/pole nie będzie widoczne w wersji mobilnej aplikacji.
- `Sekcja/Pole dostępne w przeglądarce nawet gdy niewidoczne`: włączenie umożliwia wykonywanie operacji na wartościach pól w sekcji niewidocznej na formularzu.
- `Ustawienia zaawansowane`: okno edytora SQL, w którym można wpisać warunek SQL określający wartości przełączników `Sekcja/Pole widoczne`, `Pole wymagane` i `Sekcja/Pole tylko do odczytu`. Warunek ma postać zapytania, które musi zwracać trzy kolumny o podanych nazwach i w podanej kolejności:

```
SELECT
  CAST(warunek1 AS BIT) AS isVisible,
  CAST(warunek2 AS BIT) AS isRequired,
  CAST(warunek3 AS BIT) AS isReadOnly
```

gdzie *warunek**n*** to wyrażenie, które zwróci wartość `0` lub `1`. Na przykład poniższe zapytanie będzie blokować możliwość edycji danego pola, jeśli do daty w polu `TerminZakonczenia` pozostanie mniej niż 5 dni:

```
SELECT
  CAST(1 AS BIT) AS isVisible,
  CAST(1 AS BIT) AS isRequired,
  CAST(CASE
         WHEN DATEDIFF(day, GETDATE(), {@TerminZakonczenia}) < 5
         THEN 1
         ELSE 0
         END
  AS BIT) AS isReadOnly
```

UWAGI

- Warunki dostępności są obliczane już po załadowaniu wszystkich danych formularza i można w nim używać odwołań do pól formularza, pól technicznych i pól dodawanych w źródle danych dla formularza; obsługiwane są także wartości pól przekazywane w akcji otwierania formularza.
- Wyłączenie widoczności kontrolki dla nowego rekordu wyłącza zapis danych z tej kontrolki w tabeli, nawet jeśli dla kontrolki zdefiniowano wartość domyślną. Może to powodować błędy w przypadku, gdy kolumna powiązana z kontrolką nie zezwala na wartości `NULL`. Aby móc zapisywać w bazie wartość domyślną z ukrytej kontrolki, należy do jej ukrycia użyć funkcji `Modyfikacja zależności kontrolki`.
- Wyłączenie widoczności przełącznikiem `Sekcja/Pole widoczne` ma pierwszeństwo przed ustawieniami pól `dla dokumentu w statusie` i `dla ról biznesowych`.
- Wprowadzenie wyrażenia SQL w sekcji `Ustawienia zaawansowane` nadpisuje ustawienia wyświetlania i wprowadzania danych.

4.1.5.2. USTAWIENIA DOSTĘPNOŚCI POLA/SEKCJI

Ustawienia sterowania widocznością sekcji i kontrolek na formularzu są dostępne na karcie `Dostępność pola` (w kreatorze formularza — widok drzewa) i w oknie dialogowym `Ustawienia dostępności pola/sekcji` (w designerze formularza) wyświetlanym przyciskiem `Ustaw` w sekcji `Ograniczenia widoczności/dostępu` na panelu `Właściwości pola`.

Do sterowania widocznością, wymagalnością i dostępnością (możliwością zmiany wartości) elementów formularza służą następujące ustawienia:

- `dla dokumentu w statusie`: lista wielowyboru; kliknij i wybierz statusy, w których sekcja/kontrolka ma być widoczna; w pozostałych statusach sekcja/kontrolka będzie niewidoczna; brak wybranych statusów (ustawienie domyślne) powoduje, że sekcja/kontrolka jest wyświetlana we wszystkich statusach.
- `dla ról biznesowych`: lista wielowyboru; kliknij i wybierz role, dla których sekcja/kontrolka ma być widoczna; dla pozostałych ról sekcja/kontrolka będzie niewidoczna; brak wybranych ról (ustawienie domyślne) powoduje, że sekcja/kontrolka jest wyświetlana dla wszystkich ról.
- `Sekcja/Pole widoczne`: wyłączenie przełącznika powoduje ukrycie sekcji/kontrolki na formularzu; w przypadku sekcji ten przełącznik jest sprzężony z przełącznikiem `Sekcja aktywna`.
- `Pole wymagane`: (przełącznik niedostępny dla sekcji formularza) pozwala w

niektórych przypadkach przesłonić ustawienie `Zezwól Null` dla kolumny tabeli powiązanej z polem formularza. Domyślny stan przełącznika odpowiada ustawieniu dla kolumny (jeśli atrybut `Zezwól Null` ma wartość *false*, to przełącznik `Pole wymagane` jest załączony i odwrotnie). W ogólności służy do zmiany w „obowiązkowe" pola, dla którego powiązana kolumna tabeli zezwala na wartości NULL. W sytuacji odwrotnej, to jest kiedy kolumna nie zezwala na wartości NULL, wyłączenie tego przełącznika będzie powodowało zapisywanie wartości `''` (pusty string) w polach typu `varchar` i `nvarchar` oraz `0` w polach typu liczbowego i będzie ignorowane w polach innych typów.

- `Sekcja/Pole tylko do odczytu`: domyślnie wyłączone, załączenie powoduje zablokowanie możliwości edycji wartości danej kontrolki, w przypadku sekcji blokuje możliwość edycji wartości we wszystkich polach sekcji.
- `Sekcja/Pole widoczne w wersji mobilnej`: domyślnie włączone, wyłączenie powoduje, że dana sekcja/pole nie będzie widoczne w wersji mobilnej aplikacji.
- `Sekcja/Pole dostępne w przeglądarce nawet gdy niewidoczne`: włączenie umożliwia wykonywanie operacji na wartościach pól w sekcji niewidocznej na formularzu.
- `Ustawienia zaawansowane`: okno edytora SQL, w którym można wpisać warunek SQL określający wartości przełączników `Sekcja/Pole widoczne`, `Pole wymagane` i `Sekcja/Pole tylko do odczytu`. Warunek ma postać zapytania, które musi zwracać trzy kolumny o podanych nazwach i w podanej kolejności:

```
SELECT
  CAST(warunek1 AS BIT) AS isVisible,
  CAST(warunek2 AS BIT) AS isRequired,
  CAST(warunek3 AS BIT) AS isReadOnly
```

gdzie *warunek**n*** to wyrażenie, które zwróci wartość `0` lub `1`. Na przykład poniższe zapytanie będzie blokować możliwość edycji danego pola, jeśli do daty w polu `TerminZakonczenia` pozostanie mniej niż 5 dni:

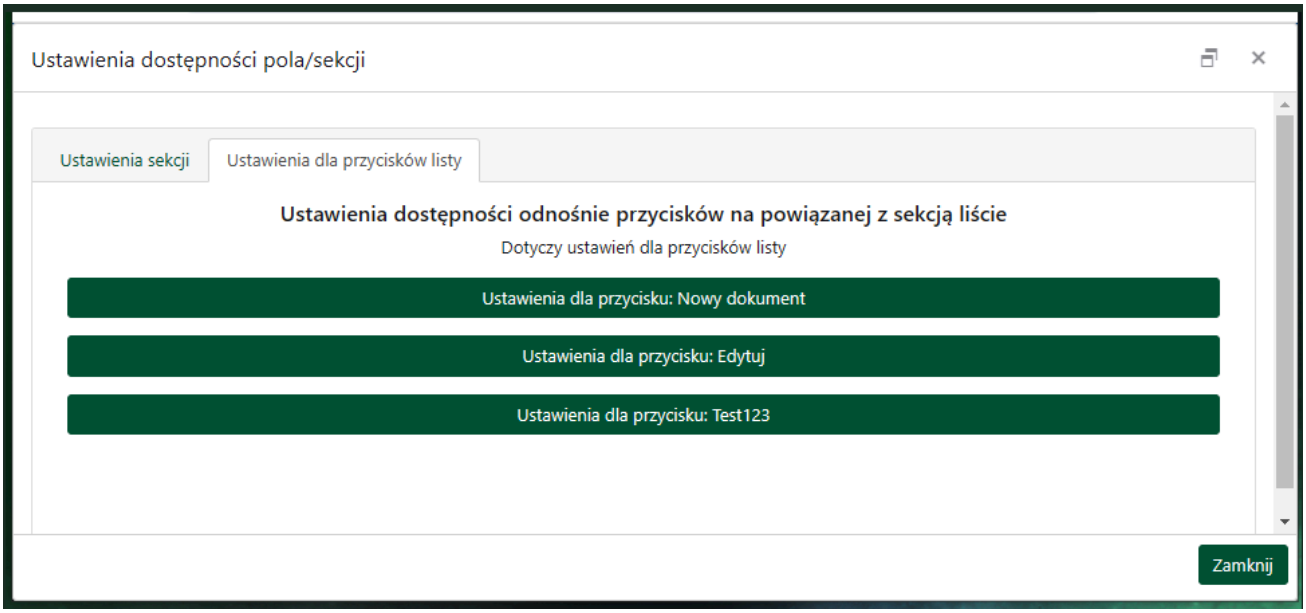
```
SELECT
  CAST(1 AS BIT) AS isVisible,
  CAST(1 AS BIT) AS isRequired,
  CAST(CASE
         WHEN DATEDIFF(day, GETDATE(), {@TerminZakonczenia}) < 5
         THEN 1
         ELSE 0
         END
  AS BIT) AS isReadOnly
```

UWAGI

- Warunki dostępności są obliczane już po załadowaniu wszystkich danych formularza i można w nim używać odwołań do pól formularza, pól technicznych i pól dodawanych w źródle danych dla formularza; obsługiwane są także wartości pól przekazywane w akcji otwierania formularza.
- Wyłączenie widoczności kontrolki dla nowego rekordu wyłącza zapis danych z tej kontrolki w tabeli, nawet jeśli dla kontrolki zdefiniowano wartość domyślną. Może to powodować błędy w przypadku, gdy kolumna powiązana z kontrolką nie zezwala na wartości `NULL`. Aby móc zapisywać w bazie wartość domyślną z ukrytej kontrolki, należy do jej ukrycia użyć funkcji `Modyfikacja zależności kontrolki`.
- Wyłączenie widoczności przełącznikiem `Sekcja/Pole widoczne` ma pierwszeństwo przed ustawieniami pól `dla dokumentu w statusie` i `dla ról biznesowych`.
- Wprowadzenie wyrażenia SQL w sekcji `Ustawienia zaawansowane` nadpisuje ustawienia wyświetlania i wprowadzania danych.

USTAWIENIA DOSTĘPNOŚCI SEKCJI TYPU LISTA

W przypadku sekcji typu lista po kliknięciu przycisku ustawień dostępności jest wyświetlane okno dialogowe z dwoma kartami: `Ustawienia sekcji` i `Ustawienia przycisków listy`. Na tej drugiej karcie można skonfigurować ustawienia dostępności przycisków listy używanych w nagłówku listy i w kolumnie akcji (w tym dla przycisków do obsługi edycji inline). Same ustawienia dostępności dla przycisków są identyczne jak dla sekcji i innych kontrolek formularza.



Ustawienia dostępności przycisków listy (dla sekcji typu lista)

4.1.5.3. POLA FORMULARZA

Podsumowanie:
Domyślny typ sekcji. Służy do umieszczania na sekcji kontrolek powiązanych z polami źródła danych formularza.

Oprócz właściwości współdzielonych z innymi typami sekcji, w tej sekcji jest dostępny przełącznik:

- `Edytor JavaScript`: wyświetla pole, w którym można wpisać kod JavaScript z odwołaniem do wartości z kontekstu formularza (zmienna *FormDataContext*), sekcji wybranej z listy `Zależność od sekcji` (zmienna *DSDataContext*) lub aktualizujący stan kontrolek na formularzu (kod aktualizowanej przez system kontrolki: *FormDataKey*).

Powiązane tematy:

- Wspólne właściwości sekcji

4.1.5.4. KOMENTARZE

Podsumowanie:
Sekcja komentarzy jest domyślnie dodawana do formularzy generowanych automatycznie poleceniem `Generuj elementy aplikacji` i umożliwia dodawanie komentarzy do dokumentów wyświetlanych przez użytkowników w formularzu.

Ten rodzaj sekcji nie ma żadnych specjalnych właściwości.

Powiązane tematy:

- Wspólne właściwości sekcji

4.1.5.5. LISTA

Podsumowanie:
W tej sekcji można osadzić na formularzu widok listy.

Oprócz właściwości współdzielonych ta sekcja wymaga określenia następujących właściwości:

- `Wyświetlana lista`: wybierz widok listy, który będzie wyświetlany w sekcji.
- `Ustawienia`: wyświetla okno dialogowe z ustawieniami listy osadzonej w

formularzu dotyczącymi jej wyświetlania w aplikacji *FrontSPA*:

  - `Wyświetlaj zdefiniowane filtry listy`: pozwala korzystać z rozwijanej listy zdefiniowanych filtrów.
  - `Wyświetlaj konfiguracje listy w nagłówku`: wyświetla listę zapisanych konfiguracji.
  - `Wyświetlaj przycisk eksportu listy do Excela`: wyświetla ikonę eksportu danych listy do Excela.
  - `Wyświetlaj przycisk odświeżania listy`: wyświetla ikonę odświeżania danych w sekcji typu lista.
  - `Zapamiętuj tymczasową konfigurację listy`: pozwala zapamiętywać konfigurację listy ustawioną przez użytkownika w aplikacji *FrontSPA* (grupowanie, sortowanie, szerokość i kolejność kolumn) na czas przełączenia do innej karty.
  - `Wymuś wyświetlanie listy w trybie mobilnym`: stosuje ustawienia trybu mobilnego do wyświetlania listy.
  - `Wyświetlaj nazwę kolumn listy w tej sekcji`: pozwala wyłączyć wyświetlanie nazw kolumn w sekcji.
  - `Wymuś wyświetlanie przycisków kolumny akcji jeden nad drugim`: wymusza „pionowe” rozmieszczenie przycisków w kolumnie akcji.
  - `Pokazuj w paginatorach liczbę kart i nawigację między kartami (jeżeli włączono paginację)`: pozwala ograniczyć informacje wyświetlane na pasku paginacji pod listą. - `Edytuj`: wyświetla okno `Kreator listy`, w którym można dokonać edycji listy osadzonej w sekcji formularza.

- `Wyświetlana lista`: wybierz widok listy, który będzie wyświetlany w sekcji.
- `Ustawienia`: wyświetla okno dialogowe z ustawieniami listy osadzonej w formularzu dotyczącymi jej wyświetlania w aplikacji *FrontSPA*:
  - `Wyświetlaj zdefiniowane filtry listy`: pozwala korzystać z rozwijanej listy zdefiniowanych filtrów.
  - `Wyświetlaj konfiguracje listy w nagłówku`: wyświetla listę zapisanych konfiguracji.
  - `Wyświetlaj przycisk eksportu listy do Excela`: wyświetla ikonę eksportu danych listy do Excela.
  - `Wyświetlaj przycisk widoczności kolumn`: wyświetla przycisk widoczności kolumn w sekcji typu lista.
  - `Wyświetlaj przycisk odświeżania listy`: wyświetla ikonę odświeżania danych w sekcji typu lista.

    - `Zapamiętuj tymczasową konfigurację listy`: pozwala zapamiętywać konfigurację listy ustawioną przez użytkownika w aplikacji *FrontSPA* (grupowanie, sortowanie, szerokość i kolejność kolumn) na czas przełączenia do innej karty.
    - `Wymuś wyświetlanie listy w trybie mobilnym`: stosuje ustawienia trybu mobilnego do wyświetlania listy.
    - `Wyświetlaj nazwę kolumn listy w tej sekcji`: pozwala wyłączyć wyświetlanie nazw kolumn w sekcji.
    - `Wymuś wyświetlanie przycisków kolumny akcji jeden nad drugim`: wymusza „pionowe" rozmieszczenie przycisków w kolumnie akcji.
    - `Pokazuj w paginatorach liczbę kart i nawigację między kartami (jeżeli włączono paginację)`: pozwala ograniczyć informacje wyświetlane na pasku paginacji pod listą. - `Edytuj`: wyświetla okno `Kreator listy`, w którym można dokonać edycji listy osadzonej w sekcji formularza.

- `Zależność od sekcji`: pozwala określić sekcję zawierającą pole, którego wartości będą mogły zostać użyte do filtrowania listy; możliwe jest także filtrowanie bieżącej listy zależnie od rekordów zaznaczonych we wskazanej sekcji typu lista; ponadto zależność powoduje, że sekcja zależna będzie odświeżana każdorazowo po zmianie danych w sekcji nadrzędnej.

- `Warunek dla zależności`: wyrażenie SQL filtrujące rekordy sekcji typu lista, zależnie od wartości pola na innej sekcji; możliwe są dwa rodzaje warunków:
    - *KolumnaListy operator {@_Section_PoleFormularza}*: *KolumnaListy* to nazwa kolumny listy wyświetlanej w sekcji, *@_Section_* to przedrostek, a *PoleFormularza* to właściwość *Pole formularza* pola w sekcji wskazanej jako nadrzędna. Wartość w tym polu jest używana do filtrowania rekordów listy.
    - *KolumnaListy IN (SELECT [value] FROM OPENJSON({@_Section_KodSekcji_Selected}))*: w tym przypadku zmienna *{@_Section_KodSekcji_Selected}* zwraca tablicę JSON (*[Id1, Id2...]*) z identyfikatorami rekordów zaznaczonych w sekcji nadrzędnej.

W edytorze SQL obsługiwany jest mechanizm *smart numbers*. Po kliknięciu przycisku `Podgląd zapytania SQL` wyświetlane jest okno dialogowe edytora, w którym można sprawdzić poprawność składni. Aby to było możliwe, w obszarze u góry okna wyświetlana jest w trybie tylko do odczytu część zapytania dla zależności w formie:

```
SELECT * FROM (SELECT * FROM List_View) WHERE...
```

Poniżej jest wyświetlany sam warunek.

Aby sprawdzić poprawność składni, należy w polach z prawej strony podać wartości parametrów (*{@Nazwa}*) występujących w zapytaniu. Dodatkowo, po kliknięciu przycisku `Wykonaj zapytanie` zostanie wyświetlone okno edytora SQL z całym zapytaniem warunku zależności. Po podstawieniu za parametry przykładowych wartości i ponownym kliknięciu przycisku `Wykonaj zapytanie` w dolnej części okna zostaną wyświetlone wyniki zwrócone przez zapytanie.

> **OSTRZEŻENIE**
> Składnia *{@_Section_KodSekcji_Selected}* może być używana wyłącznie w warunku zależności od sekcji. W innych przypadkach (np. w akcji SQL), aby odwołać się do listy zaznaczonych rekordów na liście osadzonej w formularzu, należy użyć zmiennej *{@KodSekcji_Selected}*.

- `Dostępność`: kliknięcie przycisku `Ustaw` wyświetla okno dialogowe z dwiema kartami:
  - `Ustawienia sekcji`: ustawienia dostępności całej sekcji.
  - `Ustawienia dla przycisków listy`:
    - `Ustawienia dla przycisku Nowy dokument`: ustawienia dostępności przycisku `Nowy dokument` wyświetlanego standardowo w nagłówku listy.
    - `Ustawienia dla przycisku Edytuj`: ustawienia dostępności przycisku `Edytuj` wyświetlanego standardowo w każdym rekordzie listy (w kolumnie akcji).

  Ustawienia dostępności dla całej sekcji typu lista oraz dla wymienionych powyżej przycisków określa się tak samo, jak dla innych sekcji i kontrolek formularza.

- `Obliczanie dostępności przycisków nagłówka i kolumny akcji listy`: dostępne wartości to:
  - `z ustawień sekcji`: dla przycisków listy osadzonej na formularzu obowiązują ustawienia dostępności określone we właściwościach sekcji typu lista.
  - `z definicji listy`: dla przycisków listy osadzonej na formularzu obowiązują ustawienia dostępności określone w definicji listy.

PRZYKŁAD

Poniższy warunek zależności spowoduje wyświetlenie na liście zależnej tylko tych rekordów, które zostały zaznaczone na liście nadrzędnej. *KodSekcji* to kod sekcji nadrzędnej (wybranej z listy `Zależność od sekcji`).

UWAGA
W odwołaniu *@_Section_**KodSekcji**_Selected* kod sekcji musi się zaczynać od wielkiej litery, niezależnie od tego, od jakiej litery zaczyna się kod sekcji nadrzędnej.

```
[Id] IN
(SELECT [value]
FROM
OPENJSON({@_Section_KodSekcji_Selected}))
```

Powiązane tematy:

- Wspólne właściwości sekcji
- Listy

4.1.5.6. PROSTA LISTA

Podsumowanie:
W tej sekcji można wyświetlić tabelę z danymi dokumentów zwróconymi przez zapytanie SQL. Jedną z kolumn tej tabeli jest kolumna z linkiem do dokumentu otwierającego ten dokument w odpowiednim formularzu.

Oprócz właściwości współdzielonych ta sekcja wymaga określenia właściwości w oknie dialogowym otwieranym przyciskiem `Ustawienia`:

`Polecenie SQL`: zapytanie SQL zwracające kolumny, które zostaną wyświetlone w sekcji; jeśli w sekcji ma być wyświetlany link zapytanie musi zwrócić kolumnę z identyfikatorem dokumentu, w takim przypadku zapytanie musi zwracać dodatkową kolumnę z wartością, która będzie wyświetlana jako tekst linku. Na przykład:

```
SELECT Id,
       Code AS Link,
       TekstReklamacji AS Opis
FROM [dbo].[RejestrReklamacji]
```

Sekcja pól użytkownika | Reklamacje

Link	Opis
0028/2021	nic nie działa
0029/2021	28-02-2021 przez całe popołudnie nie było usługi, w ogóle
0030/2021	nic nie działa
0031/2021	01.03.2021, 07:30-12:30, brak usług w ogóle
0032/2021	01.03.2021, 07:30-12:30, brak usługi
0033/2021	nic nie działa
0034/2021	nic nie działa
0001/2021	brak usługi w dn. 21-07-21 po godz. 18
0002/2021	W dn. 30-08-21 usługa nie spełnia parametrów
0003/2021	10-09-21 przerwy w działaniu usługi między 12:00 i 16:00

Sekcja typu prosta lista

`Nazwa kolumny z ID`: nazwa kolumny z identyfikatorem (potrzebna, jeśli ma zostać użyta w linku).

`Ukryć kolumnę z ID`: pozwala ukryć kolumnę z identyfikatorem.

`Nazwa kolumny z linkiem`: Nazwa kolumny, w której mają być wyświetlane linki do odpowiednich dokumentów.

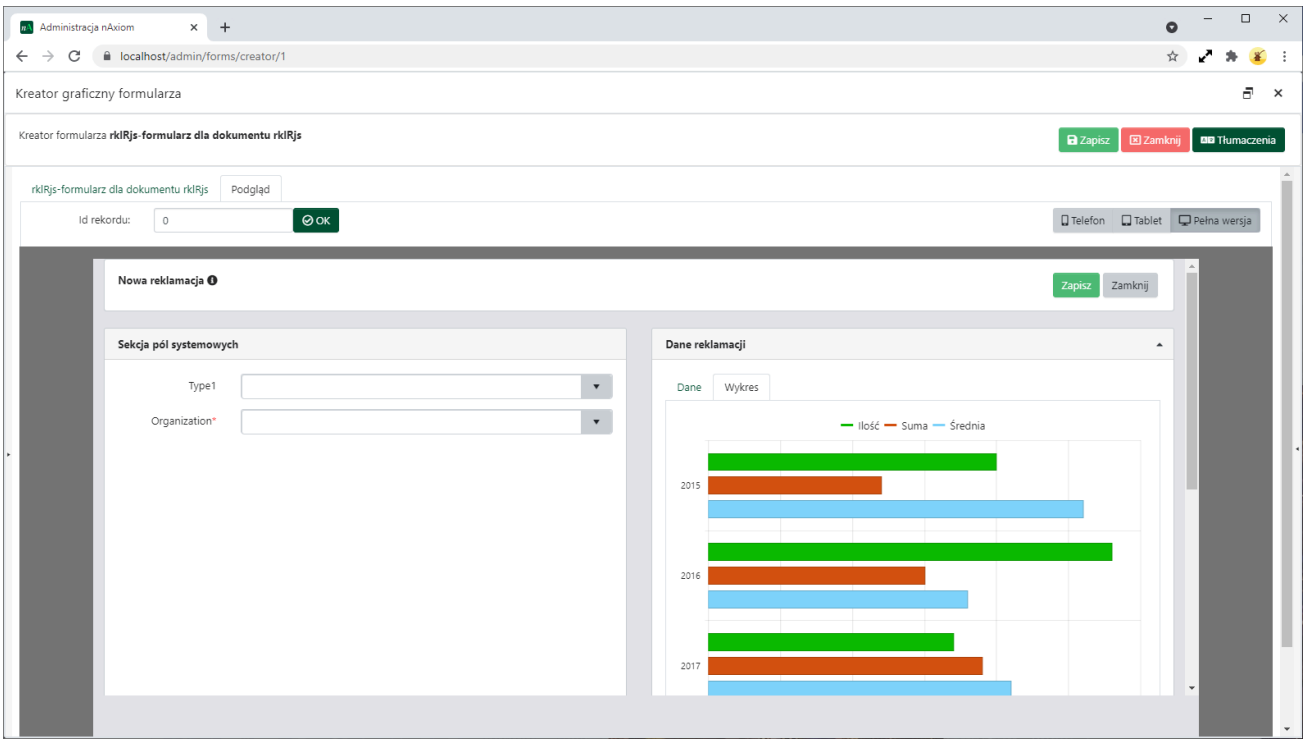
`URL`: przycisk `Konfiguruj` wyświetla okno dialogowe, w którym można skonfigurować sposób działania linku (podobne jak okno dialogowe akcji `Otwieranie formularza`).

Powiązane tematy:

- Wspólne właściwości sekcji
- Otwieranie formularza

4.1.5.7. WYKRES

Podsumowanie:
Ta sekcja pozwala wyświetlić na formularzu wykres.



Sekcja typu Wykres na podglądzie formularza

Oprócz właściwości współdzielonych ta sekcja wymaga ustawienia następujących właściwości:

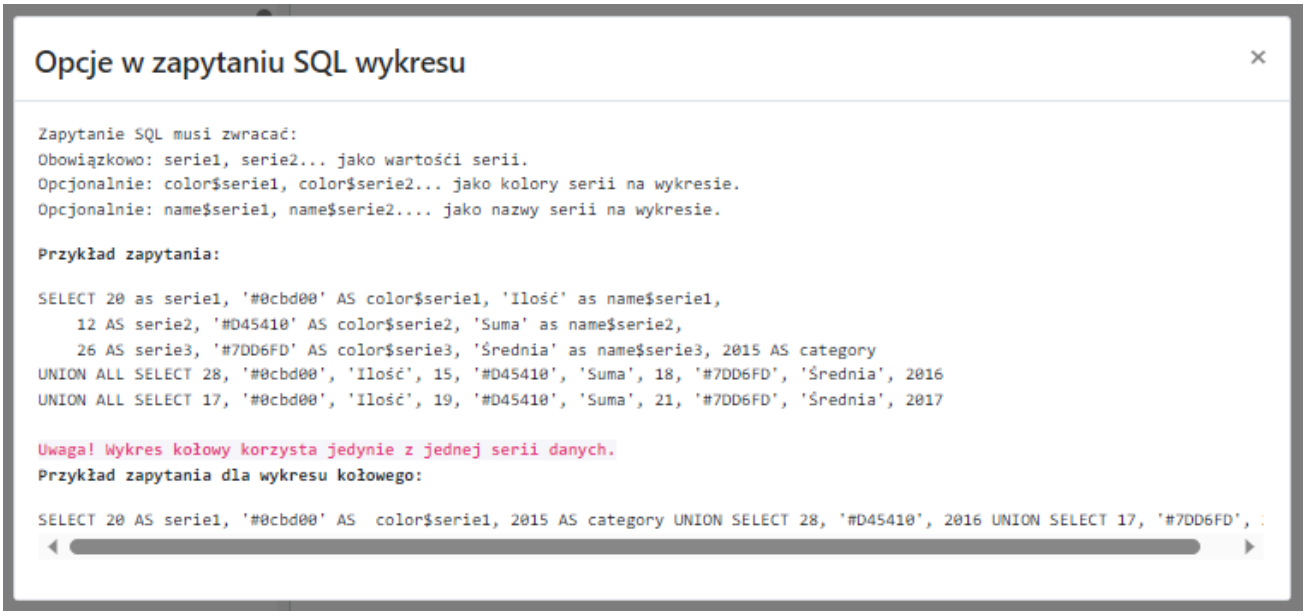
`SQL wykresu`: zapytanie SQL zwracające dane do wykresu; zapytanie zwraca kolumny:

- `serien`: wartość serii o numerze *n* (wymagane)
- `color$serien`: kolor serii o numerze *n* (opcjonalnie)
- `name$serien`: nazwa serii o numerze *n* (opcjonalnie)
- `category`: kategoria serii
- `Typ wykresu`: rodzaj wykresu, dostępne wartości:
  - `Wykres słupkowy`: (domyślny) dane serii są prezentowane jako poziome słupki.
  - `Wykres kołowy`: na wykresie kołowym można przedstawić tylko jedną serię danych.
  - `Wykres kolumnowy`: dane serii są prezentowane jako pionowe kolumny.
- `Serie skumulowane`: powoduje nałożenie na siebie słupków/kolumn na wykresie, ustawienie niedostępne dla wykresu kołowego.
- `Kolor bazowy wykresu`: podstawowy kolor wykresu; można wpisać wartość szesnastkową koloru w tym polu.
- `Losowy kolor`: wybiera losowy kolor bazowy wykresu.
- `Linia trendu`: rysuje na wykresie linię trendu; ustawienie dostępne tylko dla

wykresu kolumnowego.

- `Drill down`: umożliwia podanie zapytania SQL, które zwróci dane dla parametru {@category}. Dane te zostaną wyświetlone w oknie wyskakującym po kliknięciu słupka, kolumny lub wycinka na wykresie.

Przycisk `Przykład zwracanego SQL` wyświetla przykłady zapytań i dodatkowe informacje.



Pomoc do zapytania SQL dla wykresu

Powiązane tematy:

- Wspólne właściwości sekcji

4.1.5.8. SEKCJA ZAŁĄCZNIKI

Podsumowanie:
Sekcja typu `Załączniki` służy do dodawania do dokumentu załączników przez upuszczanie ich na tę sekcję. Jest ona dodawana domyślnie do formularzy generowanych automatycznie poleceniem `Generuj elementy aplikacji`. W tej sekcji można także dodać krótki opis załącznika. Dane dotyczące załączników są przechowywane w tabeli *core.Attachments*.

WAŻNE
W przypadku włączonej obsługi uprawnień ACL do dokumentów, operacje na załącznikach (przez UI i przez API) wymagają posiadania uprawnień do aktualizacji dokumentu (dodanie/usunięcie załącznika). Aby pobrać załącznik żądaniem API wymagane jest uprawnienie do odczytu dokumentu.

Załączniki dodawane do dokumentów można przypisywać do różnych kategorii, które definiuje się poleceniem `DANE` > `Kategorie załączników` > `Nowa kategoria załączników`. W definicji kategorii określa się między innymi dozwolone rozszerzenia plików załączników. Dodając sekcję załączników na formularzu, należy określić kategorię, do której będą przypisywane załączniki dodawane w tej sekcji. Służy do tego właściwość `Kategoria załączników`. Wybór kategorii dla sekcji jest obowiązkowy. Jeśli nie jest planowane tworzenie kategorii załączników, należy wybrać kategorię domyślną.

Kategoria domyślna jest standardowo tworzona podczas generowania definicji dokumentu i domyślnego formularza (poleceniem `Generuj elementy aplikacji` w kreatorze modeli danych). Dla tej kategorii obsługiwane typy plików definiuje się w ustawieniach nAxiom (`ADMINISTRACJA` > `Ustawienia systemu` > `Załączniki`).

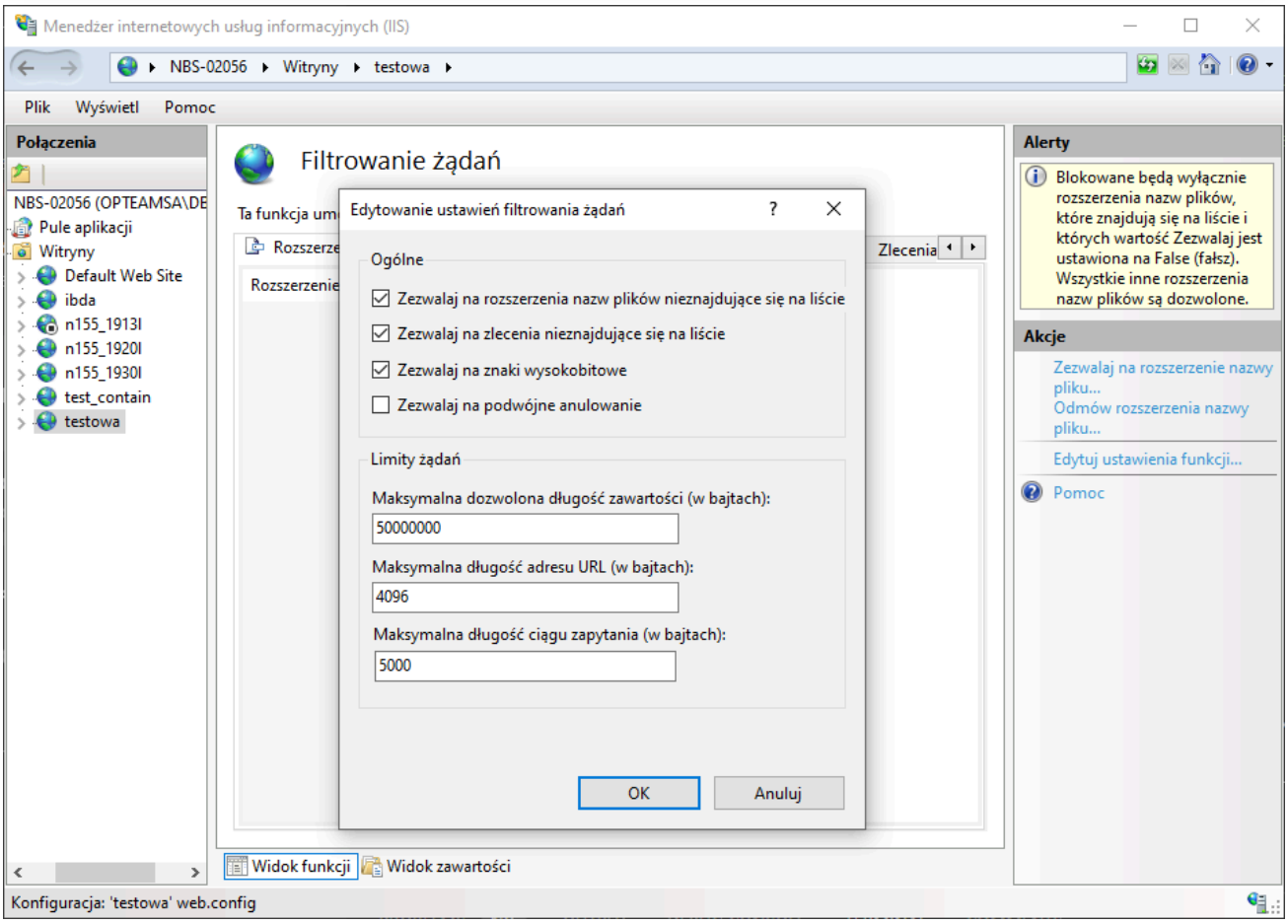
Na formularzu można dodać wiele sekcji załączników dla różnych kategorii. Aby wyświetlić podgląd załącznika, konieczne jest dodanie sekcji podglądu. Taka sekcja może obsługiwać wyświetlanie załączników z wszystkich sekcji lub tylko z wybranej sekcji załączników.

Sekcja typu `Załączniki` na podglądzie formularza

Po dodaniu załącznika można kliknąć etykietę `dodaj opis`, aby włączyć tryb edycji inline i wpisać opis załącznika. Opis nie może przekraczać 300 znaków i jest dodawany do indeksu wyszukiwania.

INFO

Maksymalny rozmiar załącznika w nAxiom zależy od parametru `Maksymalna dozwolona długość zawartości (w bajtach)` dostępnego w programie *Menedżer internetowych usług informacyjnych (IIS)*. Aby skonfigurować ten parametr, należy kliknąć dwukrotnie funkcję *Filtrowanie żądań*, a następnie kliknąć polecenie `Edytuj ustawienia funkcji` w menu z prawej strony. W wyświetlonym oknie dialogowym należy wpisać żądaną wartość parametru. Parametr można zdefiniować dla wszystkich witryn na poziomie serwera (hosta) lub indywidualnie dla każdej witryny.



Ustawianie maks. wielkości załącznika w Menedżerze IIS

Powiązane tematy:

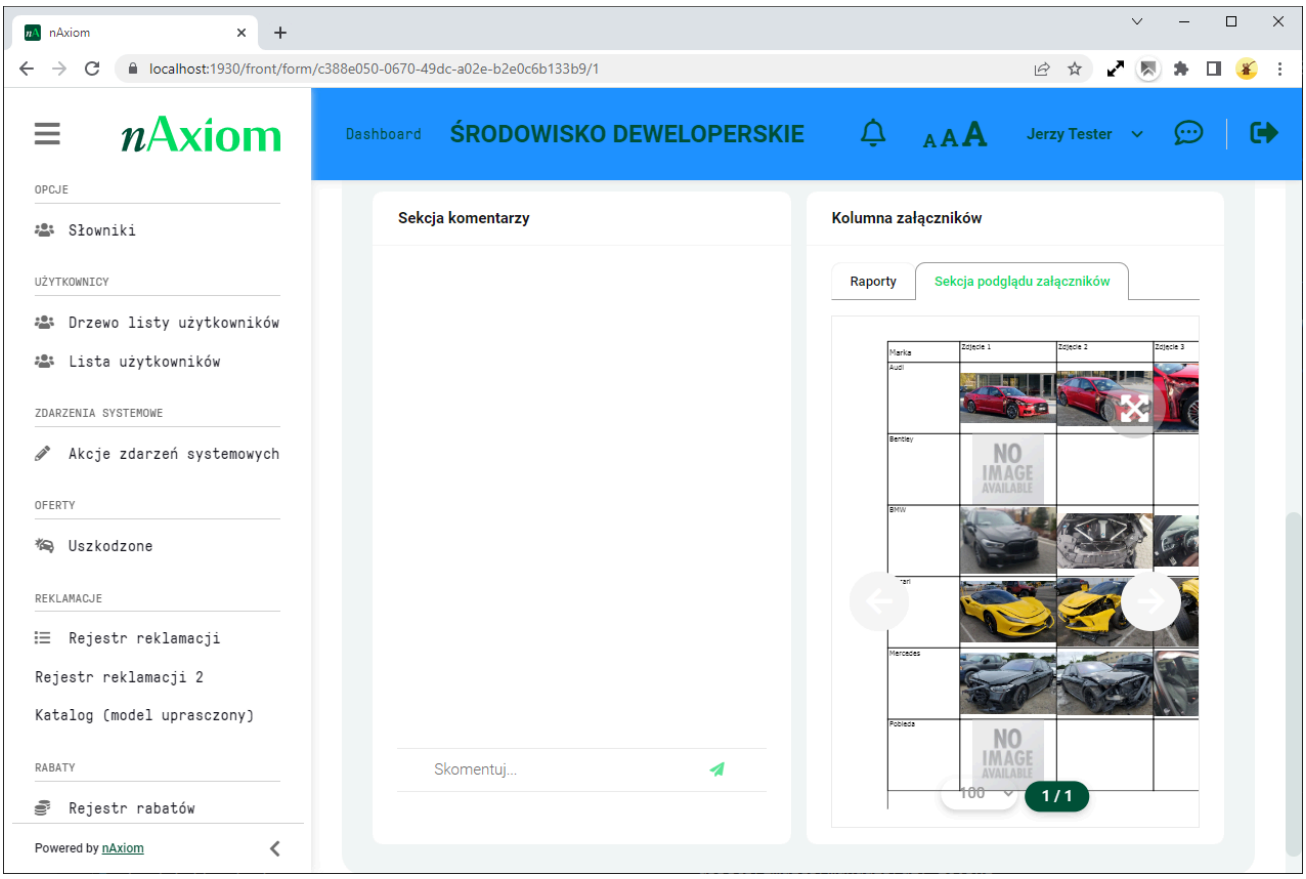
- Wspólne właściwości sekcji

Podsumowanie:
Sekcja typu `Załączniki – podgląd` służy do wyświetlania podglądu załączników dodanych do dokumentu. Obsługiwane formaty to: *PNG*, *JPG*, *PDF*, *TXT*, *JPE*, *JPEG*, *APNG*, *GIF*, *SVG*, *WEBP*, *BMP*, *JSON*, *XML*, *DOC* i *XLSX*. Aby wyświetlić podgląd załącznika, należy najpierw kliknąć go na liście w sekcji załączników.

WSKAZÓWKA
W celu wyświetlenia podglądu pliki programu Word (*doc*) są konwertowane do formatu PDF, dlatego ich obraz może różnić się od oryginalnego pliku.

W przypadku pliku załącznika w formacie, dla którego wyświetlenie podglądu nie jest obsługiwane, wyświetlany jest komunikat z informację, że dla danego typu pliku podgląd jest niedostępny. Nie jest także wyświetlany plików PDF z ustawionym hasłem.

Ta sekcja jest dodawana domyślnie do formularzy generowanych automatycznie poleceniem `Generuj elementy aplikacji`. Podobnie jak dla sekcji załączników, dla sekcji podglądu również można określić ustawienie `Kategoria załączników`, które w tym wypadku nie jest wymagane. Wyświetlany jako placeholder dla braku wartości tekst `Wszystkie kategorii` informuje, że w tej sekcji wyświetlany jest podgląd załączników wszystkich kategorii, to jest ze wszystkich sekcji załączników na danym formularzu.



Sekcja typu `Załączniki – podgląd` w widoku formularza

Po kliknięciu przycisku `Ustawienia` obok listy rozwijanej `Typ sekcji` wyświetlane jest okno dialogowe, w którym można określić zachowanie sekcji podglądu po wykonaniu okreslonej akcji dotyczącej załącznika. Dostępne ustawienia:

- `Rodzaj okna dla podglądu`: dostępne ustawienia:
  - `Okno`: wyświetla podgląd załącznika w trybie okna z możliwością zmiany rozmiaru, przenoszenia, minimalizacji i maksymalizacji; w tym trybie można wyświetlać podgląd wielu załączników; okna podglądu są minimalizowane do pasków tytułu widocznych w lewym dolnym rogu okna przeglądarki, po minimalizacji czwartego okna, paski tytułów są grupowane do rozwijanego przycisku z listą okien podglądu.
  - `Dialog`: wyświetla podgląd załącznika w oknie dialogowym (oknie modalnym); w tym trybie nie ma możliwości minimalizacji/maksymalizacji okna ani wyświetlenia podglądu kolejnego załącznika.
- `Po dodaniu załącznika w tej sekcji podglądu załączników pokaż`: dostępne ustawienia:
  - `obecnie najnowszy` (domyślne),
  - `obecnie najstarszy`,
  - `nie zmieniaj załącznika (zachowaj ostatni wyświetlony, jeżeli jakiś był wyświetlany)`,
  - `nic nie wyświetlaj (zresetuj podgląd)`.
- `Po usunięciu załącznika w tej sekcji podglądu załączników pokaż`: dostępne ustawienia:
  - `nic nie wyświetlaj (zresetuj podgląd)` (domyślne),
  - `obecnie najnowszy`,
  - `obecnie najstarszy`.
- `Po otwarciu formularza w tej sekcji podglądu załączników pokaż`: dostępne ustawienia:
  - `nic nie wyświetlaj (zresetuj podgląd)` (domyślne),
  - `obecnie najnowszy`,
  - `obecnie najstarszy`.
- `Domyślne powiększenie podglądu załącznika`: określa domyślne powiększenie podglądu załącznika w formacie PDF.

Powiązane tematy:

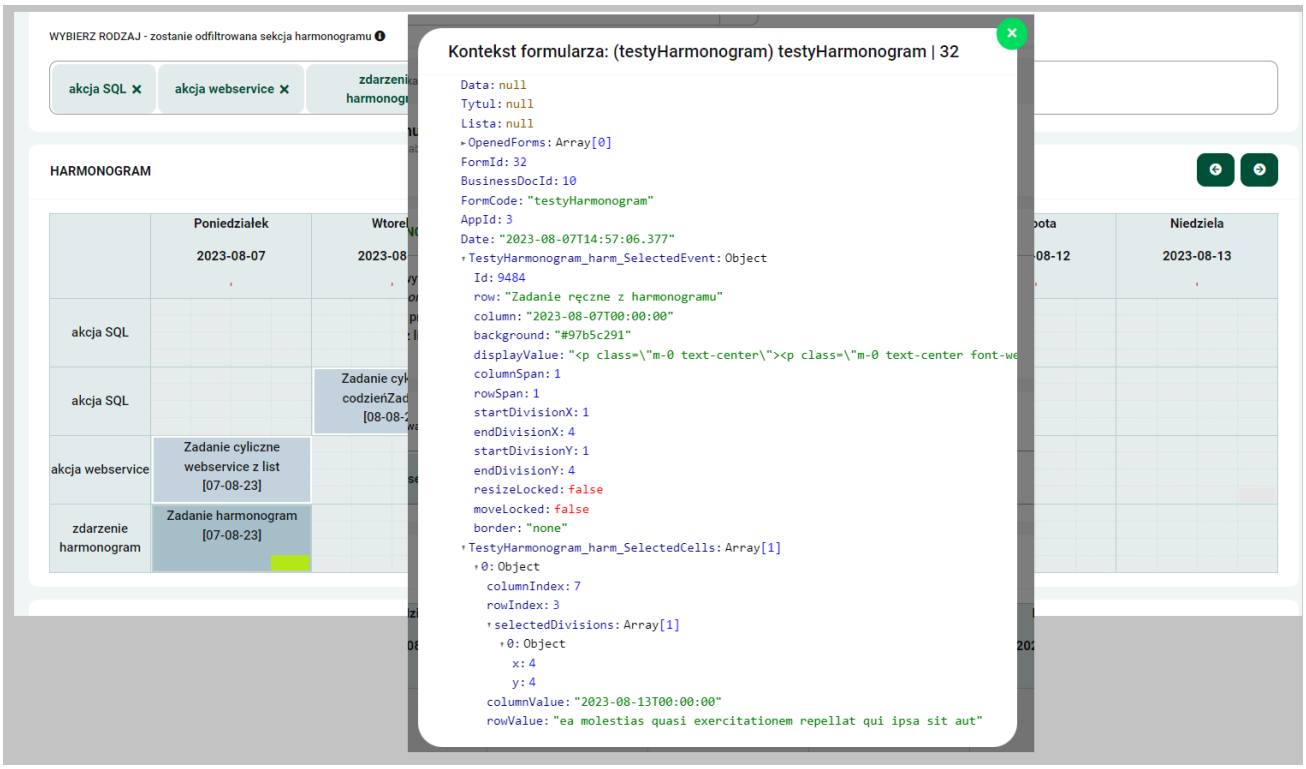
- Wspólne właściwości sekcji

4.1.5.10. HARMONOGRAM

Podsumowanie:
W sekcji typu harmonogram można zdefiniować interaktywną tabelę do wyświetlania danych (zdarzeń) w siatce dowolnie definiowanych wierszy i kolumn.

Skonfigurowanie harmonogramu wymaga określenia ustawień w oknie dialogowym `Ustawienia sekcji` oraz akcji dla czynności wykonywanych przez użytkownika w ramach interakcji z harmonogramem. Do tych interakcji należą: kliknięcie prawym przyciskiem myszy komórki w harmonogramie, zmiana zakresu zdarzenia przez przeciągnięcie obramowania zdarzenia, przeniesienie zdarzenia w inne miejsce na harmonogramie (przeciągnięcie z wciśniętym lewym przyciskiem myszy i lewym klawiszem ALT), zaznaczanie komórek harmonogramu przez klikanie lewym przyciskiem z wciśniętym przyciskiem CTRL.

Informacje o zaznaczonym zdarzeniu i/lub zaznaczonych komórkach są dostępne w kontekście formularza jako obiekt *KodSekcji_SelectedEvent* oraz jako tablica *KodSekcji_SelectedCells*, jak na ilustracji poniżej.



Informacje o zdarzeniu i zaznaczonych komórkach (komórka wyróżniona kolorem zielonym) w kontekście formularza

USTAWIENIA SEKCJI

Na panelu właściwości sekcji typu harmonogram kliknij przycisk `Ustawienia sekcji` i określ ustawienia w wyświetlonym oknie dialogowym:

- `Wysokość`: określa wysokość harmonogramu w sekcji, domyślnie *auto*.
- `Podział komórki w osi X`: określa na ile części mają być dzielone komórki harmonogramu w poziomie, domyślnie *1*.
- `Podział komórki w osi Y`: określa na ile części mają być dzielone komórki harmonogramu w pionie, domyślnie *1*.
- `Domyślny kolor komórki`: kolor pojedynczej komórki w harmonogramie, domyślnie *#fff*.
- `Wysokość komórki`: wysokość pojedynczej komórki w harmonogramie, domyślnie *60px*.
- `Szerokość komórki`: szerokość pojedynczej komórki w harmonogramie, domyślnie *150px*.
- `Kolor nagłówka kolumny`: kolor nagłówka kolumny, domyślnie '#e1edef'.
- `Kolor nagłówka wiersza`: kolor nagłówka wiersza, domyślnie '#e1edef'.
- `Wysokość nagłówka kolumny`: wysokość nagłówka kolumny, domyślnie *60px*.
- `Szerokość nagłówka wiersza`: szerokość nagłówka wiersza, domyślnie *100px*.
- `Wartość nagłówka przecinającego`: tekst wyświetlany dla nagłówka przecinającego kolumny z wierszami, domyślnie pusty.
- `Przenoszenie zdarzeń`: zezwala na przenoszenie zdarzeń w harmonogramie metodą *drag & drop* (z wciśniętym lewym klawiszem ALT), domyślnie wyłączone.
- `Zmiana rozmiaru zdarzenia w osi X`: jeśli przełącznik jest włączony, po najechaniu kursorem myszy na pionową krawędź obramowania komórki zdarzenia jest wyświetlana strzałka zmiany rozmiaru, a przy próbie zmiany rozmiaru jest wykonywana akcja przypisana we właściwości `Akcje dla zmiany rozmiaru`; jeśli komórki harmonogramu mają włączony podział w osi X, możliwa jest zmiana zakresu zdarzenia w tej osi w obrębie jednej komórki, aby można było zwiększyć zakres zdarzenia na sąsiednie kolumny, musi być włączony przełącznik `Rozciągaj zdarzenia dla kilku wierszy`.
- `Zmiana rozmiaru zdarzenia w osi Y`: jeśli przełącznik jest włączony, po najechaniu kursorem myszy na poziomą krawędź obramowania komórki zdarzenia jest wyświetlana strzałka zmiany rozmiaru, a przy próbie zmiany rozmiaru jest wykonywana akcja przypisana we właściwości `Akcje dla zmiany rozmiaru`; jeśli komórki harmonogramu mają włączony podział w osi Y, możliwa jest zmiana zakresu zdarzenia w tej osi w obrębie jednej komórki, aby można było zwiększyć

zakres zdarzenia na sąsiednie wiersze, musi być włączony przełącznik `Rozciągaj zdarzenia dla kilku wierszy`.

- `Zmiana rozmiaru zdarzenia w osi Y`: zezwala na zmianę zakresu zdarzenia w pionie (wiersze), domyślnie wyłączone.
- `Rozciągaj zdarzenia dla kilku kolumn`: pozwala zwiększać zakres zdarzenia na sąsiednie kolumny, przeciągając myszą pionową krawędź obramowania komórki.
- `Rozciągaj zdarzenia dla kilku wierszy`: pozwala zwiększać zakres zdarzenia na sąsiednie wiersze, przeciągając myszą poziomą krawędź obramowania komórki.
- `Wyświetlaj nagłówki kolumn`: włącza/wyłącza wyświetlanie nagłówków kolumn.
- `Wyświetlaj nagłówki wierszy`: włącza/wyłącza wyświetlanie nagłówków wierszy.
- `SQL dla kolumn`: zapytanie SQL zwracające nagłówki kolumn harmonogramu; w zapytaniu należy zwrócić następujące kolumny:
    - *displayValue* (string): wymagane; kod HTML renderowany jako tekst w nagłówku kolumny,
    - *value* (dowolny typ): wymagane; wartość identyfikująca kolumnę, do której przypisane jest zdarzenie,
    - *background* (string): opcjonalne; szesnastkowy kod koloru tła nagłówka kolumny, np. *#facc3a*,
    - *width* (string): opcjonalne; szerokość kolumny z dopiskiem jednostki, np. *300px*.

```
SELECT
CONCAT('<span class="m-0 text-center"><p>', d.[Name],'</p><p>',Format(d.[Date],'yyyy-MM-dd'
'<p class="m-0 text-center font-weight-bold text-danger" data-toggle="tooltip" data-placeme
) AS displayValue,
d.[Date] AS value
FROM (VALUES
(DATEADD(wk, DATEDIFF(wk, 0, @DataStart), 0), 'Poniedziałek'),
(DATEADD(wk, DATEDIFF(wk, 0, @DataStart), 1), 'Wtorek'),
(DATEADD(wk, DATEDIFF(wk, 0, @DataStart), 2), 'Środa'),
(DATEADD(wk, DATEDIFF(wk, 0, @DataStart), 3), 'Czwartek'),
(DATEADD(wk, DATEDIFF(wk, 0, @DataStart), 4), 'Piątek'),
(DATEADD(wk, DATEDIFF(wk, 0, @DataStart), 5), 'Sobota'),
(DATEADD(wk, DATEDIFF(wk, 0, @DataStart), 6), 'Niedziela')
) AS d ([Date], [Name])
```

- `SQL dla wierszy`: zapytanie SQL zwracające nagłówki wierszy harmonogramu; w zapytaniu należy zwrócić następujące kolumny:
    - *displayValue* (string): wymagane; kod HTML renderowany jako tekst w

nagłówku wiersza,
    - *value* (dowolny typ): wymagane; wartość identyfikująca wiersz, do którego przypisane jest zdarzenie,
    - *headerBackground* (string): opcjonalne; szesnastkowy kod koloru tła nagłówka wiersza, np. *#facc3a*,
    - *background* (string): opcjonalne; szesnastkowy kod koloru tła komórek wiersza, np. *#facc3a*,
    - *height* (string): opcjonalne; wysokość wiersza z dopiskiem jednostki, np. *300px*.
- `SQL dla zdarzeń`: zapytanie SQL zwracające dane umożliwiające prawidłowe wyświetlenie zdarzeń w harmonogramie; w zapytaniu należy zwrócić następujące kolumny:
    - *id* (dowolny typ): wymagane; identyfikator zdarzenia,
    - *column* (dowolny typ): wymagane; kolumna, do której przypisano zdarzenie (*value* z zapytania dla kolumn),
    - *columnSpan* (number): opcjonalne; liczba kolumn, które pokrywa zdarzenie; domyślnie 1,
    - *row* (dowolny typ): wymagane; wiersz, do którego przypisano zdarzenie (*value* z zapytania dla wierszy),
    - *rowSpan* (number): opcjonalne; liczba wierszy, które pokrywa zdarzenie; domyślnie 1,
    - *startDivisionX* (number): indeks pierwszej subkomórki w pierwszej komórce zdarzenia; określa początek zdarzenia w osi X w przypadku włączenia podziału komórek w osi X; przyjmuje wartości od 1 do *n*, gdzie *n* to wartość ustawienia `Podział komórki w osi X`,
    - *endDivisionX* (number): indeks ostatniej subkomórki w ostatniej komórce zdarzenia; określa koniec zdarzenia w osi X w przypadku włączenia podziału komórek w osi X; przyjmuje wartości od 1 do *n*, gdzie *n* to wartość ustawienia `Podział komórki w osi X`,
    - *startDivisionY* (number): indeks pierwszej subkomórki w pierwszej komórce zdarzenia; określa początek zdarzenia w osi Y w przypadku włączenia podziału komórek w osi Y; przyjmuje wartości od 1 do *n*, gdzie *n* to wartość ustawienia `Podział komórki w osi Y`,
    - *endDivisionY* (number): indeks ostatniej subkomórki w ostatniej komórce zdarzenia; określa koniec zdarzenia w osi Y w przypadku włączenia podziału komórek w osi Y; przyjmuje wartości od 1 do *n*, gdzie *n* to wartość ustawienia `Podział komórki w osi Y`,
    - *displayValue* (string): wymagane; kod HTML renderowany jako opis zdarzenia,
    - *background* (string): wymagane; szesnastkowy kod koloru tła komórki

zdarzenia, np. *#facc3a*,

- *border* (string): opcjonalne; obramowanie komórki zdarzenia, podawane jako CSS, np. *1px #d4d4d4 solid*,
- *resizeLocked* (bit): opcjonalne; blokada zmiany rozmiaru zdarzenia; domyślnie *0*,
- *moveLocked* (bit): opcjonalne; blokada przenoszenia zdarzenia; domyślnie *0*,
- *isEmptyEvent* (bit): opcjonalne; identyfikator zdarzenia „pustej" komórki, umożliwia przypisanie akcji we właściwości `Akcja dla pustego zaznaczenia`.

```
SELECT
f.Id AS [id],
f.MachineId AS [column],
DATEADD(
dd,
DATEDIFF(dd, 0, f.[FailureDate]),
0
) AS [row],
'#fac0d4' AS [background],
CONCAT(
'<p class="m-0 text-center">', f.Description,
' (', f.Code, ')', '</p>'
) AS displayValue
FROM
dev.MachineFailures AS f
```

AKCJE HARMONOGRAMU

Aby prawidłowo korzystać z harmonogramu, należy skonfigurować zestaw akcji wykonywanych podczas interakcji użytkownika z harmonogramem:

- `Akcje dla pustego zaznaczenia`: akcje wykonywane po kliknięciu pustej komórki prawym przyciskiem myszy i wybraniu nazwy akcji do wykonania; na przykład zapytanie SQL dodające wpis do tabeli zdarzeń harmonogramu z nowym zdarzeniem:

```
INSERT INTO [dbo].[ZdarzeniaHarmonogramu] (
TypZdarzenia, Opis, Zakres, DataWykonania
)
values
(
  {& Testy.Dictionaries.EventsTypes.MyEvent},
```

```
  {@_Section_Description},
  {@_Section_Scope},
  {@_Section_Date}
)
```

- `Akcje dla zdarzeń`: akcje wykonywane po kliknięciu komórki zdarzenia prawym przyciskiem myszy i wybraniu nazwy akcji do wykonania; może to być na przykład akcja Otwórz formularz, która otworzy formularza z informacjami o zdarzeniu; identyfikator bieżącego zdarzenia należy wskazać w akcji za pomocą zmiennej *{@KodSekcjiHarmonogramu_SelectedEvent.Id}* wpisanej w polu dla opcji `wskazany dokument`.
- `Akcje dla zmiany rozmiaru`: akcje wykonywane po zmianie rozmiaru zdarzenia, np. akcja aktualizacji informacji o nowym zakresie zdarzenia.
- `Akcje dla przenoszenia`: akcje wykonywane po przeniesieniu zdarzenia metodą *drag & drop*.

Powiązane tematy:

- Otwieranie formularza
- Wspólne właściwości sekcji

4.1.5.11. SEKCJA HTML

Podsumowanie:
W sekcji typu HTML można renderować na formularzu kod HTML. W kodzie można odwoływać się do wartości z formularza przy użyciu zmiennych *{@kodPola}*, gdzie *kodPola* to kod pola na formularzu. Ponadto w kodzie HTML można używać wyników zwracanych przez zapytanie SQL oraz kodu JavaScript.

W sekcji HTML jest także obsługiwana biblioteka chart.js, która pozwala na generowanie wykresów na podstawie danych zwracanych w zapytaniu SQL. Zasady tworzenia wykresów przy użyciu tej biblioteki opisano w osobnym temacie.

Ustawienia sekcji

Zależność od sekcji

Kod zależnej sekcji

Kod zależnej sekcji

Zezwól na parsowanie HTML z pól formularza (unsecure)

Edytor SQL Query

Aby skorzystać z danych formularza użyj składni: **{@NazwaPola}**

Tables

[dbo].[Covid]

[dbo].[CustomAcceptance

[dbo].[Rozliczenia]

Odśwież

Sprawdź składnię SQL

```
1  -- SELECT [FirstName], [LastName] from core.UserProfiles whe
2
3  SELECT '<span>raz, dwa, trzy...</span>' as html
```

Edytor HTML

Aby skorzystać z danych formularza użyj składni: **{@NazwaPola}**
Aby wyświetlić wiersze zwrócone z zapytania SQL musisz stworzyć szablon używając elementu: **<QueryRowTemplate>**
Jeśli sekcja zależna jest listą, aby wyświetlić jej wiersze musisz stworzyć szablon używając elementu: **<DSRowTemplate>**
Aby skorzystać z danych w zależnej sekcji w szablonie użyj składni: **{@DS_NazwaPola}**

```
1  <html>
2      <body>
3        <p>Sekcja HTML/JS!</p>
4        <QueryRowTemplate>{@Query_html}<!--{@Query_FirstName} {@Query_LastName}-->
5      </body>
6  </html>
```

Edytor JavaScript

Aby skorzystać z danych w formularzu użyj zmiennej o nazwie: **FormDataContext**
Aby skorzystać z danych zwróconych w zależnej sekcji użyj zmiennej o nazwie: **DSDataContext**
Aby skorzystać z danych z zapytania SQL w szablonie użyj składni: **{@Query_NazwaPola}**

```
1
```

Zamknij

Ustawienia sekcji typu `Sekcja HTML`

`Zależność od sekcji`: wyświetla pole, w którym należy wpisać kod sekcji, od której ma zależeć sekcja HTML; zależność powoduje, że zawartość sekcji HTML jest odświeżana dynamicznie po zmianie danych w sekcji nadrzędnej. Jeśli sekcja nadrzędna jest listą, wyświetlenie wierszy tej listy wymaga użycia w szablonie HTML elementu *<DSRowTemplate>*. Wyświetlenie danych z wybranych kolumn tej listy wymaga użycia składni *{@DS_kodKolumny}*, gdzie *kodKolumny* to kod kolumny listy wyświetlanej w sekcji nadrzędnej.

`Zezwól na parsowanie HTML z pól formularza (unsecure)`: umożliwia renderowanie w sekcji kodu HTML zwróconego przez SQL (np. z innych pól formularza).

`Edytor SQL Query`: umożliwia napisanie zapytania, które zwróci dane do wyświetlenia w sekcji; do wyników zapytania można się odwoływać w szablonie HTML, używając elementu *<QueryRowTemplate>* oraz zmiennych *{@Query_nazwaKolumny}, gdzie *nazwaKolumny* to nazwa kolumny zwróconej przez zapytanie.

```
SELECT [FirstName]
       ,[LastName]
FROM core.UserProfiles
WHERE Id = {@_UserId}
```

`Edytor HTML`: pole do wpisania kodu HTML, który ma być wyświetlany w sekcji; kliknięcie kółka ze znakiem zapytania wyświetla okienko z przykładem kodu HTML. Aby wyświetlić wartości z rekordów sekcji zależnej typu lista, należy użyć elementu *<DSRowTemplate>* oraz zmiennej *{@DS_ColumnName}*, gdzie *ColumnName* to nazwa kolumny listy; patrz przykład poniżej

```
<p id="docNumber">Numer dokumentu</p>
<QueryRowTemplate>{@Query_FirstName} {@Query_LastName}</QueryRowTemplate>
```

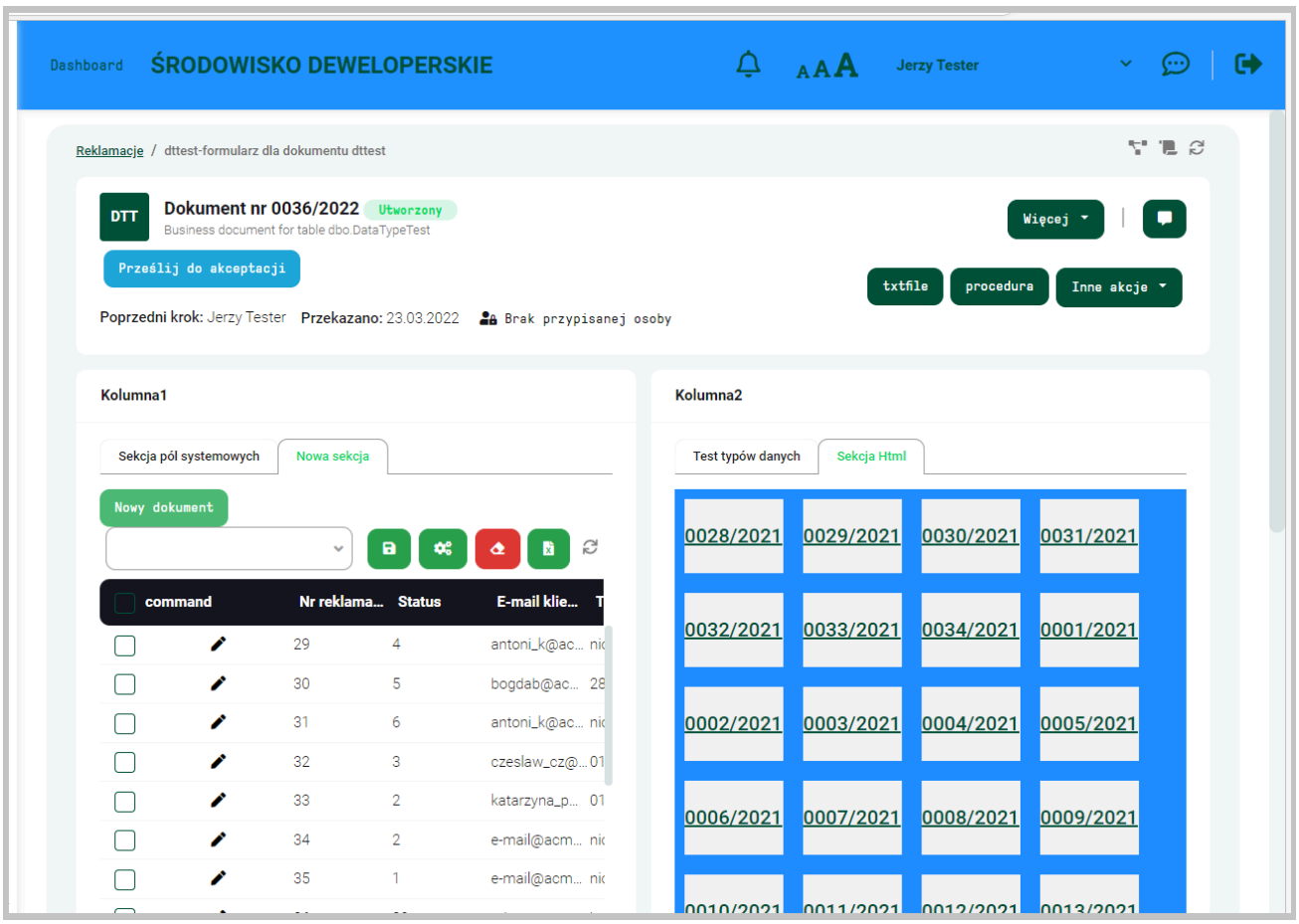
`Edytor JavaScript`: wyświetla pole, w którym można wpisać kod JavaScript z odwołaniem do wartości z kontekstu formularza (`FormDataContext.ContextFieldName`), pola rekordu w sekcji zależnej typu lista lub aktualizujący stan kontrolki na formularzu. W przypadku gdy wskazana sekcja zależna to sekcja typu lista, odwołanie ma postać: `DSDataContext[n].ColumnName`, gdzie *n* to numer rekordu na liście, a *ColumnName* to nazwa kolumny listy.

```
document.getElementById("docNumber").innerHTML = FormDataContext.Code;
```

PRZYKŁAD

Poniższy przykład pozwala wyświetlić w sekcji HTML kafelki typu *flexbox* z linkami do dokumentów z listy w sekcji zależnej.

```
<style>
.flex-container {
  display: flex;
  flex-wrap: wrap;
  background-color: DodgerBlue;
}

.flex-container > div {
  background-color: #f1f1f1;
  width: 100px;
  margin: 10px;
  text-align: center;
  line-height: 75px;
  font-size: 20px;
}
</style>
</head>
<body>
  <div class="flex-container">
    <DSRowTemplate>
      <div>
        <a href="https://naxiom_url/front/form/59CEB...DE12/{@DS_Id}">{@DS_Code}</a>
      </div>
    </DSRowTemplate>
  </div>
</body>
```



Przykład sekcji HTML

Powiązane tematy:

- Wspólne właściwości sekcji
- Obsługa biblioteki chart.js

4.1.5.12. SEKCJA KALENDARZ

Podsumowanie:
Kalendarz pozwala wyświetlać dane ze źródła danych w jednym z widoków kalendarza: miesięcznym, tygodniowym, dziennym lub osi czasu. Możliwa jest także edycja tych danych: zmiana czasu rozpoczęcia/zakończenia techniką *drag and drop* lub przez formularz do edycji tych danych. Wpisy w kalendarzu to *zdarzenia*.

W TYM TEMACIE

- Źródło danych dla zdarzeń
- Grupowanie zdarzeń
- Pokaż od dnia
- Dni wolne
- Tryby wyświetlania
- Interwał czasu dla widoku dziennego
- Opcje wyświetlania
- Pozostałe ustawienia sekcji kalendarza
- Obsługa kalendarza w aplikacji FrontSPA

Aby zamieścić kalendarz na formularzu, dodaj nową sekcję i na panelu właściwości wybierz pozycję `Kalendarz` z listy rozwijanej `Typ sekcji`. Do poprawnego działania sekcji kalendarz potrzebna jest tabela w bazie danych, która będzie zawierać dane zdarzeń, oraz formularz edycji zdarzenia wyświetlany po kliknięciu komórki kalendarza. Ponadto, oprócz parametrów konfiguracyjnych definiowanych po kliknięciu przycisku `Ustawienia` obok listy `Typ sekcji`, należy zdefiniować akcję usuwania zdarzenia i akcję aktualizacji zdarzenia we właściwościach sekcji.

Ustawienia wspólne dla wszystkich rodzajów sekcji opisano w rozdziale Wspólne właściwości sekcji .

W celu konfiguracji kalendarza zaznacz sekcję kalendarza na formularzu i kliknij przycisk `Ustawienia` obok listy `Typ sekcji` na panelu właściwości. Zostanie wyświetlone okno dialogowe `Ustawienia sekcji kalendarza`, w którym poszczególne ustawienia (okna edycji SQL) zostały dla wygody przedstawione jako nagłówki rozwijanych sekcji. Kliknij nagłówek, aby rozwinąć ustawienia.

ŹRÓDŁO DANYCH DLA ZDARZEŃ

`SQL dla wydarzeń`: zapytanie SQL, które musi zwrócić następujące kolumny z tabeli, w której mają być zapisywane zdarzenia:

- `id` (*varchar*): identyfikator zdarzenia
- `start` (*datetime*): data i godzina rozpoczęcia zdarzenia
- `end` (*datetime*): data i godzina zakończenia zdarzenia
- `title` (*varchar*): tytuł zdarzenia
- `isAllDay` (*bit*): flaga zdarzenia całodniowego

- `color` (*varchar*): (kolumna opcjonalna) szesnastkowa wartość koloru dla komórki kalendarza w formacie *#FFFFFF*

Zależnie od wybranego widoku, zdarzenie jest prezentowane na kalendarzu jako prostokątny obszar o rozmiarach odpowiadających czasowi rozpoczęcia i zakończenia opisany tytułem zdarzenia. Zdarzenia oznaczone jako całodniowe zajmują całą komórkę kalendarza.

Zapytanie może także zwracać inne kolumny z tabeli źródłowej, które będą dodatkowo opisywać zdarzenie. Edycja wartości w tych kolumnach będzie odbywać się w formularzu przypisanym we właściwości `Formularz zdarzenia`.

PRZYKŁAD

```
SELECT InspectionId AS [id],
       InspectionBegins AS [start],
       InspectionEnds AS [end],
       InspectionTitle AS [title],
       InspectionGoal,
       InspectorId,
       FieldInspection AS [isAllDay]
FROM dbo.InspectionsSchedule
```

GRUPOWANIE ZDARZEŃ

`Włącz grupowanie`: włącza/wyłącza grupowanie zdarzeń według wartości z kolumny zwróconej przez zapytanie SQL; domyślnie włączony.

`Nazwa kolumny...`: nazwa kolumny zwraconej przez zapytanie źódłowe, której wartości zostaną użyte do grupowania zdarzeń.

Okno edycji SQL: wpisz zapytanie SQL, które zwróci **unikatowe** wartości w następujących kolumnach:

- `text` (*varchar*): wartość wyświetlana dla grup zdarzeń
- `value` (*varchar*): wartość, według której zdarzenia będą grupowane
- `color` (*varchar*): (kolumna opcjonalna) szesnastkowa wartość koloru, który będzie używany dla zdarzeń z danej grupy w formacie *#FFFFFF*

PRZYKŁAD

```sql
SELECT CONCAT(MIN(FirstName), ' ', MIN(LastName)) AS [text],
       dbo.Zamowienia.UserId as [value]
FROM core.UserProfiles
RIGHT JOIN dbo.Zamowienia
ON core.UserProfiles.Id = dbo.Zamowienia.UserId
GROUP BY dbo.Zamowienia.UserId -- wymagane użycie klauzuli GROUP BY
```

POKAŻ OD DNIA

`SQL dla daty dla której kalendarz będzie ustawiony`: zapytanie zwracające datę, dla której zostanie ustawiony początkowy okres, zależnie od trybu wyświetlania, np. widok miesięczny będzie się zaczynał od poniedziałku w tygodniu obejmującym początek miesiąca, w którym przypada zwrócona data.

PRZYKŁAD

```sql
SELECT DATEADD(wk, DATEDIFF(wk,0,GETDATE()), 0) AS OstatniPoniedzialek
```

DNI WOLNE

`SQL dla dat uznanych jako dni wolne`: zapytanie zwracające listę dat i nazw dni wolnych w kolumnach:

- `value` (*date*): data dnia wolnego
- `text` (*varchar*): nazwa dnia wolnego wyświetlana w komórce kalendarza

PRZYKŁAD

```sql
SELECT HolidayDate as [value],
       HolidayName as [text]
FROM dbo.HolidaysWorldWide
WHERE CountryAlpha2Code = 'PL'
```

TRYBY WYŚWIETLANIA

Lista wielowyboru: wybierz tryby wyświetlania kalendarza, które mają być dostępne w aplikacji *FrontSPA*; dostępne wartości:

- `Miesięczny`: tablica sześciu tygodni ustawionych w rzędach, jeden pod drugim (7 x 6); pierwsza komórka to poniedziałek (zawsze) tygodnia z pierwszym dniem miesiąca lub datą `Pokaż od dnia`
- `Tygodniowy`: ustawione obok siebie kolumny poszczególnych dni tygodnia (domyślnie siedem) podzielone na interwały godzinowe; domyślnie jako pierwszy dzień tygodnia jest wyświetlany dzień bieżący lub dzień `Pokaż od dnia`; aby wyświetlać widok tygodniowy od określonego dnia tygodnia, należy w opcji `Pokaż od dnia` użyć zapytania SQL

  ```
  SELECT DATEADD(dd,-(DATEPART(WEEKDAY, GETDATE())+1),GETDATE())
  -- zwraca datę ostatniego piątku
  ```

- `Dzienny`: jedna kolumna dnia o szerokości całego kalendarza podzielona na interwały godzinowe; dzień bieżący lub dzień `Pokaż od dnia`
- `Oś czasu`: jeden wiersz dni, podzielonych lub nie na interwały godzinowe; zawsze wyświetlany od pierwszego dnia bieżącego miesiąca lub miesiąca, w którym przypada data `Pokaż od dnia`.

Tryb wybrany jako pierwszy jest ustawiany jako domyślny; kolejność trybów na liście wielowyboru nie odpowiada kolejności wyboru trybów na liście rozwijanej — jest ona taka, jak podana powyżej. Jeśli widok dzienny nie zostanie dodany do listy, nie będzie można przechodzić do widoku dziennego po kliknięciu wielokropka w widoku miesięcznym.

INTERWAŁ CZASU DLA WIDOKU DZIENNEGO

`Interwał (w minutach)`: wartość określająca podział dnia w widoku dziennym, tygodniowym i osi czasu na interwały godzinowe; dozwolone wartości od 15 do 720 minut.

OPCJE WYŚWIETLANIA

`Wyświetlana liczba dni tygodnia`: liczba dni wyświetlanych w widoku tygodniowym (łącznie z dniem bieżącym); dozwolone wartości od 1 do 7.

`Oś czasu – interwał całodniowy`: po włączeniu tego przełącznika dni w widoku osi czasu nie będą dzielone na interwały godzinowe.

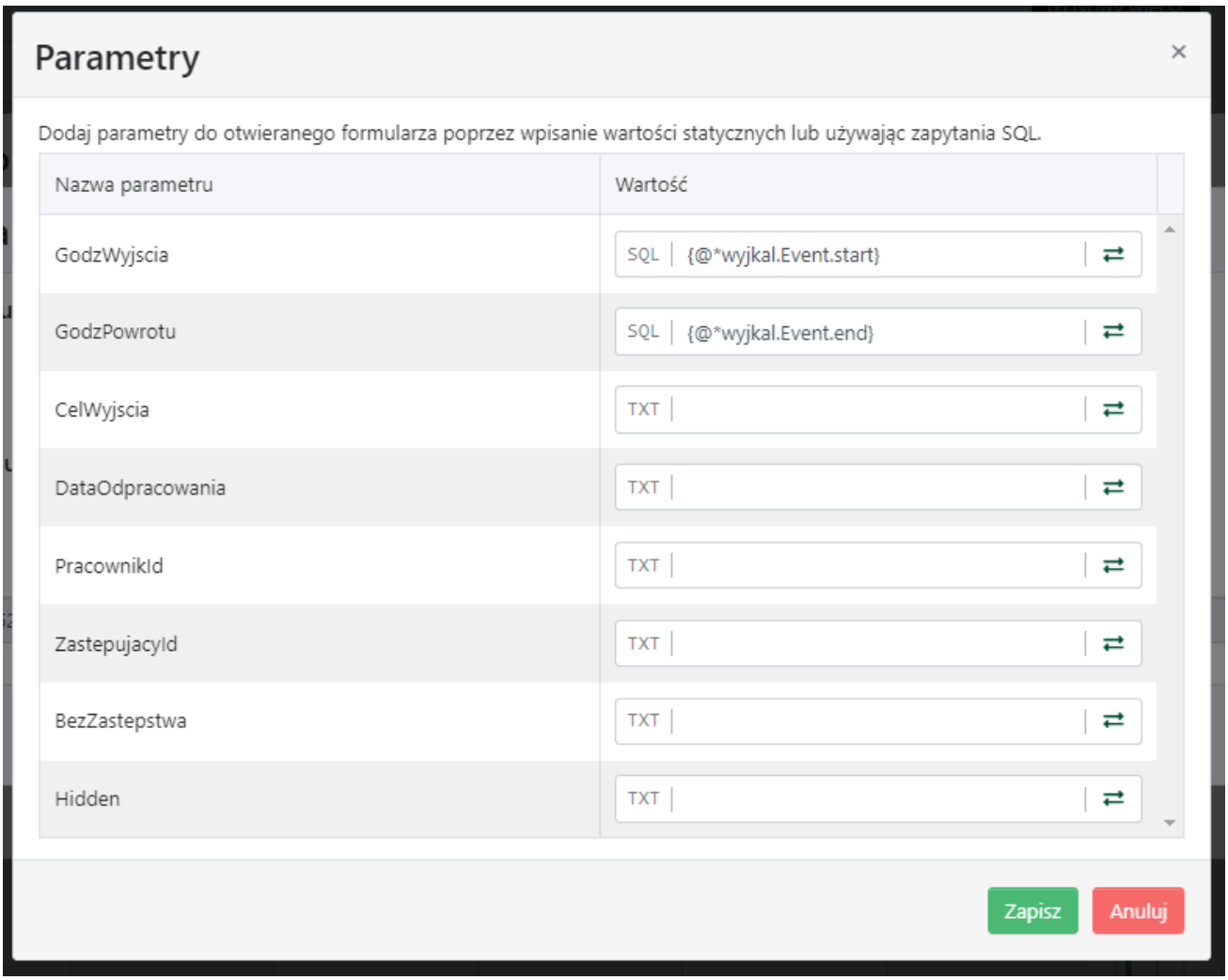
POZOSTAŁE USTAWIENIA SEKCJI KALENDARZA

W ramach konfiguracji kalendarza, należy określić jeszcze następujące ustawienia dostępne na panelu właściwości:

`Formularz zdarzenia`: formularz wyświetlany po kliknięciu komórki w kalendarzu; kliknij przycisk `Ustaw`, aby wyświetlić okno dialogowe `Wybierz akcję do wywołania` i wybierz, a następnie skonfiguruj akcję `Otwórz formularz`; aby dla nowego zdarzenia automatycznie wypełniać pole rozpoczęcia i zakończenia zgodnie z wartościami dla kliknięte j komórki formularza, użyj odwołań *{@*KodFormularza.Event.start}* i *{@*KodFormularza.Event.end}*, gdzie *KodFormularza* to kod formularza z kalendarzem.

UWAGA

Akcja otwierania formularza (w trybie *dialog*) jest wykonywana także podczas odświeżania formularza, który został otwarty przy użyciu tej akcji. Jednak w tym przypadku kontekstem domyślnym jest kontekst formularza docelowego. Dlatego ważne jest odpowiednie skonfigurowanie parametrów odwołujących się do kontekstu formularza źródłowego (tego, z którego jest inicjowana akcja).
Właściwa składnia to *{@*KodFormularza.PoleFormularza}*. Alternatywnie można skorzystać z funkcji
JSON_VALUE({@OpenedForms},'$[0].PoleFormularza')
Należy przy tym pamiętać (w obu przypadkach) o konieczności przestawienia trybu pola wprowadzania na *SQL*.
Zastosowanie zmiennej *{@PoleFormularza}* spowoduje błąd lub nieoczekiwane zachowanie po odświeżeniu formularza docelowego, ponieważ system będzie szukał wartości takiej zmiennej w kontekście odświeżanego formularza.



Przekazywanie parametrów z kalendarza do formularza zdarzenia

`Akcja aktualizacji`: akcja wykonywana po zmianie daty i godziny rozpoczęcia/zakończenia zadania metodą *drag and drop* w widoku kalendarza; na przykład:

```
UPDATE [dbo].[WyjsciaRejestr]
SET GodzWyjscia = JSON_VALUE({@Event}, '$.start'),
    GodzPowrotu = JSON_VALUE({@Event}, '$.end')
WHERE [dbo].[WyjsciaRejestr].Id = JSON_VALUE({@Event}, '$.id')
```

UWAGA
W definicji akcji należy włączyć przełącznik `Przekaż wszystkie parametry jako nvarchar`.

`Akcja usunięcia`: akcja wykonywana po kliknięciu symbolu ***x*** w prawym górnym rogu prostokąta zdarzenia; na przykład

```
DELETE FROM [dbo].[WyjsciaRejestr]
```

```
WHERE [dbo].[WyjsciaRejestr].Id = JSON_VALUE({@Event}, '$.id')
```

> **UWAGA**
> W definicji akcji należy włączyć przełącznik `Przekaż wszystkie parametry jako nvarchar`.

- `Zależność od sekcji`: pozwala określić sekcję zawierającą pole, którego wartości będą mogły zostać użyte do filtrowania listy; możliwe jest także filtrowanie bieżącej listy zależnie od rekordów zaznaczonych we wskazanej sekcji typu lista; ponadto zależność powoduje, że sekcja zależna będzie odświeżana każdorazowo po zmianie danych w sekcji nadrzędnej.

- `Warunek dla zależności`: wyrażenie SQL filtrujące rekordy sekcji typu lista, zależnie od wartości pola na innej sekcji; możliwe są dwa rodzaje warunków:
  - *KolumnaListy operator {@_Section_PoleFormularza}*: *KolumnaListy* to nazwa kolumny listy wyświetlanej w sekcji, *@_Section_* to przedrostek, a *PoleFormularza* to właściwość *Pole formularza* pola w sekcji wskazanej jako nadrzędna. Wartość w tym polu jest używana do filtrowania rekordów listy.
  - *KolumnaListy IN (SELECT [value] FROM OPENJSON({@_Section_KodSekcji_Selected}))*: w tym przypadku zmienna *{@_Section_KodSekcji_Selected}* zwraca tablicę JSON (*[Id1, Id2...]*) z identyfikatorami rekordów zaznaczonych w sekcji nadrzędnej.

W edytorze SQL obsługiwany jest mechanizm *smart numbers*. Po kliknięciu przycisku `Podgląd zapytania SQL` wyświetlane jest okno dialogowe edytora, w którym można sprawdzić poprawność składni. Aby to było możliwe, w obszarze u góry okna wyświetlana jest w trybie tylko do odczytu część zapytania dla zależności w formie:

```
SELECT * FROM (SELECT * FROM List_View) WHERE...
```

Poniżej jest wyświetlany sam warunek.

Aby sprawdzić poprawność składni, należy w polach z prawej strony podać wartości parametrów (*{@Nazwa}*) występujących w zapytaniu. Dodatkowo, po kliknięciu przycisku `Wykonaj zapytanie` zostanie wyświetlone okno edytora SQL z całym zapytaniem warunku zależności. Po podstawieniu za parametry przykładowych wartości i ponownym kliknięciu przycisku `Wykonaj zapytanie` w dolnej części okna zostaną wyświetlone wyniki zwrócone przez zapytanie.

PRZYKŁAD

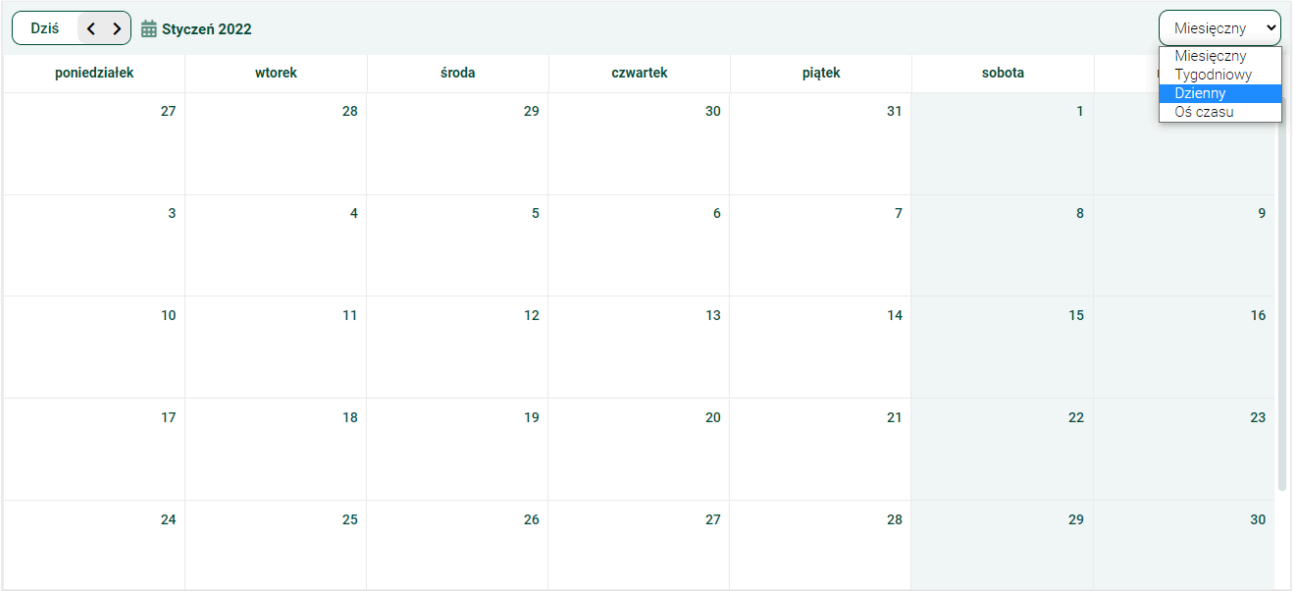
Poniższy warunek zależności spowoduje wyświetlenie na liście zależnej tylko tych rekordów, które zostały zaznaczone na liście nadrzędnej. *KodSekcji* to kod sekcji nadrzędnej (wybranej z listy `Zależność od sekcji`).

> **UWAGA**
> W odwołaniu *@_Section_**KodSekcji**_Selected* kod sekcji musi się zaczynać od wielkiej litery, niezależnie od tego, od jakiej litery zaczyna się kod sekcji nadrzędnej.

```
[Id] IN
(SELECT [value]
FROM
OPENJSON({@_Section_KodSekcji_Selected}))
```

OBSŁUGA KALENDARZA W APLIKACJI FRONTSPA

W aplikacji *FrontSPA* kalendarz jest wyświetlany w widoku skonfigurowanym jako domyślny. W nagłówku od lewej strony znajduje się przycisk `Dziś`, który wyświetla interwał obejmujący bieżącą datę, strzałki służące do zmiany wyświetlanego interwału (następny/poprzedni dzień/tydzień/miesiąc), określenie bieżącego interwału oraz listę rozwijaną z dostępnymi widokami kalendarza. Jeśli ta lista nie jest dostępna, to znaczy, że widok domyślny jest jedynym dostępnym.



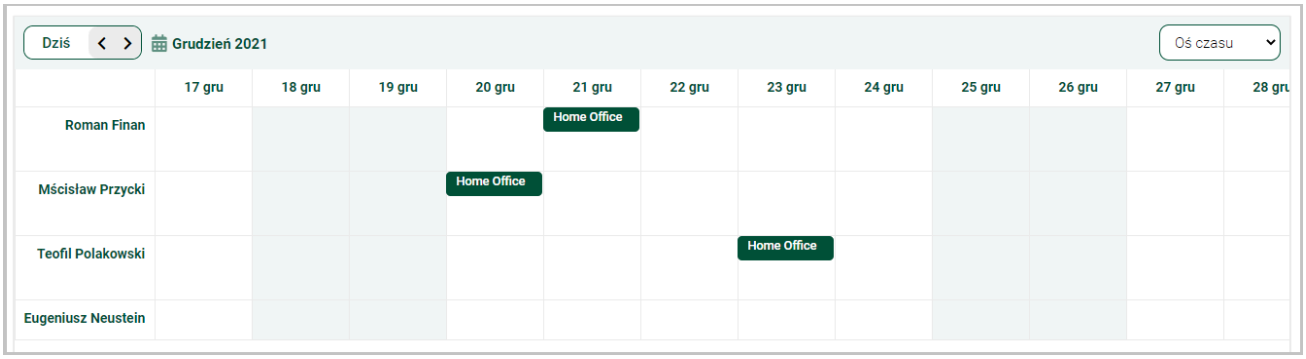
Widok miesięczny

Kliknięcie pustej komórki kalendarza wyświetla formularz, w którym można wpisać dane nowego zdarzenia. Kliknięcie zdarzenia w kalendarzu pozwala edytować to zdarzenie.

Zdarzenia są wyświetlane w kalendarzu jako prostokątne obszary. Po wskazaniu zdarzenia myszą w widoku dziennym i tygodniowym wyświetlane są na górnej i dolnej krawędzi uchwyty do zmiany rozmiaru. Ich przeciąganie pozwala zmieniać czasrozpoczęcia lub zakończenia zdarzenia. Symbol ***x*** wyświetlany w prawym górnym rogu zdarzenia służy do usunięcia zdarzenia. Dodatkowo zdarzenie można przeciągnąć myszą do innej komórki, zmieniając jednocześnie czas rozpoczęcia i zakończenia zdarzenia.

Jeśli w tym samym dniu zostanie zdefiniowanych w tym samym dniu, w widoku miesięcznym będzie obok nich wyświetlany przycisk z wielokrokiem, którego kliknięcie spowoduje przejście do widoku dziennego.

Po skonfigurowaniu i włączeniu grupowania, wybrany widok kalendarza będzie wyświetlany dla każdej grupy osobno. Grupowanie jest przydatne w widoku osi czasu.



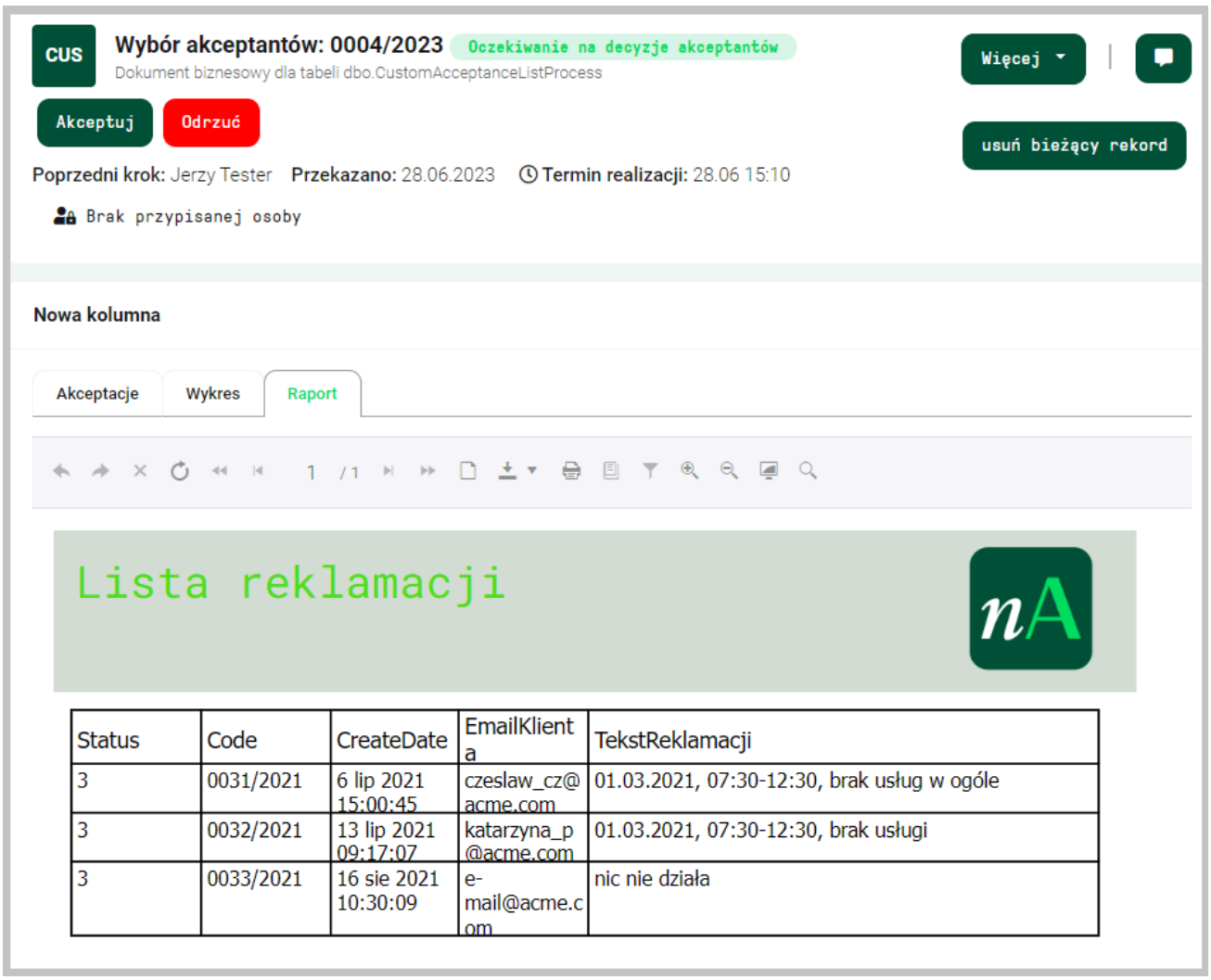
Widok osi czasu z grupowaniem

Powiązane tematy:

- Wspólne właściwości sekcji
- Akcja otwierania formularza
- Akcja SQL do bazy systemowej

4.1.5.13. RAPORT

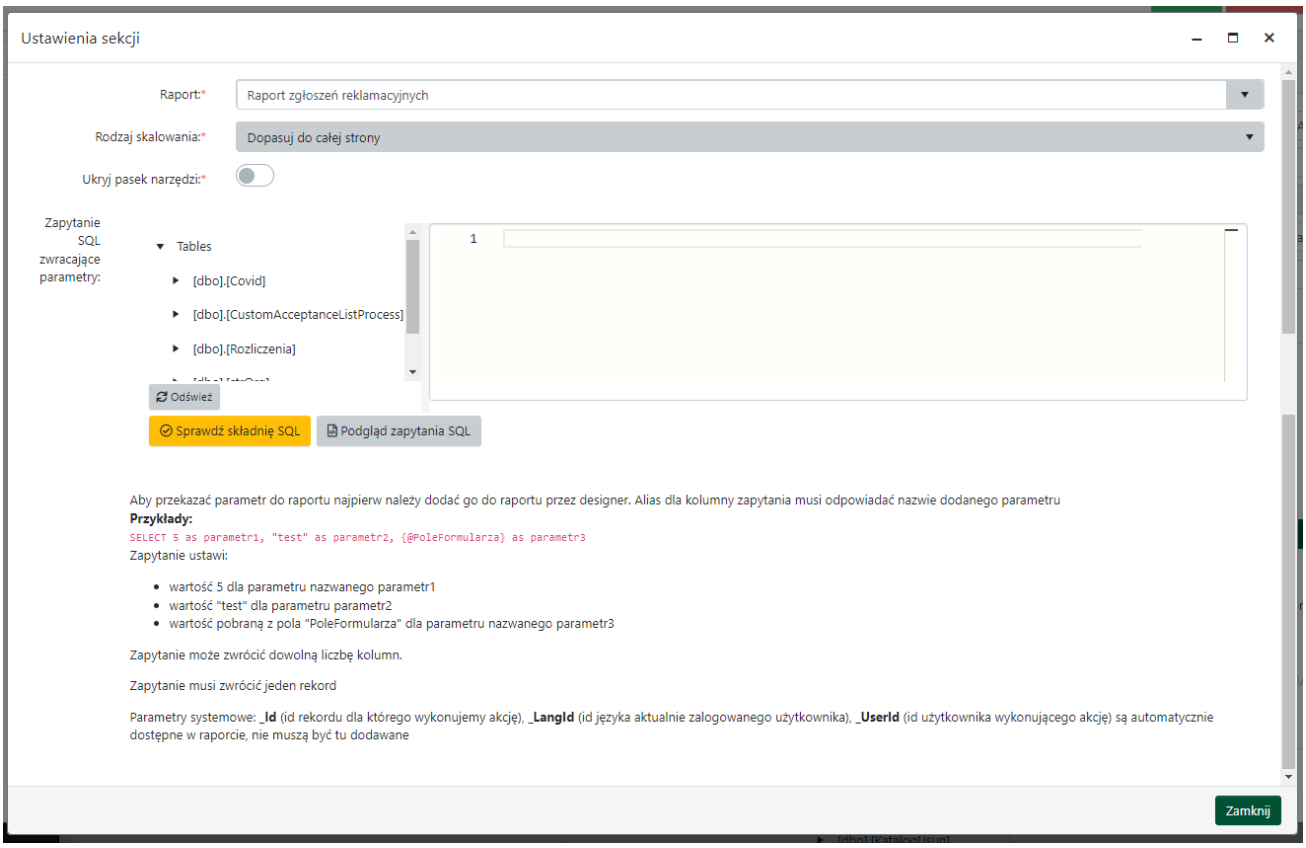
Podsumowanie:
W tej sekcji można wyświetlić raport wyrenderowany na podstawie szablonu zdefiniowanego w module Telerik Web Report Designer.



Sekcja typu raport na podglądzie formularza

Oprócz właściwości współdzielonych ta sekcja wymaga ustawienia następujących właściwości:

- `Raport`: wybierz z listy szablon raportu.
- `Rodzaj skalowania`: dostępne warości:
  - `Dopasuj do szerokości strony`
  - `Dopasuj do całej strony`
  - `Użyj skalowania procentowego`: wymaga określenia w procentach szerokości raportu względem szerokości strony.
- `Ukryj pasek narzędzi`: pozwala ukryć standardowy pasek narzędzi wyświetlany na podglądzie raportu.
- `Zapytanie SQL zwracające parametry`: pozwala określać wartości parametrów zdefiniowanych w szablonie raportu; zapytanie musi zwrócić jeden rekord, a zwrócone wartości parametrów (jeśli jest ich więcej) muszą mieć aliasy *parametr1*, *parametr2* itd.



Ustawienia sekcji raport

Powiązane tematy:

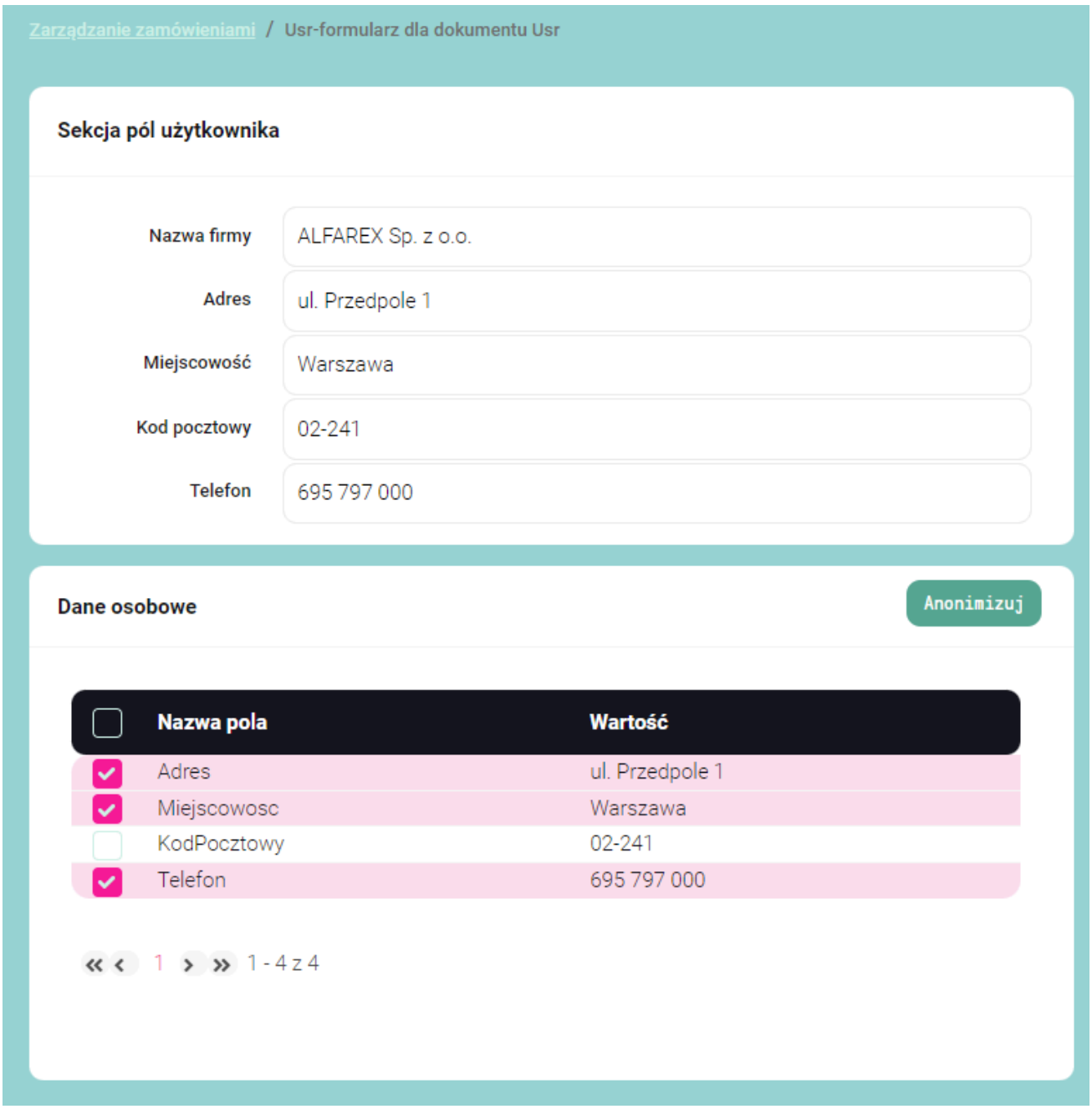
- Wspólne właściwości sekcji
- Raporty

4.1.5.14. SEKCJA DANYCH RODO

Podsumowanie:
Sekcja typu RODO służy do anonimizacji wybranych danych oznaczonych jako dane osobowe (kolumny tabel dokumentu biznesowego z zaznaczonym polem wyboru `Dane wrażliwe`). Pozwala ona anonimizować dane indywidualnie dla każdego dokumentu biznesowego.

Aby umożliwić anonimizację danych osobowych na formularzu, dodaj nową sekcję i na panelu właściwości wybierz pozycję `Sekcja danych RODO` z listy rozwijanej `Typ sekcji`. Następnie należy dodać przycisk na apsku `Toolbar` i przypisać mu akcję systemową `Anonimizuj dane`.

W aplikacji produkcyjnej sekcja danych RODO wyświetla listę kolumn tabeli źródłowej formularza oraz ich wartości dla bieżącego dokumentu. Z lewej strony w każdym wierszu znajduje się pole wyboru. Aby wykonać anonimizację danych, zaznacz pola wyboru, obok danych, które chcesz zanonimizować, i kliknij przycisk w nagłówku sekcji.



Sekcja danych RODO na formularzu

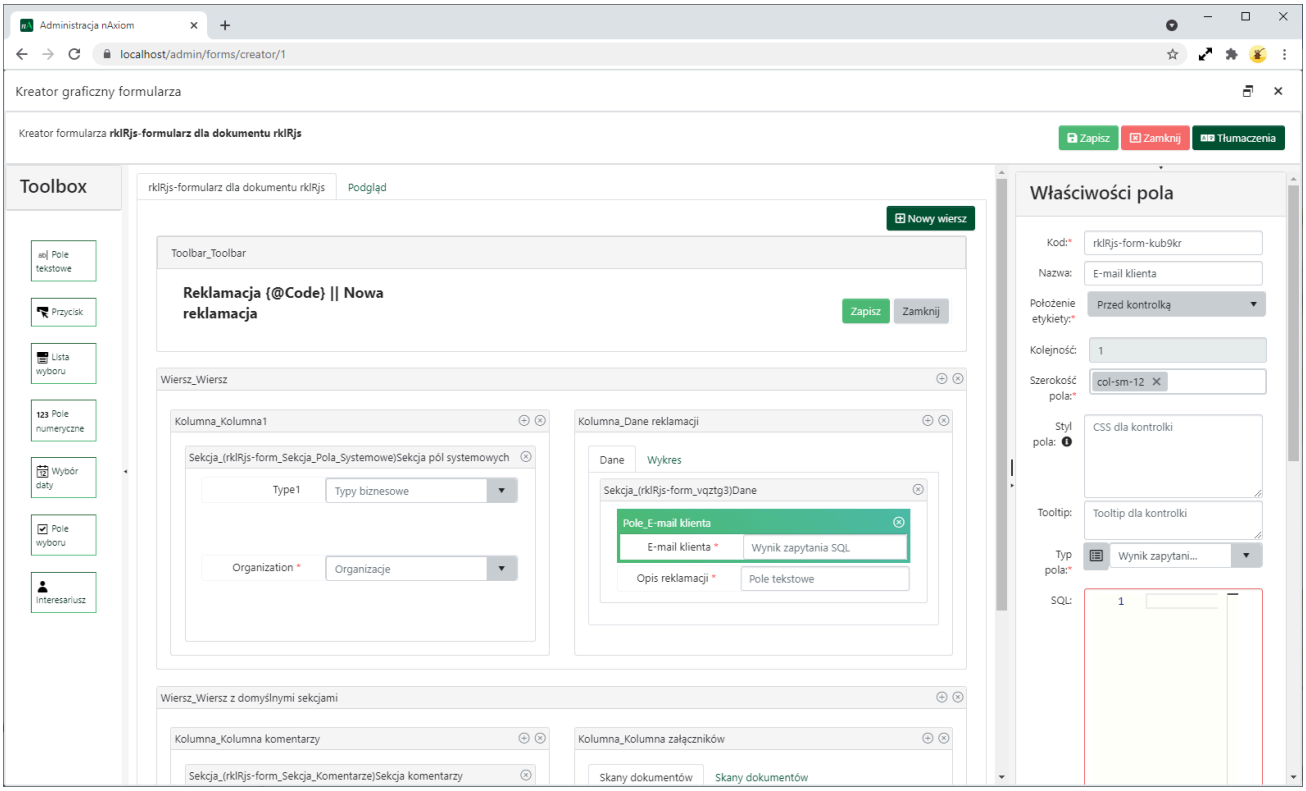
Powiązane tematy:

- Wspólne właściwości sekcji
- Akcja Anonimizuj dane

4.1.6. Kontrolki formularza

> **Podsumowanie:**
> *Kontrolki* to elementy graficznego interfejsu użytkownika, które reprezentują *pola formularza* — elementy logicznej struktury formularza powiązane z kolumnami źródła danych. Kontrolki ułatwiają wprowadzanie i edytowanie danych konkretnego typu. Dla pola formularza określonego typu można wybrać różne kontrolki, na przykład pole typu *int* może być obsługiwane przez *pole numeryczne* lub *listę wyboru*. Niektóre kontrolki pozwalają sterować ukrywaniem lub wyświetlaniem innych kontrolek lub filtrować dane w innych polach. Osobną kategorię stanowią kontroli, które nie są powiązane z polami formularza, na przykład przycisk, blok HTML czy wynik zapytania SQL.

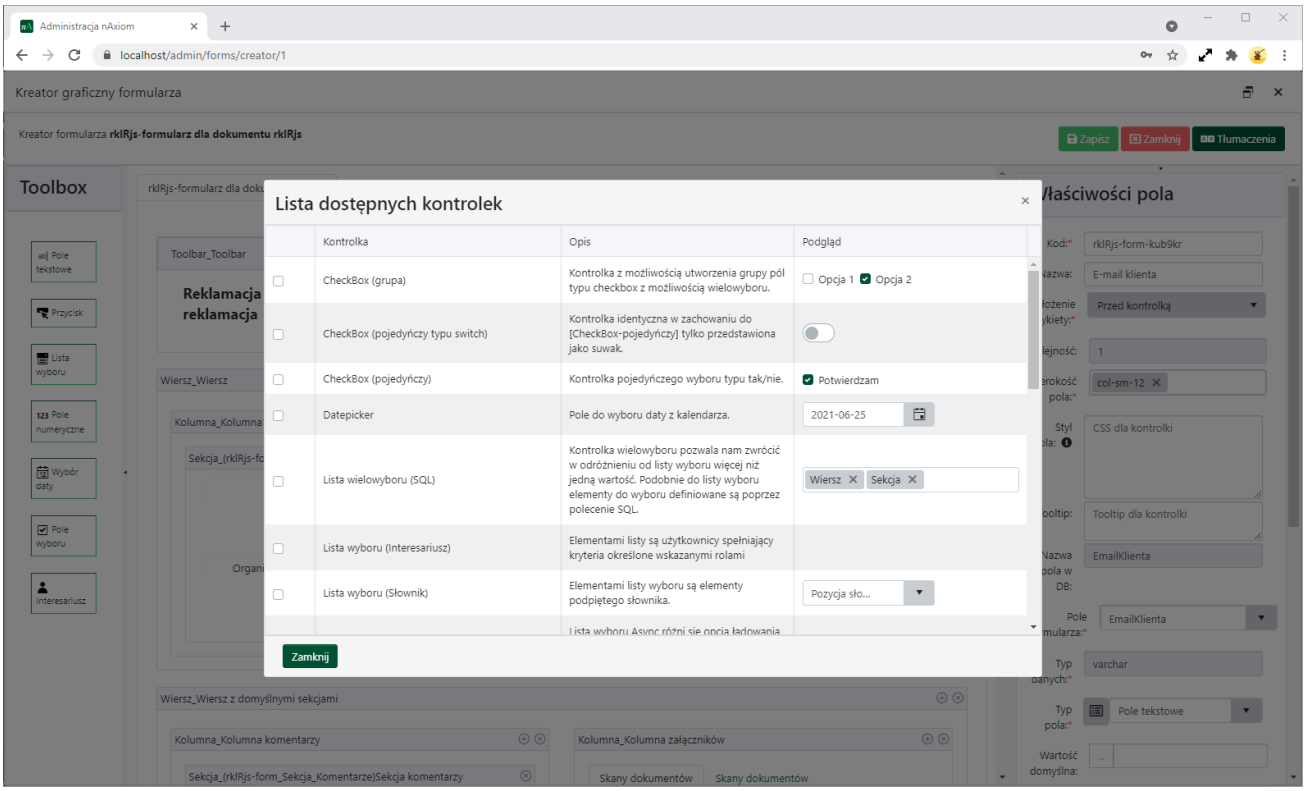
Z lewej strony okna projektu formularza w trybie designera znajduje się panel `Toolbox`. Jest to przybornik z kilkoma podstawowymi typami pól, czyli kontrolkami, dostępnymi w nAxiom. Aby utworzyć na formularzu pole danego typu, należy przeciągnąć odpowiadającą mu ikonę z panelu `Toolbox` do odpowiedniej sekcji formularza. Następnie należy ustawić właściwości pola w panelu `Właściwości pola` z prawej strony. Przyciski można umieszczać zarówno w sekcjach, jak i w wierszu systemowym `Toolbar`.



Projekt formularza w trybie designera, z lewej strony panel `Toolbox`, z prawej — panel `Właściwości pola`

Jak wspomniano, na panelu `Toolbox` znajduje się tylko kilka najczęściej używanych kontrolek. Są to: `Pole tekstowe`, `Przycisk`, `Lista wyboru`, `Pole numeryczne`, `Wybór`

`daty`, `Pole wyboru` i `Interesariusz`. Jednak w nAxiom jest dostępnych 25 rodzajów kontrolek (stan dla wersji nAxiom 1.11.*). Aby użyć na formularzu kontrolki, której nie ma na panelu `Toolbox`, należy przeciągnąć dowolną inną kontrolkę (z wyjątkiem przycisku) na sekcję formularza i na panelu właściwości wybrać odpowiedni typ kontrolki z listy `Typ pola`. Lista `Typ pola` ma z lewej strony przycisk, który wyświetla okienko z krótkim opisem oraz podglądem wszystkich dostępnych typów kontrolek.



Lista dostępnych kontrolek

W kolejnych sekcjach tego podrozdziału opisano poszczególne typy kontrolek formularza i ich właściwości.

4.1.6.1. WŁAŚCIWOŚCI KONTROLEK - PRZEGLĄD

Podsumowanie:
W tym temacie zamieszczono tabelę z wykazem wszystkich kontrolek dostępnych w nAxiom i ich właściwości. Do przewijania tabeli można używać strzałek na klawiaturze lub kółka myszy (przewijanie w poziomie kółko + Shift). Ze względu na rozmiary tabela nie jest publikowana w formacie PDF, tylko w witrynie dokumentacji.

4.1.6.2. TYPOWE WŁAŚCIWOŚCI KONTROLEK

Podsumowanie:
Wszystkie lub prawie wszystkie kontrolki formularza mają właściwości *Kod*, *Nazwa*, *Położenie etykiety* i kilkanaście innych. Te powtarzające się właściwości opisano w tym rozdziale.

Większość kontrolek ma następujące właściwości:

`Id`: identyfikator kontrolki; tylko do odczytu.

`Kod`: generowany automatycznie ciąg znaków jednoznacznie identyfikujący pole na formularzu; używany w odwołaniach do pól formularza.

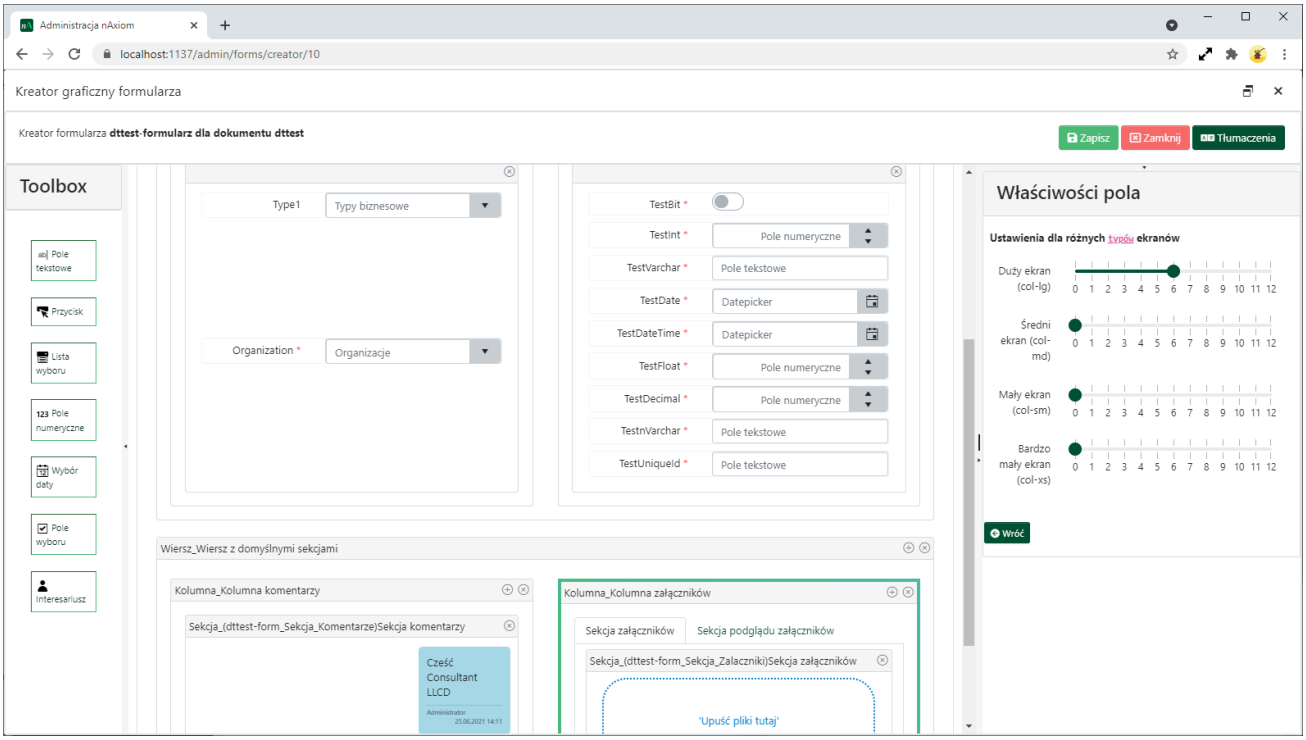
`Nazwa`: tekst wyświetlany obok kontrolki (*etykieta*); w przypadku przycisku, tekst wyświetlany na przycisku.

`Położenie etykiety`: określa położenie tekstu etykiety względem kontrolki; dostępne wartości:

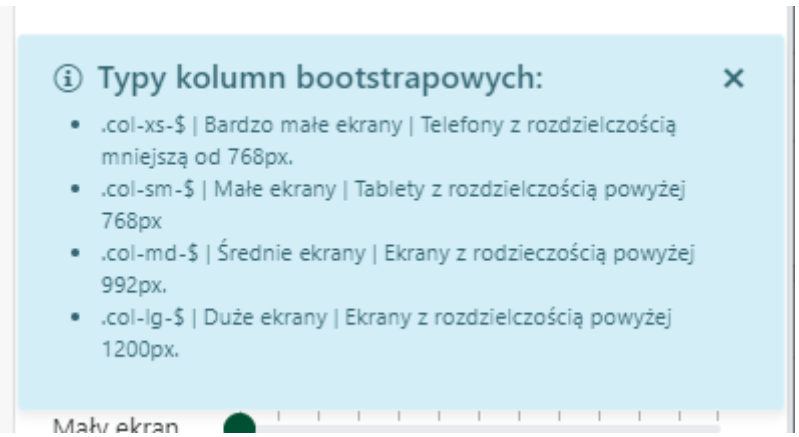
- `Przed kontrolką`: tekst jest wyświetlany z lewej strony kontrolki (wartość domyślna).
- `Nad kontrolką`: tekst jest wyświetlany nad kontrolką i wyrównany do lewej krawędzi.

`Kolejność`: (pole tylko do odczytu) numer kolejny kontrolki w bieżącej sekcji; nadawany automatycznie podczas dodawania kontrolki do formularza.

`Szerokość pola`: po kliknięciu w tym polu jest wyświetlane okno dialogowe, w którym można wybrać szerokość kontrolki w umownych jednostkach dla różnych rodzajów ekranów.



Ustawienia szerokości kontrolki formularza



Pomoc dla ustawień szerokości kontrolki formularza

`Styl pola`: pole tekstowe, w którym można wpisać definicje stylu CSS, które zostaną zastosowane do kontrolki w widoku formularza.

`Tooltip`: dowolny tekst, który będzie wyświetlany po wskazaniu kursorem myszy przycisku (`Info`) znajdującego się bezpośrednio po etykiecie kontrolki. Możliwe jest wpisanie tekstu o długości odpowiadającej limitowi *nvarchar(MAX)* (ok, 2 GB), jednak okienko podpowiedzi ma rozmiar ograniczony dostępnym miejscem na ekranie i nie obsługuje przewijania, dlatego efektywnie można wyświetlić najwyżej kilkaset wyrazów.

`Nazwa pola w DB`: (pole tylko do odczytu) nazwa kolumny w zapytaniu SQL, które jest źródłem danych formularza, w przypadku formularza generowanego poleceniem `Utwórz flow`, nazwa kolumny w tabeli dokumentu biznesowego, która jest powiązana z formularzem.

Aby zdefiniować nową kolumnę w tabeli źródłowej formularza, kliknij ikonę `+` z lewej strony listy rozwijanej. Zostanie wyświetlone okno dialogowe `Nowe pole tabeli SQL`. Wypełnij kolejno pola:

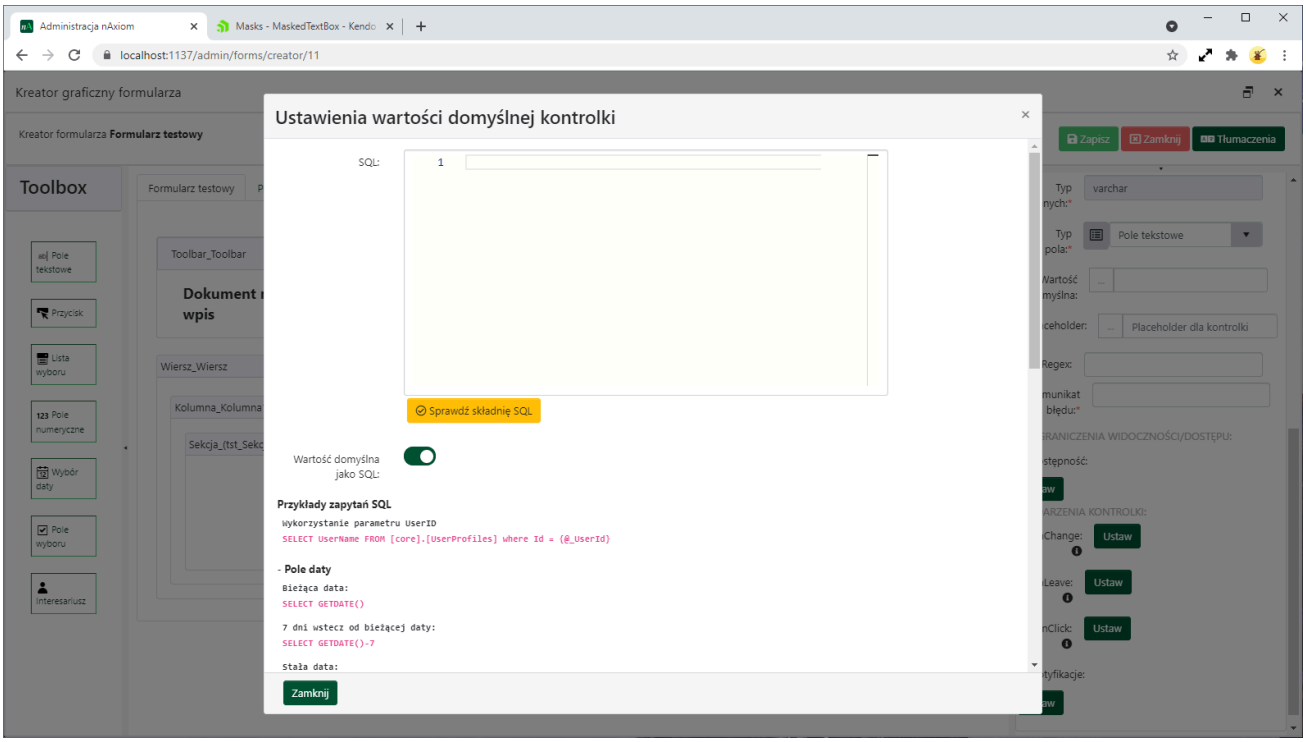
- `DataFieldName`: nazwa pola,
- `Etykieta`: etykieta pola modelu danych (maks. 255 dowolnych znaków); wartość atrybutu jest zapisywana w tabeli *core.SqlBuilderColumnsInfo*; wartość tego atrybutu może być tłumaczona; planuje się systemowe wykorzystanie tego atrybutu w przygotowywanej funkcjonalności formularzy publicznych.
- `Typ danych`: typ danych,
- `Typ danych`: rozmiar pola,
- `Nullable`: zaznacz, jeśli pole ma dopuszczać wartości *NULL*,
- `Dane wrażliwe`: zaznacz, jeśli dane w tym polu mają być traktowane jako dane osobowe,
- `Wyszukiwanie`: zaznacz, jeśli dane w tym polu mają być indeksowane na potrzeby wyszukiwania.

`Pole formularza`: nazwa kolumny w zapytaniu SQL, które jest źródłem danych formularza, w przypadku formularza generowanego poleceniem `Utwórz flow`, nazwa kolumny w tabeli dokumentu biznesowego, która jest powiązana z formularzem.

`Typ danych`: (pole tylko do odczytu) typ danych kolumny źródła danych powiązanej z danym polem formularza.

`Typ pola`: rodzaj kontrolki, która ma reprezentować dane pole w widoku formularza; przycisk z lewej strony tej listy rozwijanej wyświetla okno dialogowe z krótkim opisem dostępnych kontrolek.

`Wartość domyślna`: wartość, która będzie wpisana/ustawiona w danym polu podczas tworzenia nowego dokumentu; po kliknięciu przycisku z wielokropkiem (`...`) z lewej strony pola jest wyświetlane okno dialogowe, w którym znajduje się przełącznik `Wartość domyślna jako SQL`, jego włączenie pozwala wpisać wyrażenie SQL, które będzie zwracało wartość domyślną dla danego pola, ponadto w oknie podano przykłady określania wartości domyślnej dla różnych kontrolek.



Ustawienia wartości domyślnej kontrolki

- `Wartość minimalna/maksymalna`: określa wartość minimalną/maksymalną w danej kontrolce; wpisanie wartości niezgodnej z podanym ograniczeniem powoduje podświetlenie kontrolki na czerwono i wyświetlenie zdefiniowanego komunikatu o błędzie; zależnie od rodzaju kontrolki znaczenie wartości min./maks. jest następujące:

- **pole tekstowe/obszar tekstowy**: minimalna/maksymalna liczba znaków,

- **pole numeryczne**: minimalna/maksymalna wartość liczbowa,
- **pole czasu**, **pole daty**, **datepicker**, **datepicker i pole czasu**: najwcześniejsza i najpóźniejsza wartość daty/godziny.

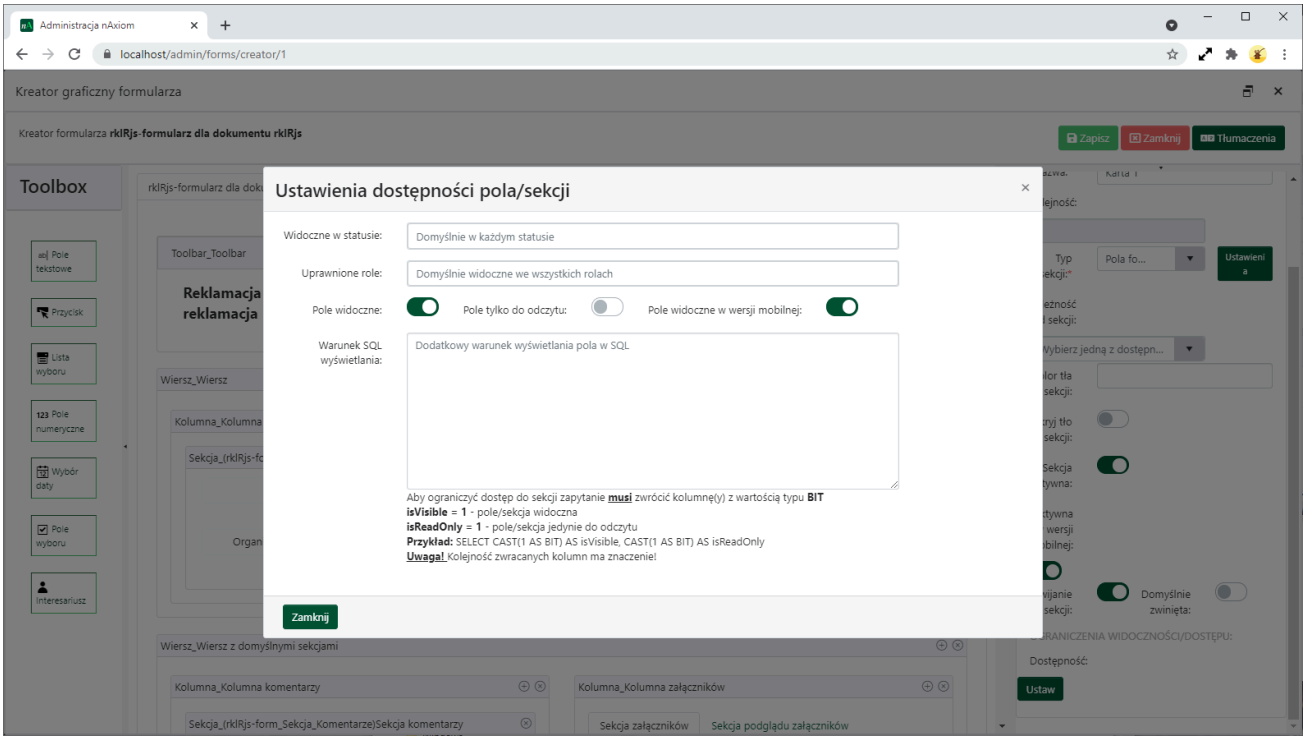
Wartości minimalne i maksymalne mogą być określane jako statyczne (stałe) lub za pomocą SQL (należy kliknąć przycisk z wielokropkiem i włączyć przełącznik `Wartość minimalna/maksymalna jako SQL`).

`Placeholder`: pomocniczy tekst wyświetlany w polu przed wpisaniem w nie wartości; po kliknięciu przycisku z wielokropkiem (`...`) z lewej strony pola jest wyświetlane okno dialogowe, w którym znajduje się przełącznik `Wartość domyślna jako SQL`, jego włączenie pozwala wpisać wyrażenie SQL, które będzie zwracało wartość właściwości `Placeholder` dla danego pola, ponadto w oknie podano przykłady określania tekstu podpowiedzi za pomocą SQL.

`Regex`: wyrażenie regularne, które będzie dopasowywane do wartości wpisywanej w polu, np. w celu sprawdzenia poprawności wpisywanych danych, ograniczenia możliwych znaków itp.; jeśli wartość wpisywana w polu będzie niezgodna z wyrażeniem, obramowanie pola zmieni kolor na czerwony i nie będzie można zapisać dokumentu.

`Komunikat o błędzie`: tekst komunikatu wyświetlany pod kontrolką w przypadku wpisania błędnych danych, np. niezgodnych z wzorcem określonym właściwością `Regex`.

`Dostępność`: kliknięcie przycisku `Ustaw` wyświetla okno dialogowe, w którym można sterować widocznością, dostępnością (możliwością zmiany wartości) i wymagalnością pola formularza przy użyciu opisanych poniżej ustawień.



Ustawienia dostępności pola/sekcji

- `dla dokumentu w statusie`: lista wielowyboru; kliknij i wybierz statusy, w których sekcja/kontrolka ma być widoczna; w pozostałych statusach sekcja/kontrolka będzie niewidoczna; brak wybranych statusów (ustawienie domyślne) powoduje, że sekcja/kontrolka jest wyświetlana we wszystkich statusach.
- `dla ról biznesowych`: lista wielowyboru; kliknij i wybierz role, dla których sekcja/kontrolka ma być widoczna; dla pozostałych ról sekcja/kontrolka będzie niewidoczna; brak wybranych ról (ustawienie domyślne) powoduje, że sekcja/kontrolka jest wyświetlana dla wszystkich ról.
- `Sekcja/Pole widoczne`: wyłączenie przełącznika powoduje ukrycie sekcji/kontrolki na formularzu; w przypadku sekcji ten przełącznik jest sprzężony z przełącznikiem `Sekcja aktywna`.
- `Pole wymagane`: (przełącznik niedostępny dla sekcji formularza) pozwala w niektórych przypadkach przesłonić ustawienie `Zezwól Null` dla kolumny tabeli powiązanej z polem formularza. Domyślny stan przełącznika odpowiada ustawieniu dla kolumny (jeśli atrybut `Zezwól Null` ma wartość *false*, to przełącznik `Pole wymagane` jest załączony i odwrotnie). W ogólności służy do zmiany w „obowiązkowe" pola, dla którego powiązana kolumna tabeli zezwala na wartości NULL. W sytuacji odwrotnej, to jest kiedy kolumna nie zezwala na wartości NULL, wyłączenie tego przełącznika będzie powodowało zapisywanie wartości `''` (pusty string) w polach typu `varchar` i `nvarchar` oraz `0` w polach typu liczbowego i będzie ignorowane w polach innych typów.
- `Sekcja/Pole tylko do odczytu`: domyślnie wyłączone, załączenie powoduje zablokowanie możliwości edycji wartości danej kontrolki, w przypadku sekcji blokuje możliwość edycji wartości we wszystkich polach sekcji.
- `Sekcja/Pole widoczne w wersji mobilnej`: domyślnie włączone, wyłączenie powoduje, że dana sekcja/pole nie będzie widoczne w wersji mobilnej aplikacji.
- `Sekcja/Pole dostępne w przeglądarce nawet gdy niewidoczne`: włączenie umożliwia wykonywanie operacji na wartościach pól w sekcji niewidocznej na formularzu.
- `Ustawienia zaawansowane`: okno edytora SQL, w którym można wpisać warunek SQL określający wartości przełączników `Sekcja/Pole widoczne`, `Pole wymagane` i `Sekcja/Pole tylko do odczytu`. Warunek ma postać zapytania, które musi zwracać trzy kolumny o podanych nazwach i w podanej kolejności:

```sql
SELECT
  CAST(warunek1 AS BIT) AS isVisible,
  CAST(warunek2 AS BIT) AS isRequired,
  CAST(warunek3 AS BIT) AS isReadOnly
```

gdzie *warunek**n*** to wyrażenie, które zwróci wartość `0` lub `1`. Na przykład poniższe zapytanie będzie blokować możliwość edycji danego pola, jeśli do daty w polu `TerminZakonczenia` pozostanie mniej niż 5 dni:

```
SELECT
  CAST(1 AS BIT) AS isVisible,
  CAST(1 AS BIT) AS isRequired,
  CAST(CASE
         WHEN DATEDIFF(day, GETDATE(), {@TerminZakonczenia}) < 5
         THEN 1
         ELSE 0
         END
  AS BIT) AS isReadOnly
```

UWAGI

- Warunki dostępności są obliczane już po załadowaniu wszystkich danych formularza i można w nim używać odwołań do pól formularza, pól technicznych i pól dodawanych w źródle danych dla formularza; obsługiwane są także wartości pól przekazywane w akcji otwierania formularza.
- Wyłączenie widoczności kontrolki dla nowego rekordu wyłącza zapis danych z tej kontrolki w tabeli, nawet jeśli dla kontrolki zdefiniowano wartość domyślną. Może to powodować błędy w przypadku, gdy kolumna powiązana z kontrolką nie zezwala na wartości `NULL`. Aby móc zapisywać w bazie wartość domyślną z ukrytej kontrolki, należy do jej ukrycia użyć funkcji `Modyfikacja zależności kontrolki`.
- Wyłączenie widoczności przełącznikiem `Sekcja/Pole widoczne` ma pierwszeństwo przed ustawieniami pól `dla dokumentu w statusie` i `dla ról biznesowych`.
- Wprowadzenie wyrażenia SQL w sekcji `Ustawienia zaawansowane` nadpisuje ustawienia wyświetlania i wprowadzania danych.

`OnChange`: kliknięcie przycisku `Ustaw` pozwala przypisać jedną lub kilka akcji JavaScript, która zostaną wykonane po wystąpieniu zdarzenia *OnChange*; to zdarzenie jest wyzwalane, gdy użytkownik zmienia wartości w kontrolce; patrz opis akcji Javascript .

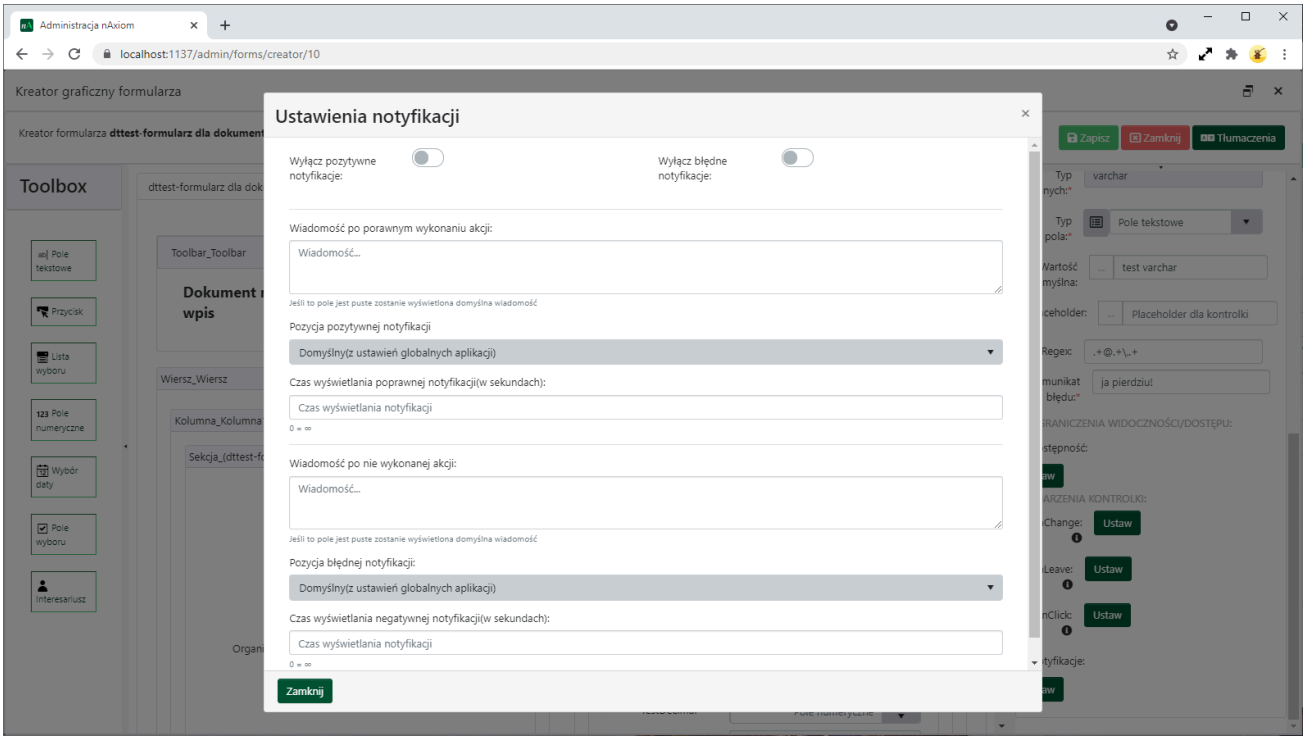
UWAGA
Zdarzenie *OnChange* nie jest wyzwalane przez programową zmianę wartości w kontrolce, np. ustawienie wartości umyślnej lub zmiana kontekstu formularza akcją JavaScript.

`OnLeave`: kliknięcie przycisku `Ustaw` pozwala przypisać jedną lub kilka akcji JavaScript, która zostaną wykonane po wystąpieniu zdarzenia *OnLeave*; to zdarzenie zachodzi z chwilą przejścia z danego pola do innego (utrata *focusu*); patrz opis akcji Javascript .

`OnClick`: kliknięcie przycisku `Ustaw` pozwala przypisać jedną lub kilka akcji JavaScript, która zostaną wykonane po wystąpieniu zdarzenia *OnClick*; to zdarzenie zachodzi z chwilą kliknięcia kontrolki na formularzu; patrz opis akcji Javascript .

WAŻNE
Wyzwalanie zdarzeń *OnChange*, *OnClick* i *OnLeave* może się różnić, w zależności od rodzaju kontrolki. Przed oprogramowaniem tych zdarzeń dla konkretnej kontrolki można określić moment wyzwolenia danego zdarzenie, korzystając z funkcjonalności logów serwisowych .

`Notyfikacje`: kliknięcie przycisku `Ustaw` wyświetla okno dialogowe `Ustawienia notyfikacji`, w którym można określać opisane poniżej ustawienia komunikatów wyświetlanych w przypadku powodzenia i niepowodzenia zapisu danych w polu formularza.



Ustawienia notyfikacji

`Wyłącz pozytywne notyfikacje`: wyłącza wyświetlanie komunikatów po poprawnym wykonaniu akcji przypisanej do przycisku.

`Wyłącz błędne notyfikacje`: wyłącza wyświetlanie komunikatów w przypadku niewykonania akcji przypisanej do przycisku.

`Wiadomość po poprawnym wykonaniu akcji`: tekst komunikatu wyświetlanego po poprawnym wykonaniu akcji.

`Pozycja pozytywnej notyfikacji`: miejsce wyświetlania na ekranie okienka z komunikatem o powodzeniu; dostępne wartości:

- `Prawa góra`: (wartość domyślna) prawy górny róg ekranu.
- `Domyślny (z ustawień globalnych aplikacji`:
- `Środkowa góra`: na środku u góry ekranu.
- `Lewa góra`: lewy górny róg ekranu.
- `Prawy dół`: prawy dolny róg ekranu.
- `Środkowy dół`: na środku u dołu ekranu.
- `Lewy dół`: lewy dolny róg ekranu.

`Czas wyświetlania pozytywnej notyfikacji (w sekundach)`: czas w sekundach, po którym komunikat zostanie wygaszony; wartość 0 powoduje, że wygaszenie komunikatu wymaga kliknięcia go przez użytkownika.

`Wiadomość po niewykonanej akcji`: tekst komunikatu wyświetlanego, kiedy akcja nie zostanie wykonana.

`Pozycja błędnej notyfikacji`: miejsce wyświetlania na ekranie okienka z komunikatem o niepowodzeniu; dostępne wartości:

- `Prawa góra`: (wartość domyślna) prawy górny róg ekranu.
- `Domyślny (z ustawień globalnych aplikacji`:
- `Środkowa góra`: na środku u góry ekranu.
- `Lewa góra`: lewy górny róg ekranu.
- `Prawy dół`: prawy dolny róg ekranu.
- `Środkowy dół`: na środku u dołu ekranu.
- `Lewy dół`: lewy dolny róg ekranu.

`Czas wyświetlania negatywnej notyfikacji (w sekundach)`: czas w sekundach, po którym komunikat zostanie wygaszony; wartość 0 powoduje, że wygaszenie komunikatu wymaga kliknięcia go przez użytkownika.

Powiązane tematy:

- Ustawienia dostępności pola/sekcji

4.1.6.3. MODYFIKACJA ZALEŻNOŚCI KONTROLKI — OKNO DIALOGOWE

Podsumowanie:
Zależności kontrolek służą do konfigurowania relacji między kontrolkami na formularzu i używania ich do sterowania widocznością kontrolki podrzędnej odpowiednio do wartości kontrolki nadrzędnej oraz do filtrowania pozycji na podrzędnej liście wyboru wartością wybraną na nadrzędnej liście wyboru.

Zależności są dostępne dla wybranych rodzajów kontrolek.

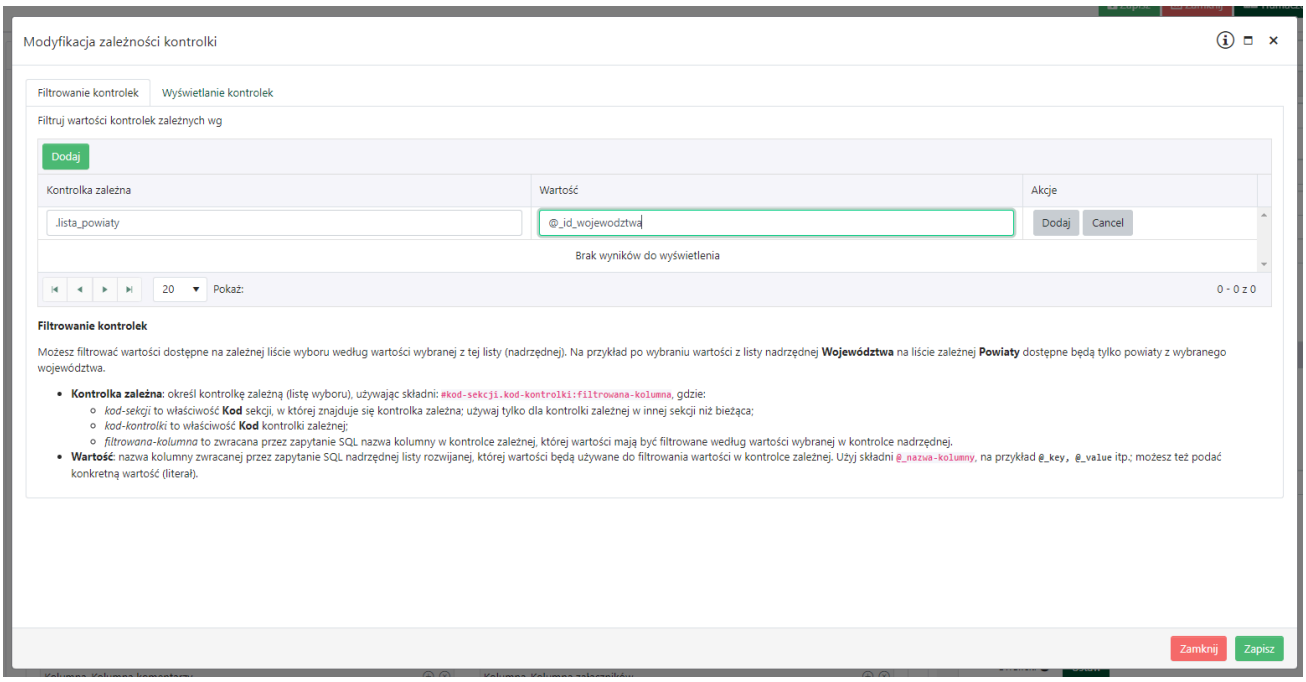
`Zależności` to właściwość następujących kontrolek: *Checkbox* (grupa i pojedynczy), *Lista wielowyboru*, *Lista wyboru* (słownik, SQL - async, SQL), *Pole numeryczne*, *Przyciski Radio* oraz *Typy biznesowe*.

Po skonfigurowaniu tej właściwości dla kontrolki staje się ona kontrolką nadrzędną dla wskazanych kontrolek zależnych, to znaczy wybór wartości w tej kontrolce wpływa na dostępne wartości lub widoczność kontrolek zależnych.

Zależność polegającą na filtrowaniu można określić tylko między kontrolkami typu lista rozwijana. Zależność związaną ze zmianą atrybutu widoczności może być definiowana dla kontrolek zależnych dowolnego typu.

W celu skonfigurowania właściwości `Zależność` kliknij przycisk `Modyfikuj zależności kontrolki`. Zostanie wyświetlone okno dialogowe `Modyfikacja zależności kontrolki`. Zależności konfiguruje się na dwóch zakładkach: `Filtrowanie kontrolek` i `Wyświetlanie kontrolek`. W obu przypadkach dodanie nowej zależności wymaga kliknięcia przycisku `Dodaj`. Następnie, odpowiednio do wybranej karty, należy podać wartości opisane poniżej.

FILTROWANIE KONTROLEK



Karta Filtrowanie kontrolek

Filtrowanie kontrolek polega na filtrowaniu wartości dostępnych na zależnej liście wyboru według wartości wybranej z listy nadrzędnej (bieżącej). Na przykład po wybraniu wartości z listy nadrzędnej `Województwa` na liście zależnej `Powiaty` dostępne będą tylko powiaty z wybranego województwa.

W celu skonfigurowania zależności należy określić następujące wartości:

- `Kontrolka zależna`: określ kontrolkę zależną (listę wyboru), używając składni: ***#kod-sekcji.kod-kontrolki:filtrowana-kolumna***, gdzie:
  - *kod-sekcji* to właściwość `Kod` sekcji, w której znajduje się kontrolka zależna; używaj tylko dla kontrolki zależnej w innej sekcji niż bieżąca;
  - *kod-kontrolki* to właściwość `Kod` kontrolki zależnej

    > **UWAGA**
    > Pamiętaj, aby przed kodem kontrolki użyć kropki, także wtedy, kiedy jest to kontrolka z sekcji bieżącej;

  - *filtrowana-kolumna* to zwracana przez zapytanie SQL nazwa kolumny w kontrolce zależnej, której wartości mają być filtrowane według wartości wybranej w kontrolce nadrzędnej.
- `Wartość`: nazwa kolumny zwracanej przez zapytanie SQL nadrzędnej listy rozwijanej, której wartości będą używane do filtrowania wartości w kontrolce zależnej. Użyj składni ***@_nazwa-kolumny***, na przykład *@_key*, *@_value* itp.; możesz też podać konkretną wartość (literał).

WSKAZÓWKA

W przypadku, gdy do filtrowania kontrolki zależnej jest używana wartość typu GUID, np. *{@UserId}*, należy zadbać, aby zarówno w kontrolce nadrzędnej jak i kontrolce zależnej w zapytaniach źródłowych przekonwertować wartość identyfikatora na tekst o takiej samej wielkości liter, np.:

```
UPPER(CONVERT(varchar(36), UserId))
-- alternatywnie:
UPPER(CAST(UserId AS varchar(36))
```

Na koniec kliknij przycisk `Dodaj` w kolumnie `Akcje`, aby dodać zależność. Dla jednej kontrolki nadrzędnej można skonfigurować kilka kontrolek zależnych.

Przykład:

Aby wybór wartości na jednej liście rozwijanej ograniczał listę dostępnych wartości na drugiej liście rozwijanej, należy dla tej pierwszej listy (nadrzędnej) skonfigurować zależność jak w poniższej tabelce.

Kontrolka zależna	Wartość
.TerytPL-sa0o7f:WojFltr	@_key

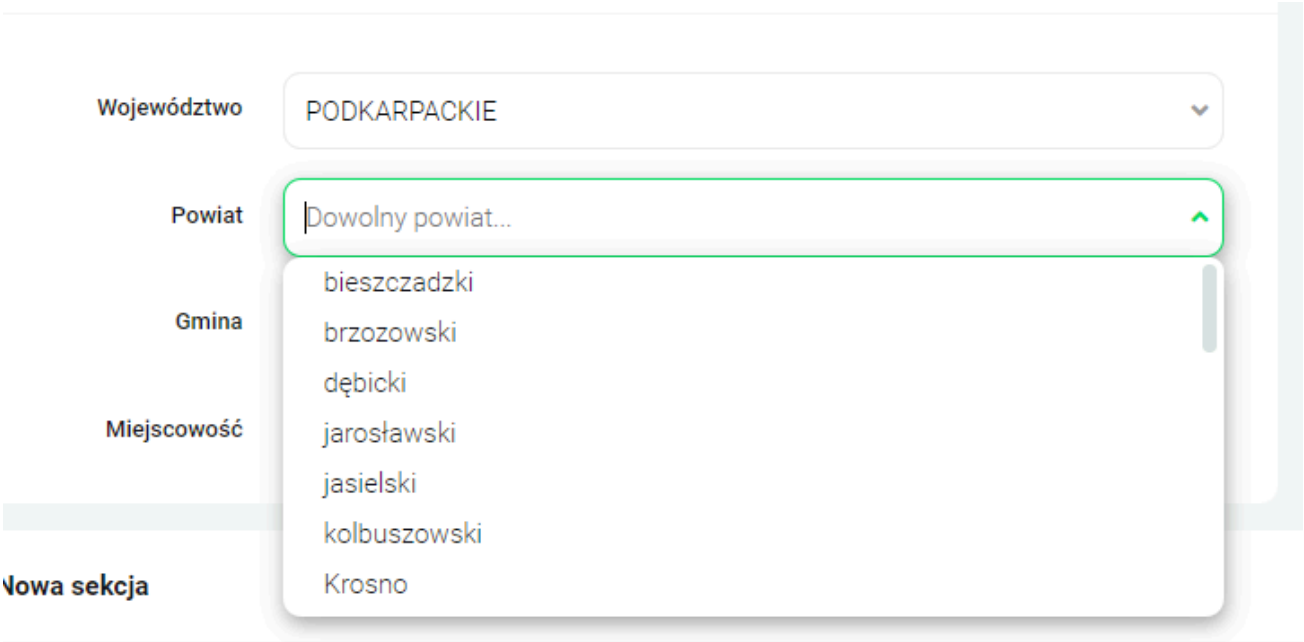
Kontrolka o kodzie *TerytPL-sa0o7f* to lista rozwijana z nazwami powiatów. Zapytanie SQL dla tej listy zwraca kolumnę *WojFltr* z identyfikatorami województw:

```
SELECT WOJ AS [WojFltr]
     ,CAST(CONCAT(WOJ, POW) AS nvarchar) AS [key]
     ,GMI
     ,RODZ
     ,NAZWA AS [value]
     ,NAZWA_DOD
FROM [dbo].[PodzialAdministracyjny]
WHERE NAZWA_DOD
IN ('powiat',
                    'miasto na prawach powiatu',
                    'miasto stołeczne, na prawach powiatu')
ORDER BY value ASC
```

Filtrowanie polega na wybraniu z listy wszystkich powiatów, dla których wartość w kolumnie *WojFltr* jest równa wartości ustawionej w kolumnie *key* na liście nadrzędnej. Ta kolumna to identyfikator województwa. Zapytanie SQL dla listy nadrzędnej jest następujące:

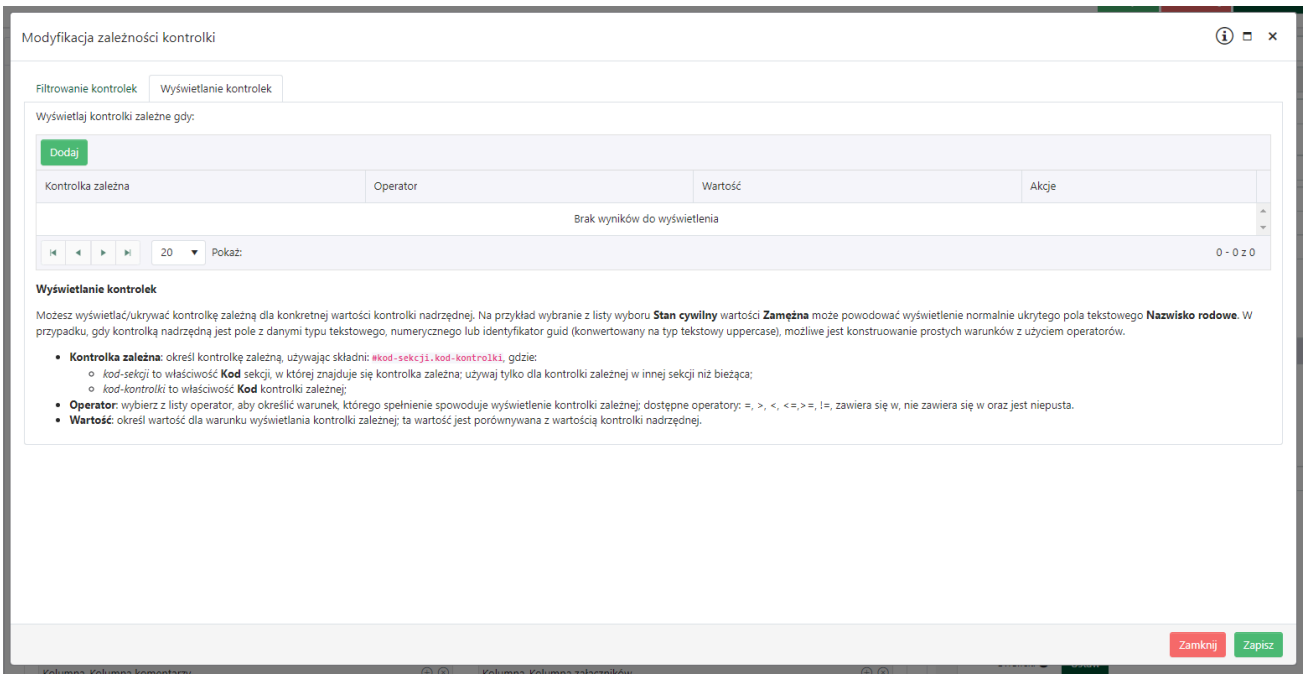
```
SELECT WOJ AS [key]
      ,NAZWA AS [value]
      ,NAZWA_DOD
FROM [dbo].[PodzialAdministracyjny]
WHERE NAZWA_DOD = 'województwo'
ORDER BY value ASC
```

Wybranie województwa na liście nadrzędnej powoduje odfiltrowanie z listy powiatów wszystkich pozycji, dla których identyfikator województwa jest inny niż na liście nadrzędnej.



Filtrowanie wartości na liście rozwijanej

WYŚWIETLANIE KONTROLEK



Karta Wyświetlanie kontrolek

Ten rodzaj zależności służy do sterowania widocznością kontrolek zależnych odpowiednio do wartości ustawionej w kontrolce nadrzędnej. W tym przypadku kontrolkami nadrzędnymi mogą być wszystkie rodzaje kontrolek, które mają właściwość `Zależność`, a kontrolkami zależnymi — dowolne kontrolki. Zależność działa w ten sposób, że jeśli podany w zależności warunek jest spełniony, kontrolka zależna jest wyświetlana, w przeciwnym razie jest ukrywana.

Na przykład wybranie z listy wyboru `Stan cywilny` wartości `Zamężna` może powodować wyświetlenie normalnie ukrytego pola tekstowego `Nazwisko rodowe`. W przypadku, gdy kontrolką nadrzędną jest pole z danymi typu tekstowego, numerycznego lub identyfikator *guid* (konwertowany na typ tekstowy *uppercase*), możliwe jest konstruowanie prostych warunków z użyciem operatorów. W celu skonfigurowania zależności należy określić następujące wartości:

- `Kontrolka zależna`: określ kontrolkę zależną, używając składni: ***#kod-sekcji.kod-kontrolki***, gdzie:
    - *kod-sekcji* to właściwość `Kod` sekcji, w której znajduje się kontrolka zależna; używaj tylko dla kontrolki zależnej w innej sekcji niż bieżąca;
    - *kod-kontrolki* to właściwość `Kod` kontrolki zależnej; pamiętaj, aby przed kodem kontrolki użyć kropki, także wtedy, kiedy jest to kontrolka z sekcji bieżącej;

> **WSKAZÓWKA**
> Jeśli w polu `Kontrolka zależna` użyjesz składni ***#kod-sekcji***, zdefiniowana zależność będzie sterować wyświetlaniem sekcji o podanym kodzie.

- `Operator`: wybierz z listy operator, aby określić warunek, którego spełnienie spowoduje wyświetlenie kontrolki zależnej; dostępne operatory: =, >, <, <=, >=, *!=*, *zawiera się w*, *nie zawiera się w* oraz *jest niepusta*; dla operatorów *zawiera się w* i *nie zawiera się w* można określić listę wartości w formacie *[wartość1,wartość2...]* lub zakres wartości (domknięty) w formacie *<wartość_początkowa;wartość_końcowa>*
- `Wartość`: określ wartość dla warunku wyświetlania kontrolki zależnej; ta wartość jest porównywana z wartością kontrolki nadrzędnej.

W tabeli poniżej podano zasady określania warunków wyświetlania kontrolki zależnej.

Typ wartości	Operator	Wartość	Uwagi
Liczba	=, >, <, <=, >=, !=	123	porównuje wartość kontrolki nadrzędnej z podaną liczbą
	zawiera się/nie zawiera się	[1,7,13]	sprawdza, czy wartość kontrolki nadrzędnej jest równa dowolnej liczbie z podanej listy
		<5;7>	sprawdza, czy wartość kontrolki nadrzędnej należy do podanego przedziału domkniętego (równoważny warunek *5 <= n <= 7*, gdzie *n* to wartość kontrolki nadrzędnej)
Tekst	=, !=	abc	porównuje wartość kontrolki nadrzędnej z podanym ciągiem znaków
	zawiera się/nie zawiera się	["abc","cda","bbc"]	sprawdza, czy wartość kontrolki nadrzędnej jest równa dowolnemu ciągowi znaków z podanej listy

Przykład:

Aby w zależności od wartości wybranej na liście `Stan cywilny` sterować widocznością pola `Nazwisko rodowe`, należy skonfigurować następującą zależność:

Kontrolka zależna	Operator	Wartość
.birthNameTxtBx	=	Zamężna

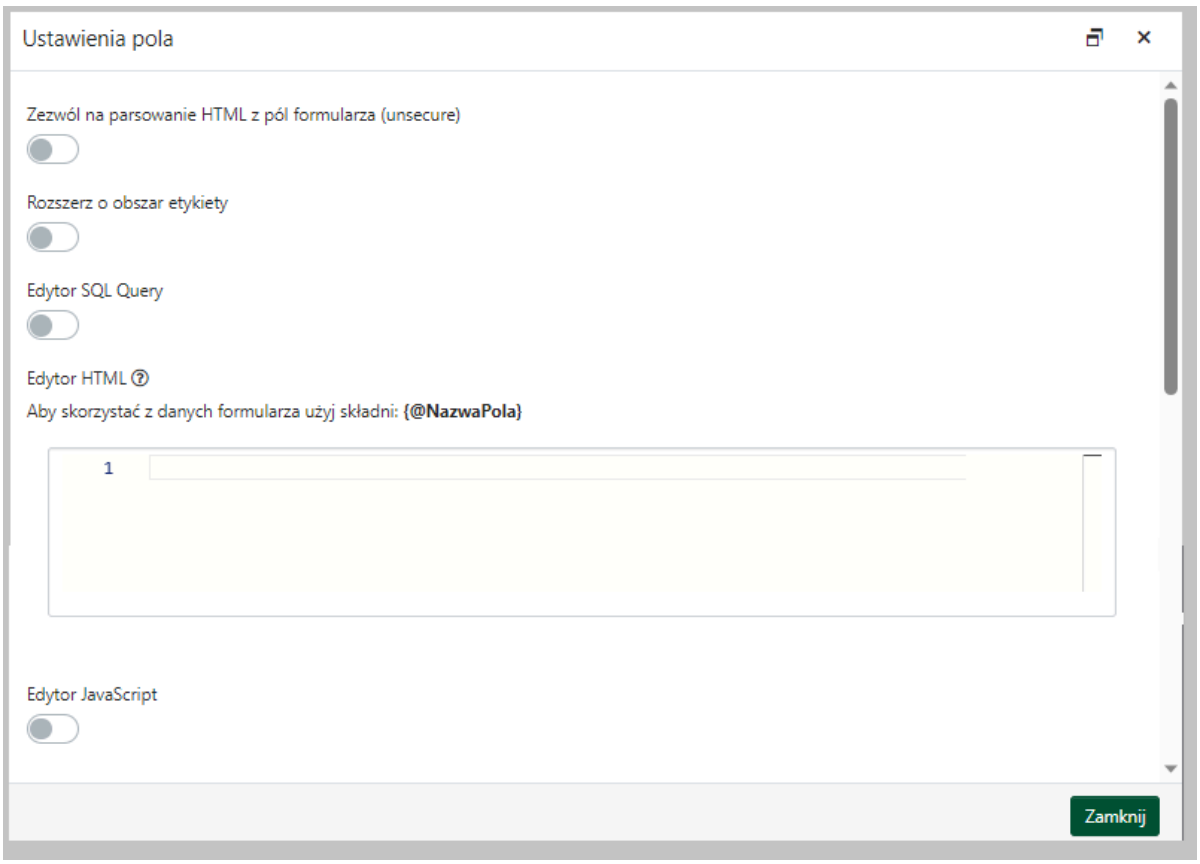
birthNameTxtBx to kod kontrolki zależnej, czyli pola tekstowego `Nazwisko rodowe`, a *Zamężna* to wartość na liście `Stan cywilny`, po wybraniu której kontrolka zależna będzie wyświetlana. W przypadku innych wartości kontrolka będzie ukryta.

4.1.6.4. BLOK HTML

> **Podsumowanie:**
> Kontrolka tylko do odczytu. Renderuje kod HTML skonfigurowany we właściwościach kontrolki.

Aby skonfigurować kod HTML do wyświetlenia w kontrolce, należy kliknąć przycisk `Konfiguracja pola`. Wyświetla on okno dialogowe z następującymi właściwościami:

- `Zezwól na parsowanie HTML z pól formularza (unsecure)`: umożliwia renderowanie w tym polu znaczników HTML zwróconych przez SQL (np. z innych pól formularza).
- `Rozszerz o obszar etykiety`: powoduje, że etykieta pola (właściwość `Nazwa`) nie jest wyświetlana.
- `Edytor SQL Query`: powoduje wyświetlenie okna edytora SQL w celu wpisania zapytania SQL; zapytanie musi zwracać wartości z innych pól formularza, które mają zostać wyświetlone w bloku HTML; należy użyć składni *{@NazwaPola}*.
- `Edytor HTML`: pole do wpisania kodu HTML; w kodzie HTML można używać wyników zwracanych przez zapytanie SQL oraz kodu JavaScript; zasady są takie same, jak dla sekcji HTML. - `Edytor JavaScript`: wyświetla pole w celu wpisania instrukcji JS, które będą wykonywane podczas renderowania zawartości pola.



Blok HTML - ustawienia

Pozostałe właściwości tej kontrolki opisano w rozdziałach z sekcji *Powiązane tematy*.

Powiązane tematy:

- Typowe właściwości kontrolek
- Właściwości kontrolek - Dostępność
- Sekcje formularza - HTML

4.1.6.5. POLE TEKSTOWE

Podsumowanie:
Pole tekstowe to kontrolka w formie prostokąta, w którą można wpisywać dane tekstowe w jednym wierszu. Zwykle pola tekstowego używa się do wprowadzania danych do kolumn typu *varchar* i *nvarchar*. W pole tekstowe można wpisać dowolną liczbę znaków, jeśli jednak będzie ona większa niż rozmiar zadeklarowany dla kolumny w źródle danych (np. *nvarchar(100)*), podczas próby zapisu rekordu zostanie wyświetlony komunikat o błędzie i rekord nie zostanie zapisany.

Właściwości tej kontrolki opisano w rozdziałach z sekcji *Powiązane tematy*.

Powiązane tematy:

- Typowe właściwości kontrolek
- Właściwości kontrolek - Dostępność
- Obszar tekstowy (TextArea)

4.1.6.6. CHECKBOX (GRUPA)

Podsumowanie:
Kontrolka CheckBox (grupa) to grupa kilku pól wyboru definiowanych przez podanie ich wartości (ciąg znaków opisujący pole wyboru). Wartości zaznaczonych pól wyboru są zapisywane w kolumnie źródła danych. Zwykle grupy pól wyboru używa się do wprowadzania danych do kolumn typu *varchar* i *nvarchar*.

Właściwość specyficzna dla grupy pól wyboru to:

- `Wartości`: ciągi znaków przypisane poszczególnym polom wyboru/przyciskom radiowym w grupie; po wpisaniu każdej wartości naciśnij klawisz `Enter` lub kliknij symbol `+` obok wpisywanej wartości wyświetlony na rozwiniętym poniżej pola panelu (tylko kontrolka *checkbox*).

Powiązane tematy:

- Typowe właściwości kontrolek
- Właściwości kontrolek - Zależności
- Właściwości kontrolek - Dostępność

4.1.6.7. CHECKBOX (POJEDYNCZY TYPU SWITCH)

Podsumowanie:
Kontrolka CheckBox (pojedynczy typu switch) to pole wyboru typu przełącznik. Pole wyboru reprezentuje na formularzu kolumnę źródła danych typu *bit* (wartości 0 lub 1), jednak w widoku formularza bieżący stan kontrolki jest opisywany wartościami logicznymi, to jest *true* lub *false*.

Właściwość specyficzna dla tej kontrolki to:

- `Domyślnie zaznaczone`: po włączeniu powoduje, że pole wyboru jest domyślnie zaznaczone (przełącznik jest włączony); ta właściwość jest połączona sumą logiczną z właściwością `Wartość domyślna`.

Powiązane tematy:

- Typowe właściwości kontrolek
- Właściwości kontrolek - Zależności
- Właściwości kontrolek - Dostępność

4.1.6.8. CHECKBOX (POJEDYNCZY)

Podsumowanie:
Kontrolka CheckBox (pojedynczy) to pole wyboru. Pole wyboru reprezentuje na formularzu kolumnę źródła danych typu *bit* (wartości 0 lub 1), jednak w kontekście formularza bieżący stan kontrolki jest opisywany wartościami logicznymi, to jest *true* lub *false*.

Właściwości tej kontrolki opisano w rozdziałach z sekcji *Powiązane tematy*.

Powiązane tematy:

- Typowe właściwości kontrolek
- Właściwości kontrolek - Zależności
- Właściwości kontrolek - Dostępność

4.1.6.9. DATEPICKER

Podsumowanie:
Datepicker to kontrolka z kalendarzem, która służy do wprowadzania i wyświetlania typów danych *date* i *datetime*.

Oprócz właściwości typowych dla większości kontrolek, datepicker wymaga ustawienia właściwości:

- `Format`: pozwala określić format wyświetlania daty i godziny; podstawowe zasady opisano w poniższej tabeli:
  Formaty danych typu data/godzina

Parametr formatu	Alternatywna składnia	Opis formatu	Przykład
`d`	`d.M.y`	Data krótka	7.6.2021
`D`	`EEEE, d MMMM y`	Data długa	poniedziałek, 7 czerwca 2021
`F`	`EEEE, d MMMM y HH:mm`	Pełna data i godzina	poniedziałek, 7 czerwca 2021 16:34
`g`	`d.M.y H:mm`	Krótka data i godzina	7.6.2021 16:34
`G`	`d.M.y HH:mm:ss`	Krótka data i pełna godzina	7.6.2021 16:34:18
`M`	`d MMMM`	Dzień i nazwa miesiąca	7 czerwca
`m`	`d MMM`	Dzień i skrót miesiąca	7 cze
`y`	`MMM y`	Skrót miesiąca i rok	cze 2021
`t`	`HH:mm`	Krótka godzina	16:34
`T`	`HH:mm:ss`	Pełna godzina	16:34:18
`s`	`"yyyy-MM-dd HH:mm:ss`	Uniwersalny, sortowalny format daty i godziny	2021-06-07 16:34
`u`	`"yyyy-MM-dd HH:mm:ss Z`	Uniwersalny, sortowalny format daty i godziny UTC	2021-06-07 16:34 +0100

- `Domyślnie zaznaczone`: po włączeniu powoduje, że pole wyboru jest domyślnie zaznaczone (przełącznik jest włączony); ta właściwość jest połączona sumą logiczną z właściwością `Wartość domyślna`.

Powiązane tematy:

- Typowe właściwości kontrolek
- Właściwości kontrolek - Zależności
- Właściwości kontrolek - Dostępność

Podsumowanie:
Kontrolka z edytorem tekstu typu Rich Text Editor. Umożliwia wprowadzanie i formatowanie treści tekstowych, tabel, wstawianie obrazów i linków do plików. Dane są zapisywane w powiązanym polu tabeli modelu danych w formacie HTML.

Konfiguracja kontrolki Edytor tekstu, oprócz określenia standardowych właściwości, sprowadza się do określenia, które funkcje edytora mają być wyświetlane na pasku narzędzi kontrolki. Po kliknięciu przycisku `Konfiguracja pola` jest wyświetlane okno dialogowe z trzema kartami, na których można włączyć lub wyłączyć wyświetlanie następujących funkcji edytora:

- `Wyrównanie i formatowanie tekstu`:
  - `Wyrównaj do lewej`
  - `Wyrównaj do środka`
  - `Wyrównaj do prawej`
  - `Pogrubienie`
  - `Kursywa`
  - `Wyjustuj`
  - `Podkreślenie`
  - `Przekreślenie`
  - `Indeks górny`
  - `Indeks dolny`
  - `Zwiększ wcięcie`
  - `Zmniejsz wcięcie`
  - `Zmień format`
  - `Zmień czcionkę`
  - `Zmień rozmiar czcionki`
  - `Kolor czcionki`
  - `Kolor tła`
  - `Blok cytatu`
  - `Wyczyść formatowanie`
- `Tabele i listy`:
  - `Dodaj wiersz przed`
  - `Dodaj wiersz po`

    - `Dodaj kolumnę przed`
    - `Dodaj kolumnę po`
    - `Usuń wiersz`
    - `Usuń kolumnę`
    - `Usuń tabelę`
    - `Wstaw tabelę`
    - `Scal komórki`
    - `Podziel komórkę`
    - `Wstaw listę numerowaną`
    - `Wstaw listę wypunktowaną`

- `Akcje i wstawianie treści`:
    - `Wstaw link`
    - `Usuń link`
    - `Wstaw plik`
    - `Wstaw obraz`
    - `Pokaż źródło`
    - `Cofnij`
    - `Ponów`
    - `Zaznacz wszystko`
    - `Drukuj`

- `Akcje i wstawianie treści`:
    - `Wstaw link`
    - `Usuń link`
    - `Wstaw plik`
    - `Wstaw obraz`
    - `Pokaż źródło`
    - `Cofnij`
    - `Ponów`
    - `Zaznacz wszystko`

Powiązane tematy:

- Typowe właściwości kontrolek
- Właściwości kontrolek - Dostępność

4.1.6.11. LISTA WIELOWYBORU (SQL)

Podsumowanie:
Lista wielowyboru pozwala wpisać w polu formularza kilka wartości ze zdefiniowanego zbioru. Zbiór wartości definiuje się zapytaniem SQL jako parę kolumn *key* i *value*. W polu tabeli powiązanym z kontrolką zapisywane są wartości *key* dla wybranych pozycji, a na liście wyświetlane sa wartości *value* dla tych pozycji. Lista wielowyboru jest odpowiednia dla pól formularza o typie danych *nvarchar*.

Oprócz właściwości typowych dla większości kontrolek, dla listy wielowyboru należy określić następujące zależności:

- `Zapisz w tabeli relacyjnej`: jeśli ten przyłącznik jest wyłączony (domyślnie), wartości wybrane na liście wielowyboru są zapisywane w kolumnie tabeli źródłowej formularza w formacie *key1$,$key2$,$key3*..., gdzie *key1/2/3* to wartości kolumny *[key]* zwracanej przez zapytanie źródłowe dla kontrolki; po włączeniu przełącznika wartości są zapisywane osobno w tabeli *core.FormFIeldEnrollments* (razem z id definicji dokumentu, id bieżącego dokumentu i id pola formularza).

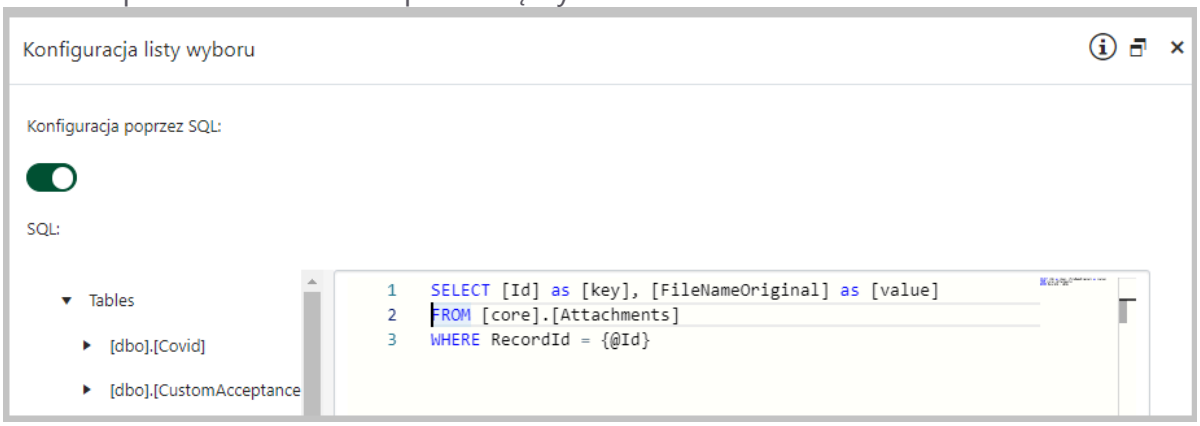
  UWAGA
  Niezależnie od stanu opcji zapisu w tabeli relacyjnej wartości z kolumny *key* są zapisywane jako tekst. Pozwala to obsłużyć wartości klucza o dowolnym typie danych. Jednak w przypadku, kiedy te wartości mają być następnie przetwarzane w ramach logiki biznesowej, należy pamiętać o ich konwersji na oryginalny typ danych.

- `Zachowuj wybrane wartości jako ustalone`: po włączeniu zapisuje wartości wyświetlane na liście wielowyboru w kolumnie *SelectedValue* tabeli *core.FormFIeldEnrollments*; funkcja dostępna tylko wtedy, gdy wybrane wartości są zapisywane w tabeli relacyjnej.
- `Zaznacz/odznacz wszystko`: po włączeniu dodaje do listy polecenie, która pozwala zaznaczyć wszystkie pozycje na liście lub anulować zaznaczenie wszystkich pozycji.
- `Pokazuj pełny tekst wybranych elementów`: domyślnie tekst wartości wyświetlanych na liście może zostać skrócony, po włączeniu tego przełącznika zawsze będzie wyświetlany pełny opis.

- `Szerokość wybranego elementu`: określa szerokość listy w jednostkach *rem*, a w przypadku listy wielowyboru, szerokość wybranej pozycji listy.

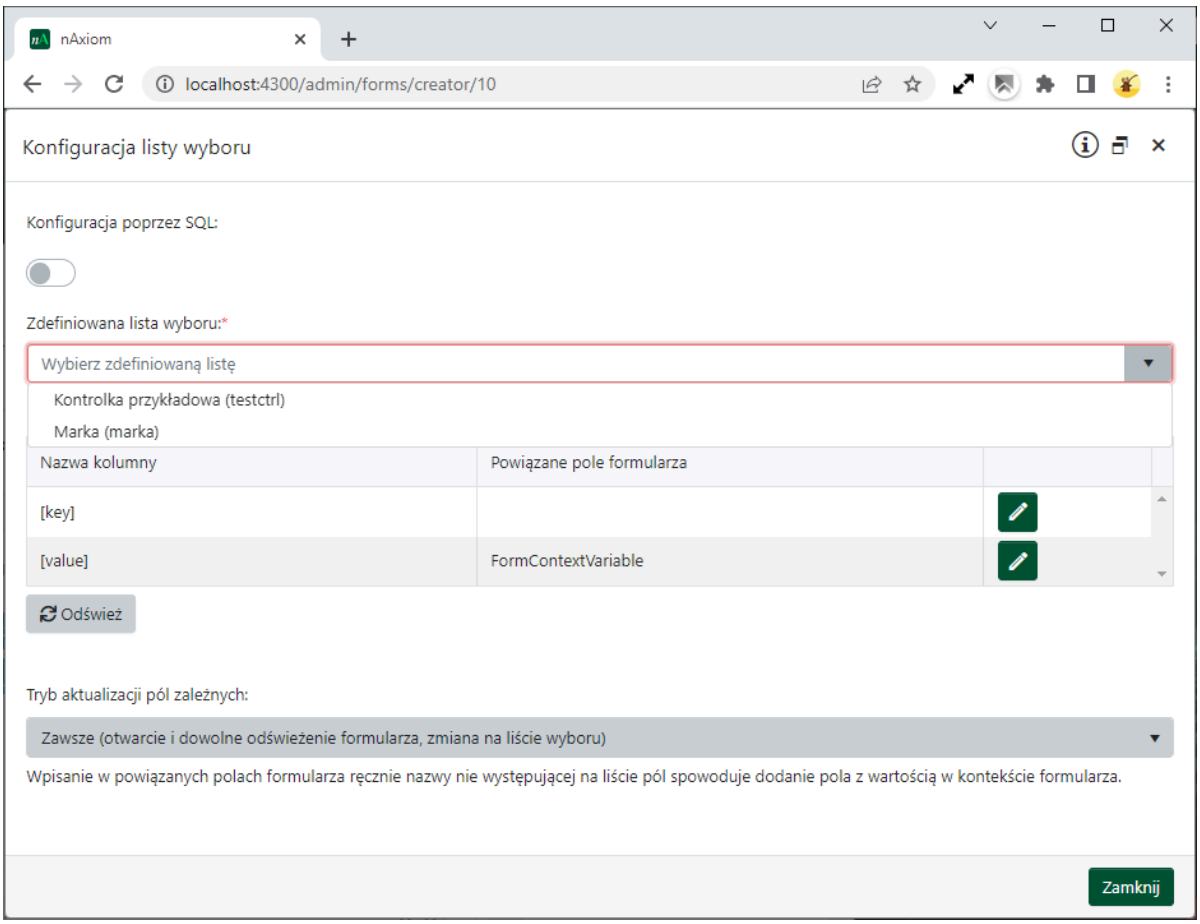
`Konfiguracja poprzez SQL`: (domyślnie włączone) pozwala definiować źródło danych dla listy jako zapytanie SQL lub (po wyłączeniu) przy użyciu zdefiniowanej osobno kontrolki.

- `SQL`: zapytanie SQL zwracające wartości listy; to zapytanie musi zwracać co najmniej dwie kolumny o nazwach *[key]* i *[value]*; wartości z kolumny *[key]* są zapisywane w polu formularza, a wartości z kolumny *[value]* (muszą być typu tekstowego) są wyświetlane na liście; zapytanie może zwracać inne kolumny, ich nazwy należy również podać w nawiasach prostokątnych.



SQL dla listy wyboru

- `Zdefiniowana lista wyboru`: lista zdefiniowanych kontrolek (`INTERFEJS UŻYTKOWNIKA` > `Kontrolki`)



Wybór predefiniowanej kontrolki

UWAGA

Dostępna w oknie dialogowym `Konfiguracja listy wyboru` tabela `Lista zależności: kolumna - pole formularza` nie ma zastosowania do listy wielowyboru.

Powiązane tematy:

- Typowe właściwości kontrolek
- Właściwości kontrolek - Zależności
- Właściwości kontrolek - Dostępność

4.1.6.12. LISTA WYBORU (INTERESARIUSZ)

Podsumowanie:
Lista wyboru interesariuszy to lista wielowyboru służąca do wyboru konkretnych osób jako członków grupy interesariuszy. Grupy interesariuszy definiuje się w definicji dokumentu. Grupa użytkowników na liście wyboru interesariuszy może być ograniczona przez rolę lub kombinację ról albo przez konfigurację zapytania SQL zwracającego dane do listy.

W aplikacji FrontSPA w kontrolce wyświetlane są także role i role w jednostkach przypisane do grupy interesariuszy, a także użytkownicy dodani za pomocą akcji, których nie ma na rozwijanej liście kontrolki. Kontrolka pozwala na ich usuwanie z grupy interesariuszy, ale ich dodawanie jest możliwe wyłącznie przy użyciu odpowiednich akcji.

Oprócz właściwości typowych dla większości kontrolek, lista wyboru interesariuszy wymaga ustawienia właściwości:

- `Interesariusz`: nazwa grupy interesariuszy określonej w definicji dokumentu.
- `Minimalna liczba użytkowników`: minimalna liczba osób, które trzeba wybrać na liście; jeśli zostanie ustawiono, nie będzie można zapisać formularza bez spełnienia tego warunku, nawet jeśli dla kontrolki będzie wyłączona wymagalność.
- `Maksymalna liczba użytkowników`: maksymalna liczba osób, które można wybrać na liście.
- `Wymagane role globalne`: jedna lub kilka ról biznesowych, które muszą mieć użytkownicy, aby byli dostępni na liście; w przypadku kilku ról wyświetlane będą osoby, które maję przypisane wszystkie role; ustawienie tej właściwości powoduje modyfikację zapytania SQL zwracającego dane do listy.
- `Jako opis jednostki organizacyjnej wyświetlaj`: określa sposób wyświetlania jednostek organizacyjnych, które mogą należeć do grupy interesariuszy; dostępne wartości:
  - `Kod JO`: wyświetla kod jednostki organizacyjnej.
  - `Nazwę JO`: wyświetla nazwę jednostki organizacyjnej.
- `Wartość domyślna dla użytkowników`: można wpisać identyfikator GUID dowolnego użytkownika lub zapytanie, które zwróci listę identyfikatorów użytkowników, na przykład:

```
SELECT [UserProfileId]
FROM [core].[OUInstanceUsers]
WHERE [OUInstanceId] = 17
```

> **UWAGA**
> Próba ustawienia wartością domyślną użytkownika, którego nie ma na liście wyboru, nie powiedzie się bez zgłaszania błędu.

- `Zaznacz/odznacz wszystko`: po włączeniu dodaje do listy polecenie, która pozwala zaznaczyć wszystkie pozycje na liście lub anulować zaznaczenie wszystkich pozycji.
- `Szerokość wybranego elementu`: określa szerokość listy w jednostkach *rem*, a w przypadku listy wielowyboru, szerokość wybranej pozycji listy.
- `Ustawienia`: kliknięcie przycisku `Konfiguracja listy wyboru` otwiera okno dialogowe, o tej samej nazwie.
- `SQL`: zapytanie SQL zwracające wartości identyfikatorów użytkowników z tabeli *[core].[UserProfiles]*; to zapytanie musi zwracać co najmniej dwie kolumny o nazwach *[key]* i *[value]*; wartości z pola *Id* mają być zwracane w kolumnie *[key]*, a w kolumnie *[value]* mogą być zwracane na przykład wartości z kolumny *UserName* (muszą to być wartości typu tekstowego); wartość zwrócona w kolumnie *[value]* jest wartością wyświetlaną w kontrolce; nazwy kolumn należy podać w nawiasach prostokątnych. Przykład:

```
SELECT [Id] as [key], [UserName] as [value] FROM [core].[UserProfiles]
```

> **WSKAZÓWKA**
> Ustawiony dla kontrolki warunek wymagalności będzie spełniony także wtedy, gdy w kontrolce będzie wyświetlana rola lub rola w jednostce przypisane do danej grupy interesariuszy. Aby wymusić na użytkowniku końcowym dodanie indywidualnej osoby do grupy przy użyciu kontrolki, należy ustawić właściwość `Minimalna liczba użytkowników`.

Informacje o wartościach kontrolki wyboru interesariuszy są przechowywane w kontekście formularza w tablicy *StakeholdersData* z identyfikatorami obiektów przypisanych do grupy interesariuszy podzielonymi na kategorie: *UserIds*, *GlobalRoleIds* i *OrgUnitRoleIds*.

Tablica StakeholdersData w kontekście formularza

Powiązane tematy:

- Typowe właściwości kontrolek
- Właściwości kontrolek - Zależności
- Właściwości kontrolek - Dostępność
- Uprawnienia ACL do dokumentów
- Przypisywanie ról globalnych do interesariuszy
- Przypisywanie ról w jednostkach organizacyjnych do interesariuszy

4.1.6.13. LISTA WYBORU (SŁOWNIK)

Podsumowanie:
Lista wyboru typu słownik to lista wyboru, której źródłem danych jest zdefiniowany osobno słownik. Wyświetlane na liście pozycje słownika są definiowane przez użytkowników końcowych w aplikacji *FrontSPA*. Lista wyboru typu słownik może być powiązana z polem formularza typu liczba całkowita, w którym zostanie zapisany identyfikator wybranej pozycji słownika.

Oprócz właściwości typowych dla większości kontrolek, lista wyboru typu słownik wymaga ustawienia właściwości:

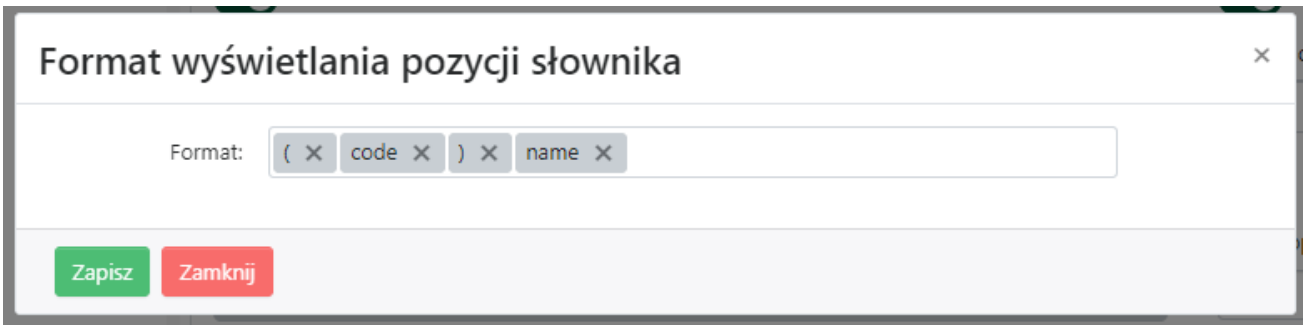
`Pole wartości ustalonej`: wybierz z listy kolumnę w tabeli źródłowej formularza, w której zostanie zapisana wartość wyświetlana na liście rozwijanej (wartość z kolumny *[value]* zwracanej przez zapytanie w konfiguracji listy). Po zapisaniu instancji dokumentu ta wartość będzie wyświetlana na formularzu nawet wtedy, gdy odpowiadająca wartość w źródle danych listy rozwijanej zostanie zmieniona. W takiej sytuacji, po rozwinięciu listy podświetlana będzie zaktualizowana wartość odpowiadająca wartości ustalonej.

UWAGA

Pole wartości ustalonej nie jest widoczne w kontekście formularza ( ), a przypisana do niego kontrolka formularzu nie będzie wyświetlać danych.

`Słownik dla listy`: nazwa słownika, który będzie źródłem danych dla listy.

`Pozycje słownika`: przycisk `Format` otwiera okno dialogowe, w którym można określić format wyświetlania wpisów ze słownika. Format określa się na liście wielowyboru z możliwością edycji, wybierając predefiniowane atrybuty wpisu do słownika, w tym *code*, *name*, *description*, *numericDescription* i inne, oraz wpisując dodatkowe znaki i/lub tekst, które poprawią czytelność wyświetlanych danych (np. nawiasy, myślnik, dwukropek, ukośnik itp.).



`Wyświetlaj pozycje słownika`: dostępne wartości:

- `Wszystkie`: wyświetla wszystkie pozycje słownika, niezaleznie od ich aktywności,
- `Tylko aktywne`: wyświetla tylko aktywne pozycje słownika,
- `Tylko nieaktywne`: wyświetla tylko nieaktywne pozycje słownika.

Powiązane tematy:

- Typowe właściwości kontrolek
- Właściwości kontrolek - Zależności
- Właściwości kontrolek - Dostępność
- Słowniki

4.1.6.14. LISTA WYBORU (SŁOWNIK HIERARCHICZNY)

Podsumowanie:
Lista wyboru typu słownik hierarchiczny to lista wyboru, której źródłem danych jest zdefiniowany osobno słownik hierarchiczny. Słownik hierarchiczny różni się od zwykłego słownika tym, że jego pozycje mogą zawierać pozycje podrzędne. Wyświetlane na liście pozycje słownika są definiowane przez użytkowników końcowych w aplikacji *FrontSPA*. Lista wyboru typu słownik hierarchiczny może być powiązana z polem formularza typu liczba całkowita, w którym zostanie zapisany identyfikator wybranej pozycji słownika.

Oprócz właściwości typowych dla większości kontrolek, lista wyboru typu słownik hierarchiczny wymaga ustawienia właściwości:

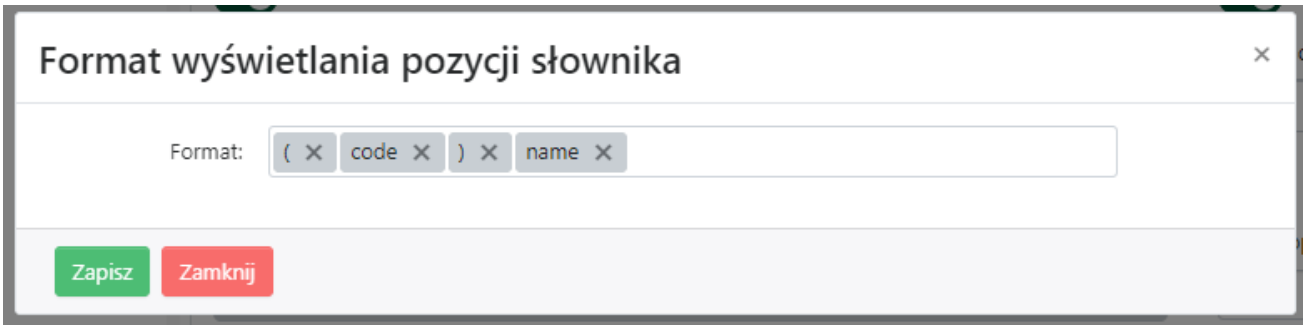
`Pole wartości ustalonej`: wybierz z listy kolumnę w tabeli źródłowej formularza, w której zostanie zapisana wartość wyświetlana na liście rozwijanej (wartość z kolumny *[value]* zwracanej przez zapytanie w konfiguracji listy). Po zapisaniu instancji dokumentu ta wartość będzie wyświetlana na formularzu nawet wtedy, gdy odpowiadająca wartość w źródle danych listy rozwijanej zostanie zmieniona. W takiej sytuacji, po rozwinięciu listy podświetlana będzie zaktualizowana wartość odpowiadająca wartości ustalonej.

UWAGA

Pole wartości ustalonej nie jest widoczne w kontekście formularza ( ), a przypisana do niego kontrolka formularzu nie będzie wyświetlać danych.

`Słownik dla listy`: nazwa słownika, który będzie źródłem danych dla listy.

`Pozycje słownika`: przycisk `Format` otwiera okno dialogowe, w którym można określić format wyświetlania wpisów ze słownika. Format określa się na liście wielowyboru z możliwością edycji, wybierając predefiniowane atrybuty wpisu do słownika, w tym *code*, *name*, *description*, *numericDescription* i inne, oraz wpisując dodatkowe znaki i/lub tekst, które poprawią czytelność wyświetlanych danych (np. nawiasy, myślnik, dwukropek, ukośnik itp.).



Powiązane tematy:

- Typowe właściwości kontrolek
- Właściwości kontrolek - Zależności
- Właściwości kontrolek - Dostępność
-

4.1.6.15. LISTA WYBORU (SQL - ASYNC)

Podsumowanie:
Lista wyboru *SQL - Async* to lista z funkcją autouzupełniania. Po wpisaniu przez użytkownika określonej liczby znaków wczytywany jest podzbiór pozycji listy, które zawierają wpisany ciąg.

Oprócz właściwości typowych dla większości kontrolek, lista wyboru SQL - Async wymaga ustawienia właściwości:

`Pole wartości ustalonej`: wybierz z listy kolumnę w tabeli źródłowej formularza, w której zostanie zapisana wartość wyświetlana na liście rozwijanej (wartość z kolumny *[value]* zwracanej przez zapytanie w konfiguracji listy). Po zapisaniu instancji dokumentu ta wartość będzie wyświetlana na formularzu nawet wtedy, gdy odpowiadająca wartość w źródle

danych listy rozwijanej zostanie zmieniona. W takiej sytuacji, po rozwinięciu listy podświetlana będzie zaktualizowana wartość odpowiadająca wartości ustalonej.

UWAGA

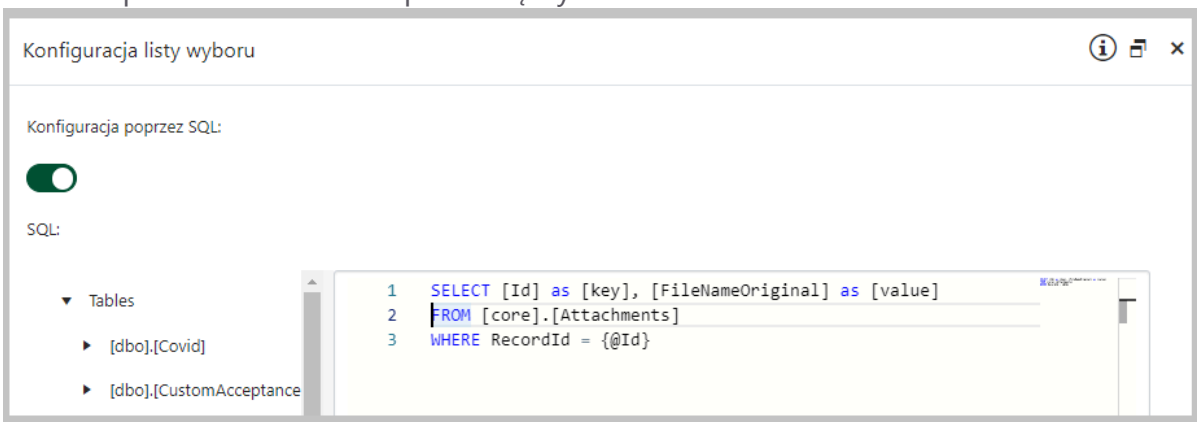
Pole wartości ustalonej nie jest widoczne w kontekście formularza ( ), a przypisana do niego kontrolka formularzu nie będzie wyświetlać danych.

- `Wypełnij po`: liczba znaków, po wpisaniu których lista wyświetli pasujące wartości; aby lista działała zgodnie z przeznaczeniem, nie wolno wpisywać wartości 0 ani wartości przekraczającej długość pozycji na liście.

INFO

Wpisany ciąg znaków jest wyszukiwany w całym tekście pozycji listy, np. jeśli lista zawiera nazwy miejscowości i wymagane jest wpisanie trzech znaków, dla ciągu *rze* zostaną zwrócone między innymi *Przeworsk*, *Rzeszów* i *Zabrze*.

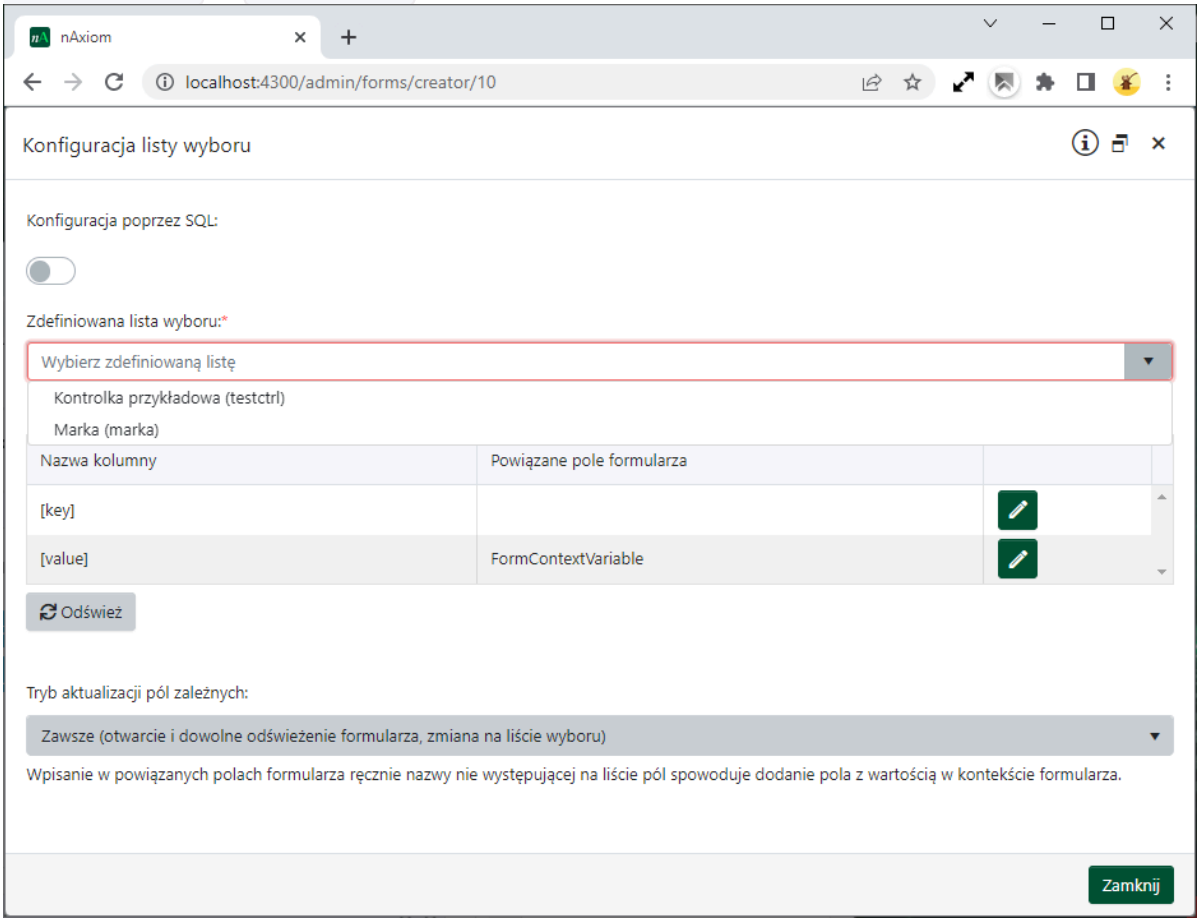
- `Maks. liczba elementów`: liczba pozycji listy zgodnych z wpisanym ciągiem, które zostaną wyświetlone na liście; wartość domyślna 50; ma na celu ograniczenie czasu wczytywania pozycji listy.
- `Ustawienia`: kliknięcie przycisku `Konfiguracja listy wyboru` otwiera okno dialogowe, o tej samej nazwie.
- `Konfiguracja poprzez SQL`: (domyślnie włączone) pozwala definiować źródło danych dla listy jako zapytanie SQL lub (po wyłączeniu) przy użyciu zdefiniowanej osobno kontrolki.
- `SQL`: zapytanie SQL zwracające wartości listy; to zapytanie musi zwracać co najmniej dwie kolumny o nazwach *[key]* i *[value]*; wartości z kolumny *[key]* są zapisywane w polu formularza, a wartości z kolumny *[value]* (muszą być typu tekstowego) są wyświetlane na liście; zapytanie może zwracać inne kolumny, ich nazwy należy również podać w nawiasach prostokątnych.



SQL dla listy wyboru

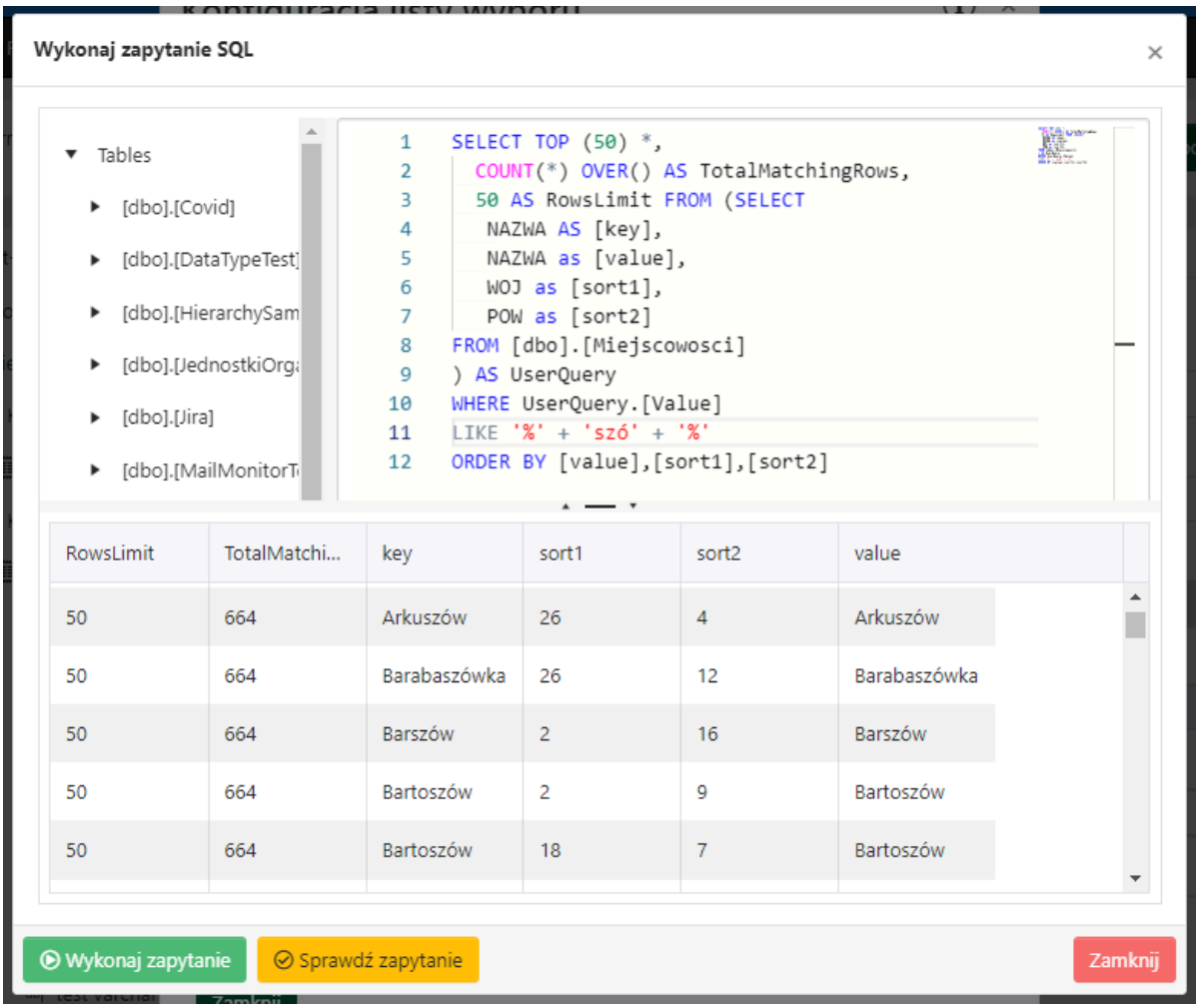
- `Zdefiniowana lista wyboru`: lista zdefiniowanych kontrolek (`INTERFEJS`

`UŻYTKOWNIKA` > `Kontrolki`)



Wybór predefiniowanej kontrolki

- `Metoda sortowania zwracanych danych`: lista wielowyboru, z której należy wybrać aliasy kolumn zdefiniowanych w zapytaniu SQL w celu posortowania pozycji na liście.
  `Pokaż zapytanie wynikowe`: wyświetla okno edytora SQL z zapytaniem które dopasuje wpisany ciąg znaków i zwróci określoną liczbę pozycji dla listy asynchronicznej; w tym oknie dostępny jest przycisk `Wykonaj zapytanie`, który pozwala wyświetlić podgląd pozycji zwróconych dla listy z uwzględnieniem ustawionego sortowania.



Podgląd ustawień sortowania dla asynchronicznej listy wyboru

- `Lista zależności: kolumna - pole formularza`: tabelka, która pozwala przypisać kolumnie zapytania źródłowego listy wyboru pole formularza. Kliknij ikonę edycji i wybierz pole formularza z rozwijanej listy. Przypisanie spowoduje, że po wybraniu pozycji z listy w powiązanym polu zostanie ustawiona wartość zwracana przez listę dla tej pozycji w odpowiedniej kolumnie.

> **WSKAZÓWKA**
> Zamiast wybierać pola formularza, w kolumnie `Powiązane pole formularza` można także wpisać dowolną nazwę; spowoduje to utworzenie zmiennej w kontekście formularza, w której będzie przechowywana wartość zwrócona przez listę. Do takiej zmiennej można odwoływać się w wyrażeniach, używając standardowej składni *{@nazwaZmiennej}*.

- `Tryb aktualizacji pól zależnych`: dostępne wartości:
  - `Zawsze`: każda operacja otwarcia i odświeżenia formularza oraz zmiana na liście wyboru powoduje aktualizację wartości w polach zależnych.
  - `Tylko po zmianie na liście wyboru`: aktualizacja wartości w polach zależnych odbywa się tylko po zmianie pozycji w liście rozwijanej (zmiana

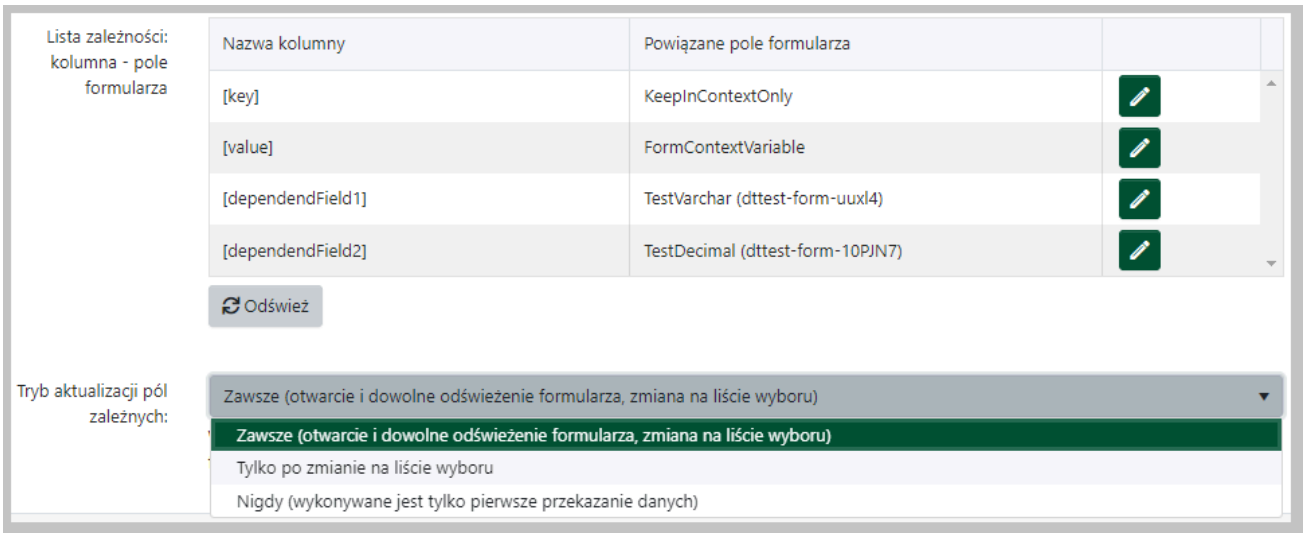
przez użytkownika lub np. przez akcję).

- `Tylko gdy wskazane pole jest puste`: wartości zwrócone przez listę wyboru są przekazywane do pól zależnych tylko wtedy, kiedy są one puste.

Na efekty działania tej funkcji ma wpływ zachowanie nAxiom przy odświeżaniu formularza. Kliknięcie ikony odświeżania w prawym górnym rogu formularza inicjuje działanie zależnie od typu okna formularza:

- `Dialog`: zostaje ponownie wykonana akcja otwierania formularza z użyciem przekazywanych w niej parametrów.
- `Link wewnętrzny`: formularz i dane rekordu zostają ponownie wczytane z bazy danych; ewentualne parametry przekazywane w akcji otwierania formularza zostają utracone. Tak samo działa przeładowanie strony w przeglądarce klawiszem F5.

Akcja odświeżenia formularza działa tak samo, niezależnie od typu okna formularza, to jest wczytuje ponownie formularz i dane rekordu z bazy danych. Jeśli formularz został otwarty akcją z przekazaniem parametrów, parametry zostaną utracone.



Tryb aktualizacji pól zależnych

Powiązane tematy:

- Typowe właściwości kontrolek
- Właściwości kontrolek - Zależności
- Właściwości kontrolek - Dostępność

Podsumowanie:
Lista wyboru SQL to lista rozwijana, z której można wybrać jedną wartość. Może być używana dla pól różnego typu, o ile definiowana w zapytaniu kolumna *[key]* będzie zawierać dane zgodnego typu.

Oprócz właściwości typowych dla większości kontrolek, dla listy wyboru SQL należy ustawić właściwości: `Pole wartości ustalonej`: wybierz z listy kolumnę w tabeli źródłowej formularza, w której zostanie zapisana wartość wyświetlana na liście rozwijanej (wartość z kolumny *[value]* zwracanej przez zapytanie w konfiguracji listy). Po zapisaniu instancji dokumentu ta wartość będzie wyświetlana na formularzu nawet wtedy, gdy odpowiadająca wartość w źródle danych listy rozwijanej zostanie zmieniona. W takiej sytuacji, po rozwinięciu listy podświetlana będzie zaktualizowana wartość odpowiadająca wartości ustalonej.

UWAGA

Pole wartości ustalonej nie jest widoczne w kontekście formularza ( ), a przypisana do niego kontrolka formularzu nie będzie wyświetlać danych.

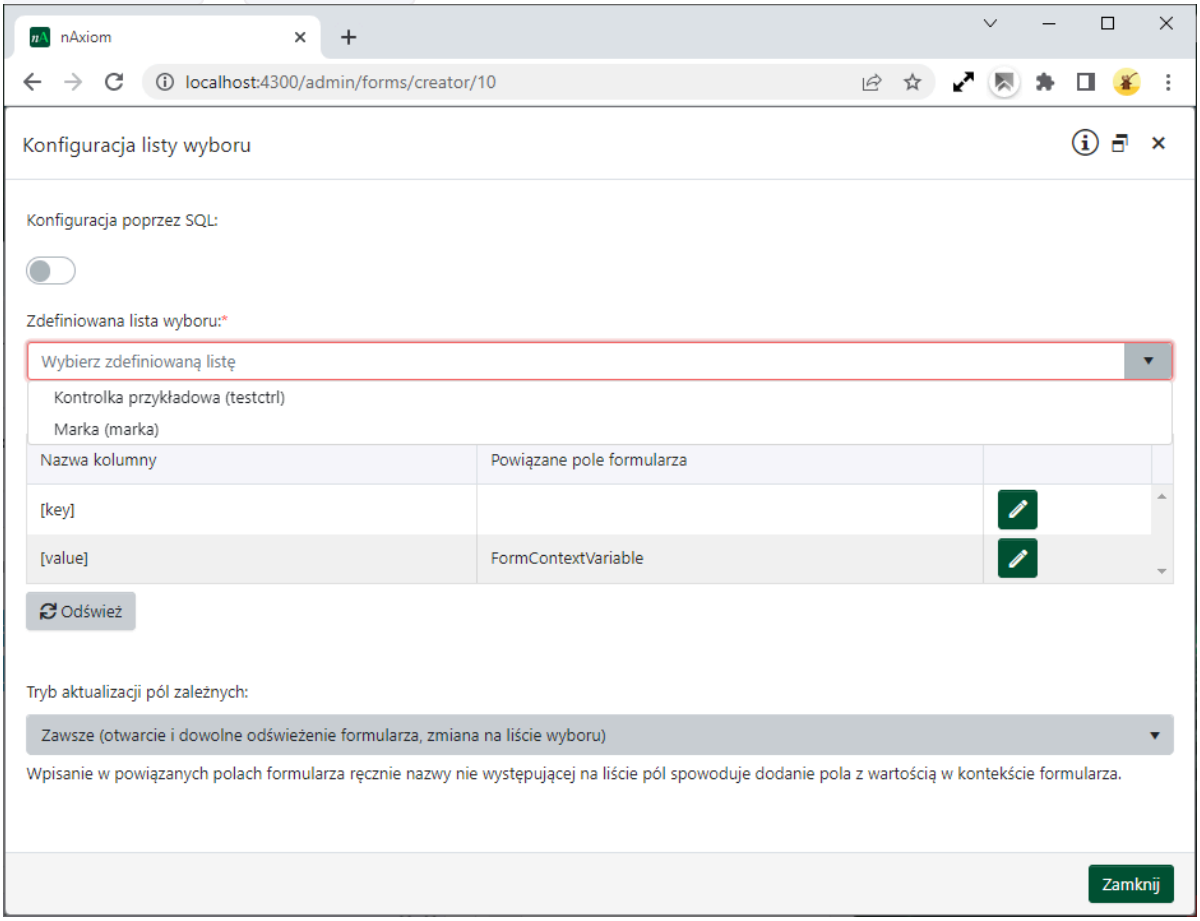
- `Używaj stałej szerokości...`: włączenie tego przełącznika ustala szerokość listy wyboru do wartości podanej w polu `Szerokość wybranego elementu`; jeśli wyłącznik będzie wyłączony, lista wypełni wszerz dostępne miejsce w sekcji formularza.
- `Szerokość wybranego elementu`: określa szerokość listy w jednostkach *rem*, a w przypadku listy wielowyboru, szerokość wybranej pozycji listy.

`Ustawienia`: kliknięcie przycisku `Konfiguracja listy wyboru` otwiera okno dialogowe, o tej samej nazwie. `Konfiguracja poprzez SQL`: (domyślnie włączone) pozwala definiować źródło danych dla listy jako zapytanie SQL lub (po wyłączeniu) przy użyciu zdefiniowanej osobno kontrolki.

- `SQL`: zapytanie SQL zwracające wartości listy; to zapytanie musi zwracać co najmniej dwie kolumny o nazwach *[key]* i *[value]*; wartości z kolumny *[key]* są zapisywane w polu formularza, a wartości z kolumny *[value]* (muszą być typu tekstowego) są wyświetlane na liście; zapytanie może zwracać inne kolumny, ich nazwy należy również podać w nawiasach prostokątnych.

SQL dla listy wyboru

- `Zdefiniowana lista wyboru`: lista zdefiniowanych kontrolek (`INTERFEJS UŻYTKOWNIKA` > `Kontrolki`)



Wybór predefiniowanej kontrolki

- `Lista zależności: kolumna - pole formularza`: tabelka, która pozwala przypisać kolumnie zapytania źródłowego listy wyboru pole formularza. Kliknij ikonę edycji i wybierz pole formularza z rozwijanej listy. Przypisanie spowoduje, że po wybraniu pozycji z listy w powiązanym polu zostanie ustawiona wartość zwracana przez listę dla tej pozycji w odpowiedniej kolumnie.

WSKAZÓWKA

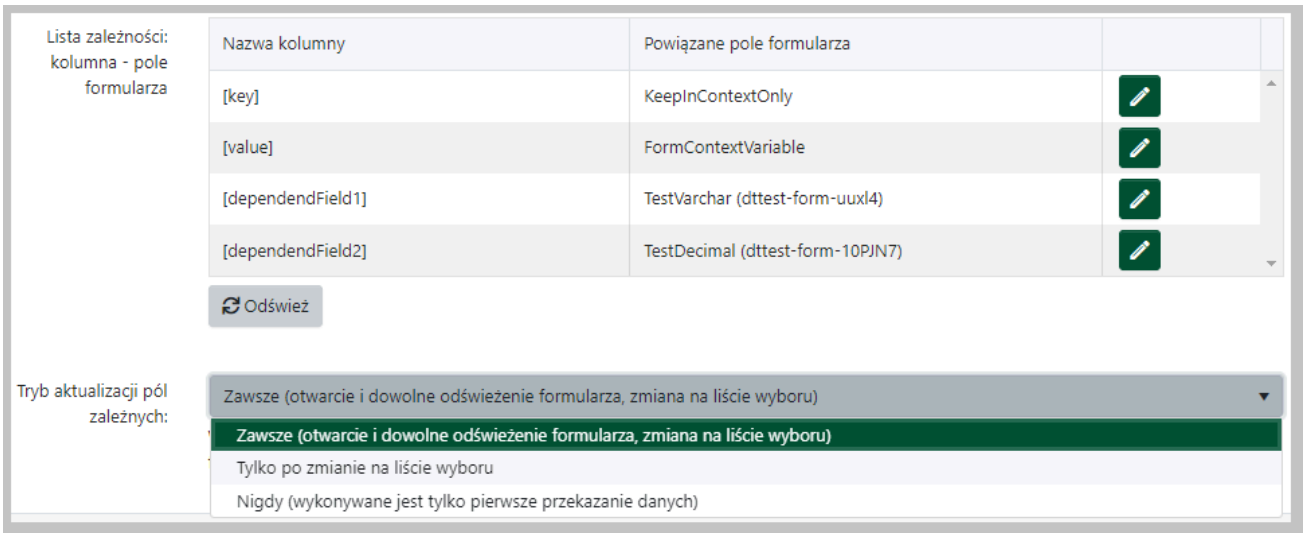
Zamiast wybierać pola formularza, w kolumnie `Powiązane pole formularza` można także wpisać dowolną nazwę; spowoduje to utworzenie zmiennej w kontekście formularza, w której będzie przechowywana wartość zwrócona przez listę. Do takiej zmiennej można odwoływać się w wyrażeniach, używając standardowej składni *{@nazwaZmiennej}*.

- `Tryb aktualizacji pól zależnych`: dostępne wartości:
  - `Zawsze`: każda operacja otwarcia i odświeżenia formularza oraz zmiana na liście wyboru powoduje aktualizację wartości w polach zależnych.
  - `Tylko po zmianie na liście wyboru`: aktualizacja wartości w polach zależnych odbywa się tylko po zmianie pozycji w liście rozwijanej (zmiana przez użytkownika lub np. przez akcję).
  - `Tylko gdy wskazane pole jest puste`: wartości zwrócone przez listę wyboru są przekazywane do pól zależnych tylko wtedy, kiedy są one puste.

Na efekty działania tej funkcji ma wpływ zachowanie nAxiom przy odświeżaniu formularza. Kliknięcie ikony odświeżania w prawym górnym rogu formularza inicjuje działanie zależnie od typu okna formularza:

- `Dialog`: zostaje ponownie wykonana akcja otwierania formularza z użyciem przekazywanych w niej parametrów.
- `Link wewnętrzny`: formularz i dane rekordu zostają ponownie wczytane z bazy danych; ewentualne parametry przekazywane w akcji otwierania formularza zostają utracone. Tak samo działa przeładowanie strony w przeglądarce klawiszem F5.

Akcja odświeżenia formularza działa tak samo, niezależnie od typu okna formularza, to jest wczytuje ponownie formularz i dane rekordu z bazy danych. Jeśli formularz został otwarty akcją z przekazaniem parametrów, parametry zostaną utracone.



Tryb aktualizacji pól zależnych

Powiązane tematy:

- Typowe właściwości kontrolek
- Właściwości kontrolek - Zależności
- Właściwości kontrolek - Dostępność

4.1.6.17. LISTA WYBORU (TENANT NAXIOM)

Podsumowanie:
Lista wyboru tenantów pozwala wybrać dowolnego tenanta zdefiniowanego w danym środowisku nAxiom. Jest przeznaczona do „adresowania" korespondencji między tenantami, to jest do wskazania identyfikatora tenanta w akcji zapytań do web service wykorzystującej źródła danych typu Tenant w środowisku nAxiom .

Oprócz właściwości typowych dla większości kontrolek, lista wyboru tenantów nie wymaga ustawienia specjalnych właściwości.

Lista zwraca następujące kolumny:

- *Id* (GUID)
- *Code*
- *Name*
- *UrlPrefix*

Listę należy powiązać z polem typu *uniqueidetifier* lub *varchar/nvarchar*. Wartości zwracane przez listę można przypisać do innych pól na formularzu, po kliknięciu przycisku `Konfiguracja listy wyboru`.

Powiązane tematy:

- Typowe właściwości kontrolek
- Właściwości kontrolek - Zależności
- Właściwości kontrolek - Dostępność

4.1.6.18. LISTA WYBORU (WEBSERVICE)

Podsumowanie:
Lista wyboru WebService to lista rozwijana, dla której źródłem danych jest akcja WebService. Odpowiedź zwrócona przez akcję jest przetwarzana zapytaniem SQL na format akceptowany przez listy wyboru w nAxiom (wynik zapytania o kolumnach *[key]*/*[value]*).

Oprócz właściwości typowych dla większości kontrolek, lista wyboru WebService wymaga ustawienia właściwości:

- `Pole wartości ustalonej`: wybierz z listy kolumnę w tabeli źródłowej formularza, w której zostanie zapisana wartość wyświetlana na liście rozwijanej (wartość z kolumny *[value]* zwracanej przez zapytanie w konfiguracji listy). Po zapisaniu instancji dokumentu ta wartość będzie wyświetlana na formularzu nawet wtedy, gdy odpowiadająca wartość w źródle danych listy rozwijanej zostanie zmieniona. W takiej sytuacji, po rozwinięciu listy podświetlana będzie zaktualizowana wartość odpowiadająca wartości ustalonej.

UWAGA

Pole wartości ustalonej nie jest widoczne w kontekście formularza ( ), a przypisana do niego kontrolka formularzu nie będzie wyświetlać danych.

- `Ustawienia`: kliknięcie przycisku `Konfiguracja listy wyboru` wyświetla okno dialogowe, w którym należy określić właściwości:
- `Akcja web serwisu`: akcja WebService, która będzie źródłem danych dla listy. W konfiguracji listy wyboru Webservice należy określić osobną procedurę obsługi odpowiedzi, która przetworzy dane zwrócone przez akcję tak, aby mogły być wyświetlane na liście wyboru. Procedura obsługi odpowiedzi określona w definicji akcji jest ignorowana.

- `Tryb obsługi odpowiedzi`: dostępne wartości:
  - `domyślny`: zapytanie SQL określone jako procedura obsługi odpowiedzi jest wykonywane w przypadku, gdy zostanie zwrócony status HTTP z zakresu 200-299; w przeciwnym razie akcja zwróci wyjątek.
  - `własny warunek obsługi odpowiedzi`: w ramach procedury odpowiedzi możesz obsłużyć każdy status HTTP odpowiedzi; patrz przykład poniżej.

`Procedura obsługi odpowiedzi`: zapytanie zwracające listę wartości z kolumnami *[key]*/*[value]*. W zapytaniu można korzystać ze zmiennych *{@_Response}*, *{@_HTTPResponseStatusCode}*, *{@_HTTPResponseStatusName}* i *{@_HTTPResponseHeader}*.

W zapytaniu dostępne są następujące zmienne:

- *@_Response*: treść (*body*) odpowiedzi; jeśli dla zwracanych danych określono format JSON, w zapytaniu użyj funkcji *OPENJSON*.
- *@_HTTPResponseStatusCode*: status odpowiedzi (*int*) np. *404*.
- *@_HTTPResponseStatusName*: nazwa statusu odpowiedzi np. *NotFound* dla statusu *404*.
- *@_HTTPResponseHeader*: nagłówki odpowiedzi w formacie JSON o następującej strukturze:

  ```
  {
  "header_1": ["value_1"],
  "header_2": ["value_2", "value_3"]
  }
  ```

  gdzie *header_x* to nazwa nagłówka odpowiedzi, a *value_x* to jego wartość. Wartości nagłówków można pozyskać używając funkcji SQL *OPENJSON*. W przypadku, kiedy nazwa nagłówka zawiera myślnik (-), odpowiadający mu klucz należy ująć w cudzysłów, na przykład:

  ```
  (SELECT top(1) value
   FROM OPENJSON({@_HTTPResponseHeader}, '$."Content-Type"')
  ```

- `Lista zależności: kolumna - pole formularza`: tabelka, która pozwala przypisać kolumnie zapytania źródłowego listy wyboru pole formularza. Kliknij ikonę edycji i wybierz pole formularza z rozwijanej listy. Przypisanie spowoduje, że po wybraniu pozycji z listy w powiązanym polu zostanie ustawiona wartość zwracana przez listę dla tej pozycji w odpowiedniej kolumnie.

> **WSKAZÓWKA**
> Zamiast wybierać pola formularza, w kolumnie `Powiązane pole formularza` można także wpisać dowolną nazwę; spowoduje to utworzenie zmiennej w kontekście formularza, w której będzie przechowywana wartość zwrócona przez listę. Do takiej zmiennej można odwoływać się w wyrażeniach, używając standardowej składni *{@nazwaZmiennej}*.

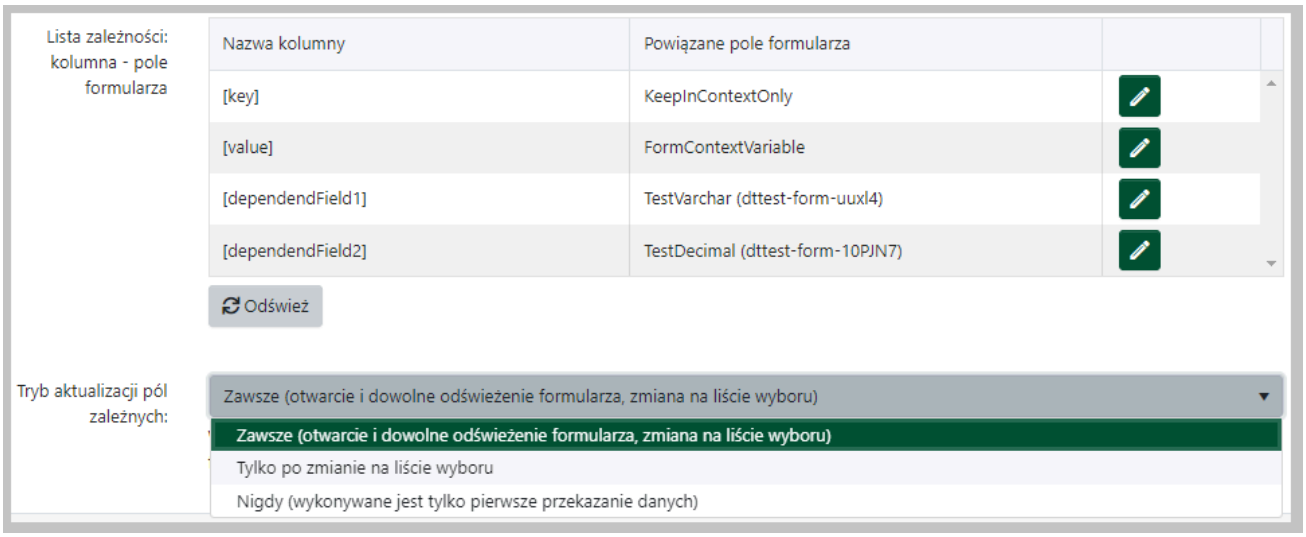
- `Tryb aktualizacji pól zależnych`: dostępne wartości:
  - `Zawsze`: każda operacja otwarcia i odświeżenia formularza oraz zmiana na liście wyboru powoduje aktualizację wartości w polach zależnych.
  - `Tylko po zmianie na liście wyboru`: aktualizacja wartości w polach zależnych odbywa się tylko po zmianie pozycji w liście rozwijanej (zmiana

przez użytkownika lub np. przez akcję).
    - `Tylko gdy wskazane pole jest puste`: wartości zwrócone przez listę wyboru są przekazywane do pól zależnych tylko wtedy, kiedy są one puste.

Na efekty działania tej funkcji ma wpływ zachowanie nAxiom przy odświeżaniu formularza. Kliknięcie ikony odświeżania w prawym górnym rogu formularza inicjuje działanie zależnie od typu okna formularza:

- `Dialog`: zostaje ponownie wykonana akcja otwierania formularza z użyciem przekazywanych w niej parametrów.
- `Link wewnętrzny`: formularz i dane rekordu zostają ponownie wczytane z bazy danych; ewentualne parametry przekazywane w akcji otwierania formularza zostają utracone. Tak samo działa przeładowanie strony w przeglądarce klawiszem F5.

Akcja odświeżenia formularza działa tak samo, niezależnie od typu okna formularza, to jest wczytuje ponownie formularz i dane rekordu z bazy danych. Jeśli formularz został otwarty akcją z przekazaniem parametrów, parametry zostaną utracone.



Tryb aktualizacji pól zależnych

PRZYKŁAD 1

Zapytanie przetwarzające odpowiedź ze źródła SOAP API w celu wyświetlenia wartości na liście wyboru (wymagane skonfigurowanie źródła danych i akcji zapytania do webserwisu).

```
DECLARE
  @xmlRes XML = REPLACE(ISNULL({@_Response},'ERROR'), '<?xml version="1.0" encoding="utf-8"?>',''),
  @msg AS nvarchar (1000);
WITH XMLNAMESPACES (DEFAULT 'http://www.oorsprong.org/websamples.countryinfo')
SELECT
```

```
  l.value('sName[1]','varchar(30)') AS [key],  -- wartość zapisywana do bazy danych
  l.value('sName[1]','varchar(30)') AS [value]   -- wartość wyświetlana
FROM @xmlRes.nodes('//ListOfContinentsByNameResult') r(n)
CROSS APPLY r.n.nodes('tContinent') lines(l)
```

PRZYKŁAD 2

Zapytanie przetwarzające odpowiedź ze źródła REST API w celu wyświetlenia wartości na liście wyboru (wymagane skonfigurowanie źródła danych i akcji zapytania do webserwisu).

```
SELECT code AS [key],
  CONCAT(currency,' - ',mid,' PLN') AS [value]
FROM OPENJSON ({@_Response}, '$[0].rates')
  WITH (
    currency VARCHAR(200)
   ,code VARCHAR(200)
   ,mid VARCHAR(200)
   )
  AS JSON
```

Powiązane tematy:

- Typowe właściwości kontrolek
- Właściwości kontrolek - Zależności
- Właściwości kontrolek - Dostępność
- Akcja zapytań do webserwisów

4.1.6.19. OBSZAR TEKSTOWY (TEXTAREA)

Podsumowanie:
Obszar tekstowy to kontrolka w formie prostokąta, w którą można wpisywać dane tekstowe w wielu wierszach. Zwykle pola tekstowego używa się do wprowadzania danych do kolumn typu *varchar* i *nvarchar*. W pole tekstowe można wpisać dowolną liczbę znaków, jeśli jednak będzie ona większa niż rozmiar zadeklarowany dla kolumny w źródle danych (np. *nvarchar(500)*), podczas próby zapisu rekordu zostanie wyświetlony komunikat o błędzie i rekord nie zostanie zapisany.

Właściwości tej kontrolki opisano w rozdziałach z sekcji *Powiązane tematy*.

Powiązane tematy:

- Typowe właściwości kontrolek
- Właściwości kontrolek - Dostępność
- Obszar tekstowy (TextArea)

4.1.6.20. POLE CZASU

Podsumowanie:
Kontrolka *Pole czasu* to prostokąt służący do wprowadzania wartości czasu (godziny). Tej kontrolki można używać dla pól typu *datetime*.

Z prawej strony kontrolki wyświetlana jest ikona zegarka, której kliknięcie wyświetla wyskakujące okienko do ustawiania czasu. W zależności od wybranego formatu, można ustawić godziny i minuty lub godziny, minuty i sekundy.

Właściwości tej kontrolki opisano w rozdziałach z sekcji *Powiązane tematy*.

Powiązane tematy:

- Typowe właściwości kontrolek
- Właściwości kontrolek - Dostępność
- Formatowanie daty i czasu

4.1.6.21. POLE DATY

Podsumowanie:
Kontrolka *Pole daty* to prostokąt służący do wprowadzania wartości daty. Tej kontrolki można używać dla pól typu *datetime* i *date*.

Format daty w tej kontrolce to *RRRR-mm-dd*. W pustej kontrolce wyświetlana jest podpowiedź *Rok-Miesiąc-Dzień*. Domyślnie kontrolka jest wyposażona w przyciski strzałek z prawej strony, które służą do zmniejszania/zwiększania wartości liczbowej zaznaczonego segmentu daty. Przyciski te można wyłączyć przełącznikiem `Spinner` we właściwościach kontrolki.

UWAGA
Od wersji 1.14 strzałki zmiany wartości pozwalają zmienić tylko pierwszy segment daty (czyli rok). Dowolny zaznaczony segment daty można zmienić za pomocą strzałek w górę i w dół na klawiaturze.

Pozostałe właściwości tej kontrolki opisano w rozdziałach z sekcji *Powiązane tematy*.

Powiązane tematy:

- Typowe właściwości kontrolek
- Właściwości kontrolek - Dostępność

4.1.6.22. POLE HASŁA

Podsumowanie:
Kontrolka *Pole hasła* to prostokąt służący do wprowadzania wartości hasła. Zamiast wartości wpisywanej w tym polu są wyświetlane kropki. Tej kontrolki można używać dla pól typu *varchar* i *nvarchar*.

Właściwości tej kontrolki opisano w rozdziałach z sekcji *Powiązane tematy*.

Powiązane tematy:

- Typowe właściwości kontrolek
- Właściwości kontrolek - Dostępność

4.1.6.23. POLE NUMERYCZNE

Podsumowanie:
Kontrolka *Pole numeryczne* służy do wprowadzania wartości liczbowych. Kontrolka ma z prawej strony dwa przyciski oznaczone trójkątami skierowanymi wierzchołkami w górę i w dół. Kliknięcie jednego z tych przycisków odpowiednio zwiększa lub zmniejsza wartość w polu o określoną wartość. Tej kontrolki można używać dla pól typu *int*, *decimal* i *float*.

Oprócz właściwości typowych dla większości kontrolek, pole numeryczne wymaga ustawienia właściwości:

`Wstaw 0 zamiast NULL dla pustej wartości`: po włączeniu powoduje, że kontrolki typu pole numeryczne są inicjowane w nowym rekordzie z wartością *0*; dla nowych kontrolek przełącznik jest domyślnie wyłączony.

`Format`: pozwala określić format wyświetlania liczb w polu numerycznym. Podstawowe zasady formatowania wartości numerycznych podano w poniższej tabeli.

Formaty danych typów numerycznych

Parametr formatu	Opis formatu	Przykład
`n`	Formatuje wartości numeryczne jako liczby dziesiętne z trzema miejscami po przecinku	`n` — 1 234 567,890
		`n2` — 1 234 567,89
`c`	Formatuje wartości numeryczne jako kwoty, zgodnie z ustawieniami narodowymi	`c` — 1 234 567,89 zł
`p`	Formatuje wartości numeryczne jako wartości procentowe, domyślnie bez miejsc po przecinku	`p` — 37%
		`p1` — 37,5%

`Spinner`: włącza lub wyłącza przyciski do zwiększania/zmniejszania wartości liczbowej.

`Wyrównaj do prawej`: powoduje wyrównanie wartości w polu do prawej strony.

`Step`: wartość o jaką zwiększa się lub zmniejsza wartość w polu po kliknięciu jednego z przycisków z prawej strony kontrolki; obsługiwane są tylko wartości całkowite.

Powiązane tematy:

- Typowe właściwości kontrolek
- Właściwości kontrolek - Zależności
- Właściwości kontrolek - Dostępność

4.1.6.24. POLE TYPU MASKA

Podsumowanie:
Kontrolka *Pole typu maska* służy do wprowadzania ciągów znaków w określonym formacie, np. kodów pocztowych. Tej kontrolki można używać dla pól typu *varchar* i *nvarchar*.

Oprócz właściwości typowych dla większości kontrolek, pole typu maska wymaga ustawienia właściwości:

`Maska`: w tym polu należy zdefiniować maskę, to jest wzorzec, z jakim muszą być zgodne wprowadzane w tej kontrolce wartości. Zasady tworzenia maski są następujące:

- `0` — wymaga wpisania cyfry (0–9).

- `9` — wymaga wpisania cyfry lub spacji.
- `#` — wymaga wpisania cyfra, spacji, znaku plus lub minus.
- `L` — wymaga wpisania litery (a-Z).
- `?` — wymaga wpisania litery lub spacji.
- `A` — wymaga wpisania znaku alfanumerycznego (0-9, a-Z).
- `a` — wymaga wpisania znaku alfanumerycznego lub spacji.
- `&` — wymaga wpisania dowolnego znaku (z wyjątkiem spacji).
- `C` — wymaga wpisania znaku lub spacji.
- `\` — znak modyfikacji („ucieczki”), pozwala użyć powyższych znaków w masce jako literałów.

PRZYKŁADY

- `00-000`: kod pocztowy
- `+48 000 000 000`: numer telefonu komórkowego

Powiązane tematy:

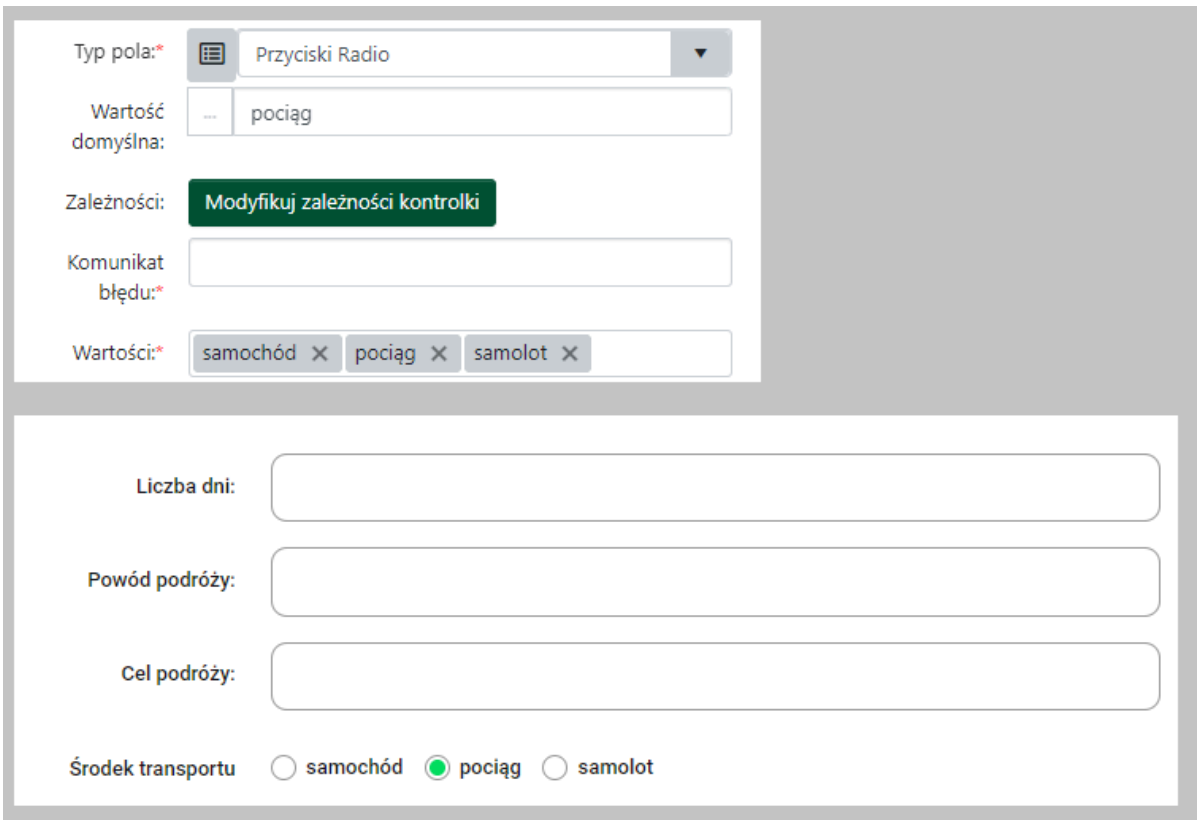
- Typowe właściwości kontrolek
- Właściwości kontrolek - Dostępność

4.1.6.25. PRZYCISKI RADIO

Podsumowanie:
Przyciski radio (przyciski opcji) to grupa kilku przycisków radiowych definiowanych przez podanie ich wartości (ciąg znaków opisujący przycisk opcji). Użytkownik może zaznaczyć tylko jeden przycisk radiowy w grupie. Wartość tego przycisku jest zapisywana w polu formularza. Tej kontrolki używa się do wprowadzania danych do kolumn typu *varchar* i *nvarchar*.

Podstawową właściwością kontrolki jest `Wartość`:

- `Wartość`: wpisz wartość pierwszego przycisku radio i naciśnij Enter; powtórz dla pozostałych przycisków; jeśli wybrany przycisk radio ma być zaznaczony domyślnie, podaj jego wartość we właściwości `Wartość domyślna`. Podana wartość jest jednocześnie etykietą przycisku opcji.



Konfiguracja i podgląd kontrolki przyciski radiobox

Pozostałe właściwości tej kontrolki opisano w rozdziałach z sekcji *Powiązane tematy*.

Powiązane tematy:

- Typowe właściwości kontrolek
- Właściwości kontrolek - Zależności
- Właściwości kontrolek - Dostępność

4.1.6.26. TYPY BIZNESOWE (LISTA WYBORU)

Podsumowanie:
Lista wyboru ze zdefiniowanymi wartościami typów biznesowych dla danego dokumentu biznesowego. Służy do obsługi systemowych kolumn tabeli dokumentu biznesowego `Type1`, `Type2` i `Type3`. Na liście wyświetlane są wartości zdefiniowane dla typu biznesowego jako *Opis*, natomiast w powiązanej kolumnie zapisywany jest identyfikator danego typu (wartość z kolumny *Id* w tabeli *core.BusinessTypes*).

Właściwości tej kontrolki opisano w rozdziałach z sekcji *Powiązane tematy*.

Powiązane tematy:

- Typy dokumentów
- Typowe właściwości kontrolek
- Właściwości kontrolek - Zależności
- Właściwości kontrolek - Dostępność

4.1.6.27. PRZYCISK

Podsumowanie:
Przycisk nie jest powiązany z żadnym polem formularza. Kliknięcie przycisku wyzwala akcję (lub sekwencję akcji) przypisaną do przycisku.

Oprócz właściwości typowych dla większości kontrolek, przycisk wymaga ustawienia właściwości:

`Zachowanie`: określ zachowanie przycisku; dostępne wartości to:

- `Domyślny`: (zachowanie domyślne) jeśli do przycisku jest przypisana akcja SQL zapisu danych lub akcja systemowa `Zapisz rekord`, staje się przyciskiem typu `Zatwierdź formularz`, w przeciwnym razie przyciskiem typu `Zwykły przycisk`.
- `Zatwierdź formularz`: kliknięcie przycisku wywołuje walidację pól formularza (wyświetlane są ewentualne błędy).
- `Zwykły przycisk`: kliknięcie przycisku nie wywołuje walidacji pól formularza.

`Przycisk zawsze dostępny`: (domyślnie wyłączone) po włączeniu powoduje, że przycisk będzie zawsze można kliknąć i wykonać przypisaną do niego akcję, niezależnie od ustawień dostępności (tego przycisku i elementów nadrzędnych) oraz uprawnień ACL formularza. Ta funkcja jest udostępniana z myślą o umożliwieniu zamknięcia formularza wyświetlonego w trybie linku wewnętrznego w sytuacji, kiedy mogą mieć zastosowanie ustawienia uniemożliwiające użycie przycisku `Anuluj`. Użycie jej w innych scenariuszach może skutkować nieoczekiwanymi rezultatami.

`Wyświetlaj`: określa sposób prezentacji przycisku na formularzu; dostępne wartości:

- `Tylko tekst`
- `Tylko ikona`
- `Tekst z ikoną`

`Kolor czcionki i ikony`: wybierz kolor czcionki i ikony z próbnika.

`Kolor wypełnienia`: wybierz kolor wypełnienia z próbnika.

`Grubość ramki`: ustaw grubość obramowania (piksele).

`Kolor ramki`: wybierz kolor obramowania z próbnika.

- `Położenie ikony`: określa położenie ikony względem tekstu; dostępne wartości:
  - `Po lewej od tekstu`
  - `Po prawej od tekstu`
  - `Nad tekstem`
  - `Pod tekstem`
- `Kolor tekstu i ikony dla krycia tła przycisku 0% lub braku tła`: określa kolor tekstu i ikony wyświetlany po najechaniu kursorem myszy na przycisk z przezroczystym tłem; domyślnie jest to `Kolor akcji` ze stylu ustawionego w kreatorze stylów. Do ustawiania kolorów służy selektor, który zawiera następujące elementy:

1. Paleta odcieni, 2. Wyłączenie koloru, 3. Suwak koloru, 4. Suwak krycia, 5. Szesnastkowy kod koloru, 6. Wartości R, G i B koloru, 7. Wartość krycia, 8. Paleta kolorów standardowych.

Selektor koloru

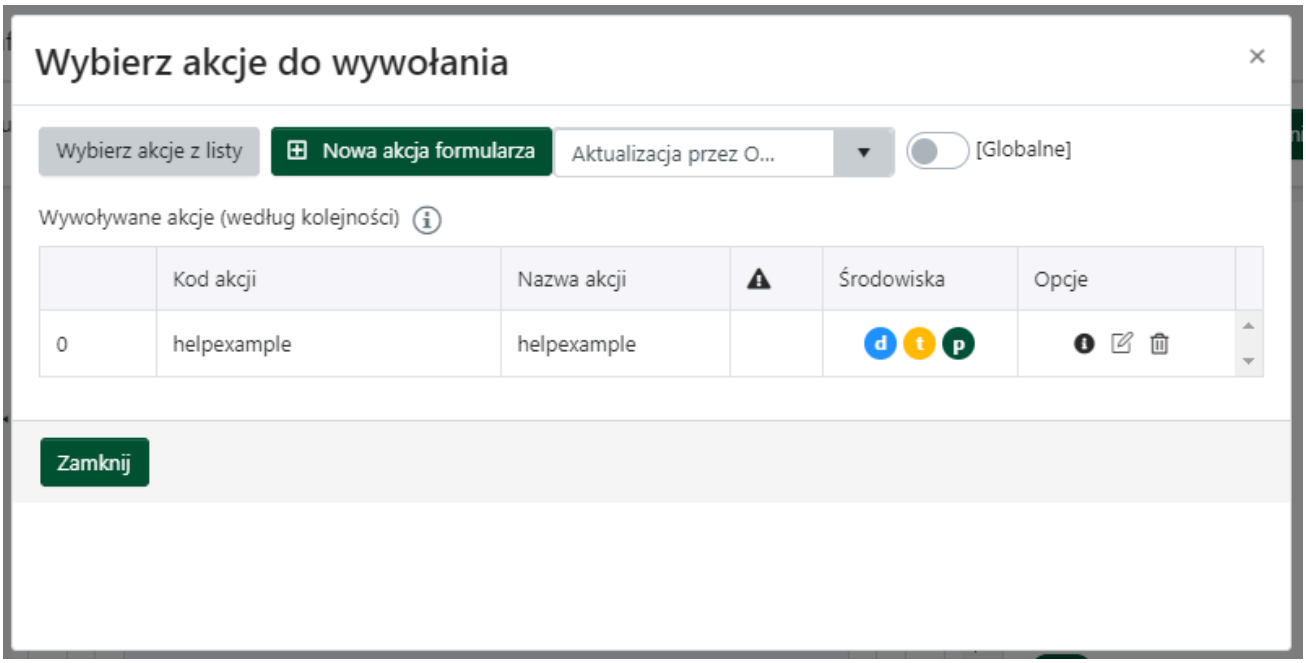
Paleta kolorów standardowych w pierwszych trzech kolumnach zawiera: kolory kreatora stylów (główny, zależne A, B i C, akcji i zależny A), standardowe odcienie szarości oraz kolory z biblioteki *Bootstrap* (btn-success, btn-danger, btn-warning, btn-info, btn-light, btn-dark).

- `Rozmiar tekstu`: określa w procentach rozmiar tekstu względem ustawienia w kreatorze stylów (`INTERFEJS UŻYTKOWNIKA` > `Kreator stylów systemu`); wartość maksymalna to 100%.
- `Rozmiar ikony`: określa w procentach rozmiar ikony względem ustawienia w kreatorze stylów; wartość maksymalna to 100%.
- `Odstęp ikony od tekstu`: określa w pikselach odległość między ikoną i tekstem przycisku.
- `Margines wewnętrzny poziomy`: określa w pikselach *padding* w poziomie (od lewej i od prawej krawędzi przycisku).
- `Margines wewnętrzny pionowy`: określa w pikselach *padding* w pionie (od górnej i od dolnej krawędzi przycisku).

`Nigdy nie grupować`: pozwala wyłączyć dany przycisk z grupowania w *split button*; patrz Grupowanie przycisków w nagłówku formularza

`Skrót klawiszowy`: pozwala przypisać do przycisku kombinację klawiszy, których naciśnięcie będzie równoważne z kliknięciem przycisku; możliwy jest wybór jednej z predefiniowanych kombinacji klawiszy.

`Akcje`: kliknięcie przycisku `Ustaw` wyświetla okno dialogowe `Wybierz akcje do wywołania`, w którym można wybrać jedną lub kilka akcji, które zostaną wykonane po kliknięciu przycisku.



Wybierz akcje do wywołania

Na liście wybranych akcji w wierszu każdej akcji są dostępne przyciski:

- (`Edycja`): wyświetla okno dialogowe edycji akcji; edycja jest niedostępna w przypadku niektórych akcji systemowych.
- (`Usuń`): usuwa akcję z listy.

`Wybierz akcje z listy`: wyświetla okno dialogowe `Lista utworzonych akcji` z listą dostępnych akcji; wyboru dokonuje się, zaznaczając pole wyboru z lewej strony akcji i klikając przycisk `Zapisz`.

`Nowa akcja formularza`: wyświetla okno dialogowe definiowania akcji dla kategorii akcji wybranej na liście rozwijanej z prawej strony przycisku; patrz Akcje w nAxiom ; utworzona akcja jest zapisywana jako akcja lokalna formularza.

`Globalne`: jeśli ten przełącznik jest wyłączony, na liście `Dostępne akcje` są wyświetlane akcje zdefiniowane lokalnie dla formularza; po włączeniu wyświetla akcje zdefiniowane jako globalne w danej witrynie nAxiom.

Powiązane tematy:

- Typowe właściwości kontrolek
- Właściwości kontrolek - Dostępność

4.1.6.28. WYNIK ZAPYTANIA SQL

Podsumowanie:
Kontrolka tylko do odczytu. Nie jest powiązana z żadnym polem formularza. Wyświetla wynik zapytania SQL zdefiniowanego we właściwościach tej kontrolki. To zapytania musi zwrócić pojedynczą wartość (wartość z jednej kolumny dla jednego rekordu).

Wartość wyświetlana w komórce jest wynikiem zapytania skonfigurowanego we właściwości `SQL`. W zapytaniu można używać zmiennych z kontekstu formularza (*{@Nazwa}*) i składni smart numbers.

Pozostałe właściwości tej kontrolki opisano w rozdziałach z sekcji *Powiązane tematy*.

Powiązane tematy:

- Typowe właściwości kontrolek
- Właściwości kontrolek - Dostępność
- Smart numbers

4.1.7. Uprawnienia ACL formularza

Podsumowanie:
Uprawnienia formularza to „przedłużenie" funkcjonalności szablonów uprawnień na poziom formularza. Pozwalają one definiować dostęp do indywidualnych sekcji i pól formularza i w ten sposób określać wartości, które mają być widoczne/„modyfikowalne" dla określonych użytkowników.

W TYM TEMACIE

- Wprowadzenie
- Definiowanie uprawnień
- Malarz uprawnień
- Typ walidacji uprawnień
- Baza danych

Uprawnienia dla formularza można definiować tylko w przypadku włączenia obsługi uprawnień na poziomie definicji dokumentu biznesowego; patrz Uprawnienia ACL do dokumentów .

Uprawnienia dotyczą obiektów na formularzu i są przyznawane określonym użytkownikom. Podobnie jak w przypadku szablonów uprawnień dla dokumentów, osobą uprawnioną może być twórca dokumentu, użytkownicy z określoną rolą lub członek grupy interesariuszy. Ponadto uprawnienia formularza określa się dla indywidualnych statusów, w tym także dla nowego dokumentu (jeszcze bez statusu), natomiast nie ma możliwości zdefiniowania zestawu uprawnień dla formularza obejmującego wszystkie statusy.

Uprawnienia formularza są realizowane za pomocą zmiany atrybutów kontrolek na formularzu, dlatego uprawnienie do odczytu określonej wartości jest równoznaczne z ustawieniem widoczności kontrolki, w której ta wartość jest wyświetlana. Analogicznie uprawnienie do modyfikacji wartości odpowiada ustawieniu atrybutu *tylko do odczytu* dla kontrolki. W uprawnieniach formularza można także określić, że podanie określonej wartości będzie dla danego użytkownika obowiązkowe.

Podobne możliwości dają także **ustawienia dostępności** dla kontrolek i sekcji formularza. Sposób łączenia ze sobą ustawień dostępności i uprawnień formularza przy użyciu odpowiedniego **typu walidacji** opisano w dalszej części.

DEFINIOWANIE UPRAWNIEŃ

Aby zdefiniować uprawnienia na poziomie formularza, przejdź do edycji formularza i kliknij przycisk `Uprawnienia` w prawym górnym rogu okna kreatora.

> **UWAGA**
> Ten przycisk jest dostępny tylko wtedy, kiedy w definicji dokumentu biznesowego tego formularza włączono obsługę uprawnień.

Zostanie wymuszony zapis formularza, a następnie wyświetlone okno `Uprawnienia dla formularza`. Uprawnienia są wyświetlane dla wybranych sekcji formularza. Domyślnie żadna sekcja nie jest zaznaczona. Aby wyświetlić uprawnienia, należy klikać przyciski z kodem sekcji u góry okna.

Uprawnienia są wyświetlane w formie tabeli, której wiersze są grupowane według osób lub statusów, zależnie od opcji wybranej w grupie `Tryb wyświetlania` (`wg statusów` albo `wg uprawnionych`), a kolumny określają sekcje i pola formularza. Pierwsza kolumna (od lewej) zawiera wykaz szablonów pogrupowanych według tytułu uprawnienia lub według statusu. Druga kolumna zawiera pięć pól wyboru, którymi można zbiorczo sterować stanem poszczególnych uprawnienia dla wszystkich elementów formularza.

Kliknij zielony przycisk `Nowy uprawniony`, aby zdefiniować nowy szablon uprawnień.

- `Nazwa statusu`: wybierz status dokumentu, w którym mają obowiązywać uprawnienia; pozycja `Nowy dokument` pozwala zdefiniować uprawnienia dla nowego dokumentu, który jeszcze nie został zapisany, w takim przypadku mogą to być wyłącznie uprawnienia dla twórcy dokumentu.
- `Uprawnienia dla twórcy dokumentu`: zdefiniowany szablon będzie obowiązywał dla użytkownika, który utworzył dany dokument (jego identyfikator jest zapisany w kolumnie *[UserId]* w tabeli modelu danych).
- `Role globalne`: wybierz jedną lub kilka ról, dla których ma obowiązywać definiowany szablon; w przypadku wybrania kilku ról szablon będzie obowiązywał dla użytkowników, którzy maja przypisaną dowolną z wybranych ról.
- `Interesariusz`: zaznacz tę opcję, aby tworzony szablon uprawnień obowiązywał dla członków grupy interesariuszy wybranej z listy obok.
- `Zamknij`: zamyka okno dialogowe bez zapisywania zmian.
- `Zapisz`: zapisuje zmiany i zamyka okno dialogowe.

Po zdefiniowaniu szablonu zaznaczaj/anuluj zaznaczenie pól wyboru w kolumnach odpowiadających sekcjom i polom formularza. W nagłówku z nazwą danego elementu formularza znajdują się pola wyboru, którymi można ustawiać odpowiednie uprawnienia zbiorczo dla wszystkich uprawnionych.

> **WSKAZÓWKA**
> Sekcja *toolbar* jest reprezentowana jako dwie podsekcje przycisków akcji.

Możesz ustawić następujące uprawnienia:

- `Pole widoczne`: decyduje o tym, czy sekcja/pole będzie wyświetlane.
- `Pole edytowalne`: decyduje o tym, czy możliwa będzie edycja danych w sekcji/polu.
- `Pole wymagane`: decyduje o tym, czy wpisanie wartości w polu będzie wymagane.
- `Możliwość dodawania`: występuje w sekcji załączników i dla listy wyboru interesariuszy; decyduje o tym, czy użytkownik może dodawać załączniki i/lub interesariuszy.
- `Możliwość usuwania`: występuje w sekcji załączników i dla listy wyboru interesariuszy; decyduje o tym, czy użytkownik może usuwać załączniki i/lub interesariuszy.

Zmiany ustawień mogą być zapisywane automatycznie po kliknięciu dowolnego przycisku sekcji, o ile przełącznik `Autozapis po zmianie sekcji` nad tabelą z prawej strony jest włączony (ustawienie domyślne). Niezapisane zmiany ustawień w poszczególnych sekcjach są sygnalizowane przez gwiazdkę widoczną w przycisku sekcji. Próba zamknięcia okna dialogowego z niezapisanymi zmianami spowoduje wyświetlenie ostrzeżenia. Jeśli opcja autozapisu jest wyłączona, zmiany są zapisywane dopiero po kliknięciu przycisku `Zapisz` lub `Zapisz i zamknij`.

> **UWAGA**
> Wartości uprawnień formularza dla konkretnej instancji dokumentu są generowane w momencie utworzenia tej instancji lub przy zmianie statusu i zapisywane w pamięci podręcznej. Podczas otwierania dokumentu w formularzu wpisy są pobierane z cache, a jeśli nie zostaną znalezione (np. cache zostanie wyczyszczony po restarcie API) to wpisy są ponownie generowane na podstawie szablonów uprawnień dla formularza.

MALARZ UPRAWNIEŃ

Aby ułatwić ustawianie uprawnień, można skorzystać z funkcjonalności „malarza uprawnień”, która pozwala powielić uprawnienia jednego elementu formularza na inny lub jednego uprawnionego na innego.

1. Kliknij ikonę pipety , aby włączyć tryb malarza.
2. Zaznacz rekord lub kolumnę z uprawnieniami, które chcesz skopiować (klikając nazwę kolumny lub rekordu).
3. Kliknij ikonę trybu malowania , aby skopiować wszystkie uprawnienia.
4. Zaznacz rekord lub kolumnę, do której chcesz skopiować uprawnienia.
5. Kliknij ikonę zatwierdzenia .

Uprawnienia zostaną ustawione w docelowym rekordzie/kolumnie i tryb malarza uprawnień zostanie wyłączony.

TYP WALIDACJI UPRAWNIEŃ

Warunki widoczności, możliwości edycji oraz wymagalności dla pól i sekcji formularza można w nAxiom definiować także w oknie `Ustawienia dostępności pola/sekcji` (panel właściwości pola sekcji > `Dostępność` > `Ustaw`). Są one przydatne w przypadku dokumentów, dla których nie używa się uprawnień ACL. Te ustawienia pozwalają definiować warunki widoczności i możliwości edycji za pomocą SQL co powoduje, że mogą być przydatne w przypadku korzystania z ACL.

Jeśli jednak ten sam atrybut kontrolki można ustawić w dwóch miejscach jednocześnie, konieczne jest określenie, które z tych ustawień ma być obowiązujące, albo jaki ma być końcowy efekt, zależnie od ich kombinacji. Służy do tego lista rozwijana `Typ walidacji` dostępna w oknie `Uprawnienia dla formularza`. Zawiera ona następujące pozycje:

- `Wg ustawień dostępności pól i sekcji`: obowiązują tylko ustawienia dostępności, uprawnienia dla formularza są ignorowane.
- `Wg uprawnień dla formularza`: obowiązują tylko uprawnienia dla formularza, ustawienia dostępności są ignorowane.
- `Rozszerzaj uprawnienia dla formularza ustawieniami dostępności`: wynikowa wartość atrybutu jest logiczną sumą wartości określonych uprawnieniami dla formularza i ustawieniami dostępności.
- `Zawężaj uprawnienia dla formularza ustawieniami dostępności`: wynikowa wartość atrybutu jest logicznym iloczynem wartości określonych uprawnieniami dla formularza i ustawieniami dostępności.

Zasady wyliczania wartości ilustruje poniższa tabela.

Wyliczanie warunków dla kontrolek i sekcji formularza			
Typ walidacji	**Ustawienia dostępności**	**Uprawnienia dla formularza**	**Wartość wynikowa**
Wg ustawień dostępności pól i sekcji	✔	⚪	✔
	✘	⚪	✘
Wg uprawnień dla formularza	⚪	✔	✔
	⚪	✘	✘
Rozszerzaj uprawnienia dla formularza ustawieniami dostępności	✔	✔	✔
	✘	✔	✔
	✔	✘	✔
	✘	✘	✘
Zawężaj uprawnienia dla formularza ustawieniami dostępności	✔	✔	✔
	✘	✔	✘
	✔	✘	✘
	✘	✘	✘

✔ = ON, ✘ = OFF, ⚪ = ON/OFF

UWAGI

- Brak zdefiniowanych uprawnień dla formularza oznacza "pełne uprawnienia". W takim przypadku można również określić sposób wyliczania wynikowych ustawień dla kontrolek i sekcji za pomocą listy `Typ walidacji`.
- Należy zwrócić uwagę, że ustawienia sterujące trybem tylko do odczytu są różnie definiowane:
  - uprawnienia dla formularza: `Pole edytowalne` - włączone oznacza możliwość edycji,
  - ustawienia dostępności: `Sekcja/Pole tylko do odczytu` (*isReadOnly*) - włączone oznacza brak możliwości edycji.
- Uprawnienia do dodawania i usuwania załączników nie zależą od wartości wybranej z listy `Typ walidacji`.

Przykład

Jeśli dane pole (sekcja) ma być widoczne tylko w razie spełnienia określonego warunku (innego niż status/rola), należy:

- wyłączyć jego widoczność w uprawnieniach formularza: `Pole widoczne` = ❌
- zdefiniować zapytanie, które zwróci *isVisible* = 1, gdy ten warunek zostanie spełniony:

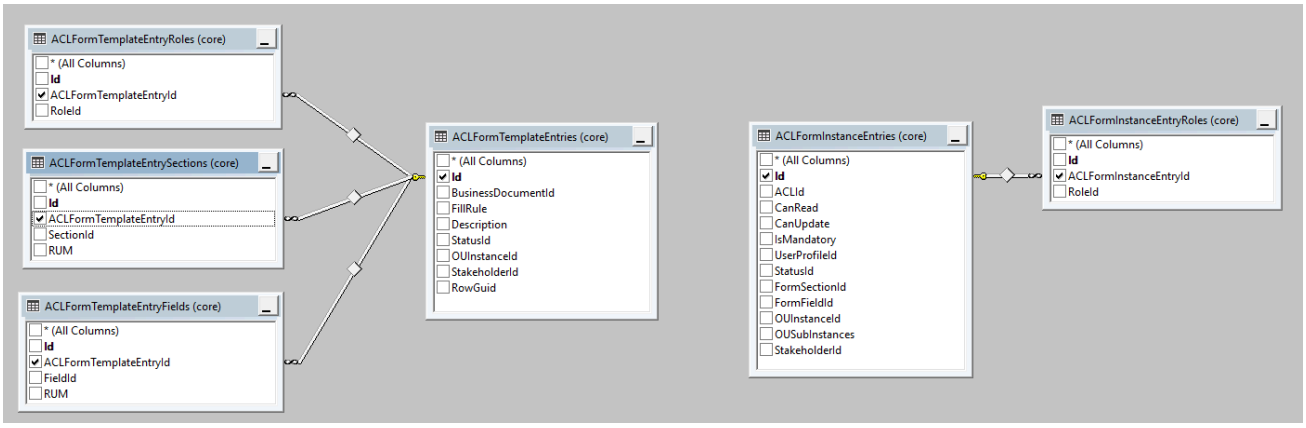
```
SELECT
 CAST(
  CASE
   WHEN <<warunek>> THEN 1
   ELSE 0
  END
 AS BIT) AS isVisible,
 CAST(1 AS BIT) AS isReadOnly;
```

- wybrać typ walidacji `Rozszerzaj...`

Taki sam efekt da włączenie widoczności pola w uprawnieniach formularza i wybranie typu walidacji `Zawężaj...`.

BAZA DANYCH

Informacje o uprawnieniach formularza są zapisywane w kilku tabelach systemowych, które przedstawiono na ilustracji. Poniżej zamieszczono krótki opis tych tabel.



Schemat tabel systemowych z informacjami o uprawnieniach formularzy

[core].[ACLFormTemplateEntries]

Ogólne informacje o zdefiniowanych zestawach uprawnień formularzy

Nazwa kolumny	Typ danych	Zezwól NULL	Opis
Id	int	☐	Kolumna klucza głównego, identyfikator rekordu w tabeli
BusinessDocumentId	int	☐	Identyfikator dokumentu biznesowego
FillRule	int	☐	Identyfikator uprawnionego, patrz przypis poniżej
Description	nvarchar(MAX)	☑	Opis zestawu uprawnień, kolumna nieużywana w przypadku uprawnień formularzy
StatusId	int	☑	Identyfikator statusu, dla którego zdefiniowano zestaw uprawnień
OUInstanceId	int	☑	(obecnie nieobsługiwane) Identyfikator jednostki organizacyjnej, w której obowiązuje zestaw uprawnień
StakeholderId	int	☑	Identyfikator grupy interesariuszy (*[core].[Stakeholders]*), dla której zdefiniowano zestaw uprawnień
RowGuid	uniqueidentifier	☐	Unikatowy identyfikator zestawu uprawnień

Kolumna *[FillRule]* może zawierać następujące wartości:

- 0: (na razie nieobsługiwane)
- 1: twórca dokumentu
- 2: właściciel dokumentu (obecnie nieobsługiwane)
- 3: przełożony twórcy (obecnie nieobsługiwane)
- 4: przełożony właściciela (obecnie nieobsługiwane)
- 5: członek grupy interesariuszy
- 6: użytkownicy z wskazaną rolą
- 7: użytkownicy z wskazaną rolą z wskazanej jednostki organizacyjnej (obecnie nieobsługiwane)
- 8: użytkownicy z wskazaną rolą z wskazanej jednostki organizacyjnej i jej jednostkę podrzędnych (obecnie nieobsługiwane)

[core].[ACLFormTemplateEntryRoles]

Informacje o rolach powiązanych z poszczególnymi zestawami uprawnień formularzy

Nazwa kolumny	Typ danych	Zezwól NULL	Opis
Id	int	☐	Kolumna klucza głównego, identyfikator rekordu w tabeli
ACLFormTemplateEntryId	int	☐	Identyfikator zestawu uprawnień (kolumna *[Id]* z tabeli *[core].[ACLFormTemplateEntries]*)
RoleId	int	☐	Identyfikator roli, dla której obowiązuje schemat uprawnień

[core].[ACLFormTemplateEntrySections]

Szczegółowe uprawnienia dla sekcji formularza

Nazwa kolumny	Typ danych	Zezwól NULL	Opis
Id	int	☐	Kolumna klucza głównego, identyfikator rekordu w tabeli
ACLFormTemplateEntryId	int	☐	Identyfikator zestawu uprawnień (kolumna *[Id]* z tabeli *[core].[ACLFormTemplateEntries]*)
SectionId	int	☐	Identyfikator sekcji formularza, której dotyczą uprawnienia
RUM	nvarchar(MAX)	☑	Symbol przyznanych uprawnień (R - read/odczyt, U - update/modyfikacja, M - mandatory/wymagalność)

[core].[ACLFormTemplateEntryFields]

Szczegółowe uprawnienia dla pól formularza

Nazwa kolumny	Typ danych	Zezwól NULL	Opis
Id	int	☐	Kolumna klucza głównego, identyfikator rekordu w tabeli
ACLFormTemplateEntryId	int	☐	Identyfikator zestawu uprawnień (kolumna *[Id]* z tabeli *[core].[ACLFormTemplateEntries]*)
FieldId	int	☐	Identyfikator pola na formularzu, którego dotyczą uprawnienia
RUM	nvarchar(MAX)	☑	Symbol przyznanych uprawnień (R - read/odczyt, U - update/modyfikacja, M - mandatory/wymagalność)

UWAGA
Schematy uprawnień formularza są przypisywane do definicji dokumentu; jeśli z daną definicją będzie używanych kilka formularzy, będą one korzystać z tego samego schematu uprawnień.

Powiązane tematy:

- Ustawienia dostępności pola/sekcji

4.2. FORMULARZE PUBLICZNE

Podsumowanie:
Formularze publiczne to formularze, za pomocą których niezalogowany użytkownik może tworzyć instancje dokumentów w nAxiom oraz dodawać do nich załączniki. Strona formularza jest renderowana na podstawie kodu HTML, CSS i Javascript podawanego w definicji formularza. Ponadto można zdefiniować strony wyświetlane w przypadku powodzenia i niepowodzenia zapisu.

Formularz publiczny jest dostępny pod adresem URL *<adres_witryny_naxiom>/front/public/<adres_względny>*. Wywołanie formularza jest chronione funkcją *rate limiting*, a do weryfikacji, czy dane są wysyłane przez człowieka można użyć mechanizmu Google reCAPTCHA v3 .

☛ *AdminSPA* > `INTERFEJS UŻYTKOWNIKA` > `Formularze publiczne`

Oprócz metadanych standardowych dla obiektów w nAxiom dla formularza publicznego należy zdefiniować następujące właściwości:

- Karta `Ogólne`:
  - `Język formularza`: określa język komunikatów o błędach walidacji wykonywanej po stronie serwera oraz wersję językową etykiet pól wyświetlanych w tych komunikatach.
  - `Dokument biznesowy`: w tabeli powiązanej z podaną definicją będą zapisywane dane z formularza publicznego, a w kategorii załączników — przesłane pliki.

    WAŻNE
    Dane z kontrolek formularza zostaną zapisane w polach modelu danych pod warunkiem, że w znacznikach kontrolek formularza zostanie użyty atrybut *name* o wartości takiej, jak nazwa odpowiedniego pola danych, np:

    ```
    <input type="text" class="form-control" name="FirstName" placeholder="Wprowadź imię" requ
    ```

  - `Adres względny...`: ciąg tekstowy używany w adresie URL formularza publicznego; nie może się powtarzać dla danego tenanta.
- Karta `Ekran formularza`: obejmuje trzy sekcje:
  - `HTML`: okno edytora Monaco, w którym można wpisać/wkleić kod HTML formularza.
  - `CSS`: okno edytora Monaco, w którym można wpisać/wkleić arkusz stylów dla formularza.
  - `Javascript`: okno edytora Monaco, w którym można wpisać/wkleić kod Javascript dla formularza.
- Karty `Ekran odpowiedzi serwera - powodzenie/niepowodzenie`: pozwalają zdefiniować komunikat o powodzeniu/niepowodzeniu zapisu danych z formularza publicznego; obie karty obejmują sekcje `HTML` i `CSS`, które zawierają wstępnie zdefiniowane strony komunikatów.

WSKAZÓWKA
W definicji formularza obsługiwane są style Bootstrap oraz biblioteki jQuery.js, Moment.js i DateRangePicker.js (https://www.daterangepicker.com/#examples).

- Panel `Pomoc`: zawiera dodatkowe informacje o nazwach pól modelu danych oraz

kategorii załączników. Dodatkowo u dołu znajdują się dwa przykładowe formularze w tym jeden z walidacją pól formularza realizowaną przez Javascript oraz z komponentem dodawania załączników (obsługuje metodę drag&drop i okno wyboru plików).

GOOGLE RECAPTCHA V3

Dla formularzy publicznych można włączyć mechanizm Google reCAPTCHA v3, który zweryfikuje, czy dane są wysyłane przez człowieka. Odpowiednie ustawienia znajdują się w TenantsAdmin SPA (*Global settings* > *Google reCAPTCHA v3*). Mechanizm reCAPTCHA konfiguruje się dla witryny nAxiom, to jest dla wszystkich tenantów. Wymagane jest zarejestrowanie domeny witryny w usłudze Google reCAPTCHA i wybranie wersji *v3*. Rejestracja obejmuje domenę i wszystkie domeny podrzędne. Możliwe jest jednorazowe zarejestrowanie wielu domen dla różnych witryn nAxiom.

Wynikiem rejestracji jest przydzielenie klucza publicznego witryny (*SiteKey*) oraz klucza tajnego (*SecretKey*). Oba parametry należy podać w TenantsAdmin SPA, gdzie również można włączyć lub wyłączyć ochronę. Dodatkowo należy skonfigurować parametr *Acceptable scoring*, to jest minimalną wartość oceny zwracaną przez mechanizm reCAPTCHA, dla której weryfikacja będzie uznawana za pomyślną (zwracane wartości należą do zakresu od 0 do 1). W przypadku oceny niższej od podanej wartości, dane z formularza nie zostaną zapisane i nastąpi przekierowanie na stronę komunikatu o błędzie. Dodatkowo zostanie zalogowana informacja o niepowodzeniu weryfikacji.

Włączenie mechanizmu reCAPTCHA spowoduje dodanie do kodu HTML formularza publicznego nagłówka:

```
<script src="<https://www.google.com/recaptcha/api.js?render=klucz_publiczny_witryny>"></script>
```

Kiedy użytkownik kliknie przycisk wysłania danych formularza, skrypt wyśle żądanie do API reCAPTCHA w celu uzyskania tokenu. Następnie token zostanie przekazany wraz z danymi formularza do punktu końcowego API nAxiom. Ten punkt przed zapisaniem danych wyśle żądanie weryfikacji, podając w nim przekazany token i klucz tajny. Odpowiedź w formacie JSON wygląda następująco:

```
{
"success": true|false,      // określa, czy przesłany token reCAPTCHA dla danej witryny był prawidłowy
"score": number             // wynik oceny żądania (0.0 - 1.0)
"action": string            // nazwa akcji dla tego żądania (na potrzeby weryfikacji)
"challenge_ts": timestamp,  // datownik wczytania wyzwania (format ISO yyyy-MM-dd'T'HH:mm:ssZZ)
"hostname": string,         // nazwa hosta witryny, dla której wykonano weryfikację reCAPTCHA
"error-codes": [...]        // opcjonalne kody błędów
```

```
}
```

Ponadto skrypt generuje logo mechanizmu reCAPTCHA w prawym dolnym rogu formularza.

UWAGI

- Bezpłatny limit dla usługi Google reCAPTCHA wynosi 10 tysięcy żądań na miesiąc dla domen obsługiwanych przez parę *SiteKey/SecretKey*. Po przekroczeniu limitu usługa może stale zwracać wynik weryfikacji równy 0,9.
- Korzystanie z Google reCAPTCHA wymaga otwarcia połączeń do domen:
  - http://www.google.com
  - www.gstatic.com
  - fonts.gstatic.com

4.3. LISTY

Podsumowanie:
Listy, nazywane także *widokami listy* to, oprócz formularzy, podstawowy sposób wyświetlania danych w aplikacjach nAxiom. Dane są wyświetlane na listach w układzie tabelarycznym — każdy wiersz odpowiada jednemu rekordowi danych z tabeli źródłowej, w przypadku tabel dokumentu biznesowego każdy rekord danych to konkretny dokument biznesowy. nAxiom w procesie automatycznego generowania modelu procesu dla tabel dokumentu biznesowego tworzy listy, które można potem dostosowywać do potrzeb. Możliwe jest także samodzielne tworzenie list.

☛ *AdminSPA* > `INTERFEJS UŻYTKOWNIKA` > `Listy`

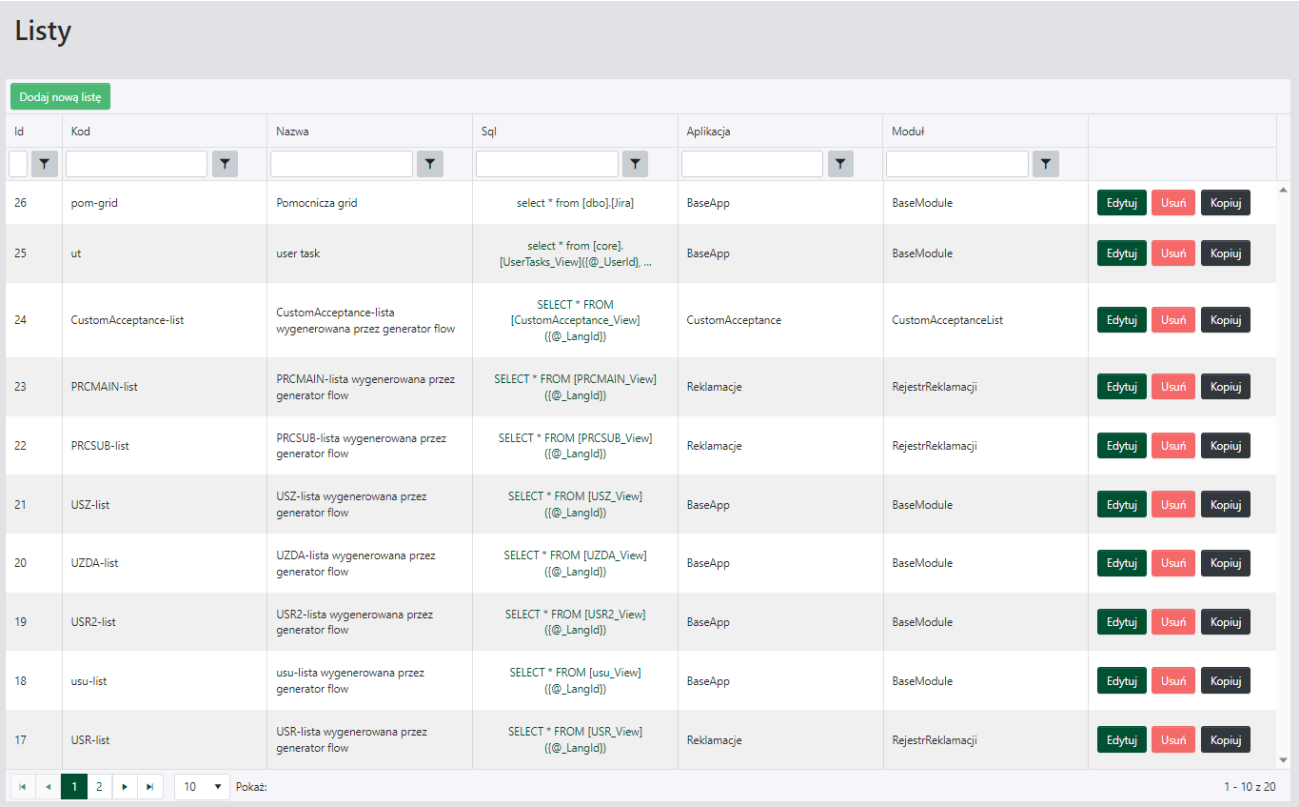
Lista to element interfejsu użytkownika służący do zbiorczego wyświetlania danych w układzie tabelarycznym. Źródłem danych dla listy jest zapytanie SQL operujące na widoku bazy danych lub tabeli (tabelach) bazy danych. Kolumny listy odpowiadają kolumnom widoku lub tabeli, a wiersze rekordom danych. Listy w nAxiom oferują także pewne dodatkowe funkcjonalności. Są to między innymi przycisk tworzenia nowego rekordu, tzw. *przyciski akcji*, to jest przyciski usuwania i edycji, które mogą być wyświetlane w każdym wierszu listy, a także mechanizmy filtrowania i sortowania danych na liście. Przykładowy widok listy opisano w sekcji Widok listy .

W trybie projektowania dostępne są rozbudowane możliwości konfigurowania wyglądu i sposobu działania listy, w tym dodawania przycisków w nagłówku listy, ukrywania kolumn, zmiany kolejności kolumn, wysokości listy na stronie, stronicowania, zaznaczania wierszy, wirtualizacji kolumn, włączania *edycji inline*, czyli edycji danych bezpośrednio na liście,

automatycznego odświeżania i rozmiaru czcionki, koloru tła i czcionki, wyróżnienia i wiele innych.

Ponadto, listy można osadzać na formularzach jako *sekcje typu lista*, patrz Formularze .

Do zarządzania listami służy polecenie `Listy` w sekcji `INTERFEJS UŻYTKOWNIKA` w menu głównym, które wyświetla okno `Listy`. W tym oknie, podobnie jak w wielu innych oknach w aplikacji *AdminSPA*, wyświetlana jest lista, w tym przypadku lista zdefiniowanych list. W tym oknie można tworzyć nowe listy, a także przechodzić do edycji zdefiniowanych list, usuwać je i kopiować.



Okno `Listy`

> **WSKAZÓWKA**
> W celu skopiowania listy, należy kliknąć przycisk `Kopiuj` w wierszu tej listy, a następnie podać kod nowej listy. Skopiowana lista ma taką samą nazwę jak lista oryginalna.

W kolejnych tematach opisano elementy widoku listy, okno `Kreator listy`, właściwości listy, kolumn i przycisków na liście oraz „specjalne” typy kolumn.

Jak wspomniano wcześniej, listy są zawsze tworzone w procesie generowania modelu procesu dla tabeli dokumentu biznesowego, dlatego opcje dostępne w oknie `Kreator listy` zostaną opisane przy założeniu, że czytelnik ma już utworzoną co najmniej jedną listę w swojej instalacji platformy nAxiom.

Powiązane tematy:

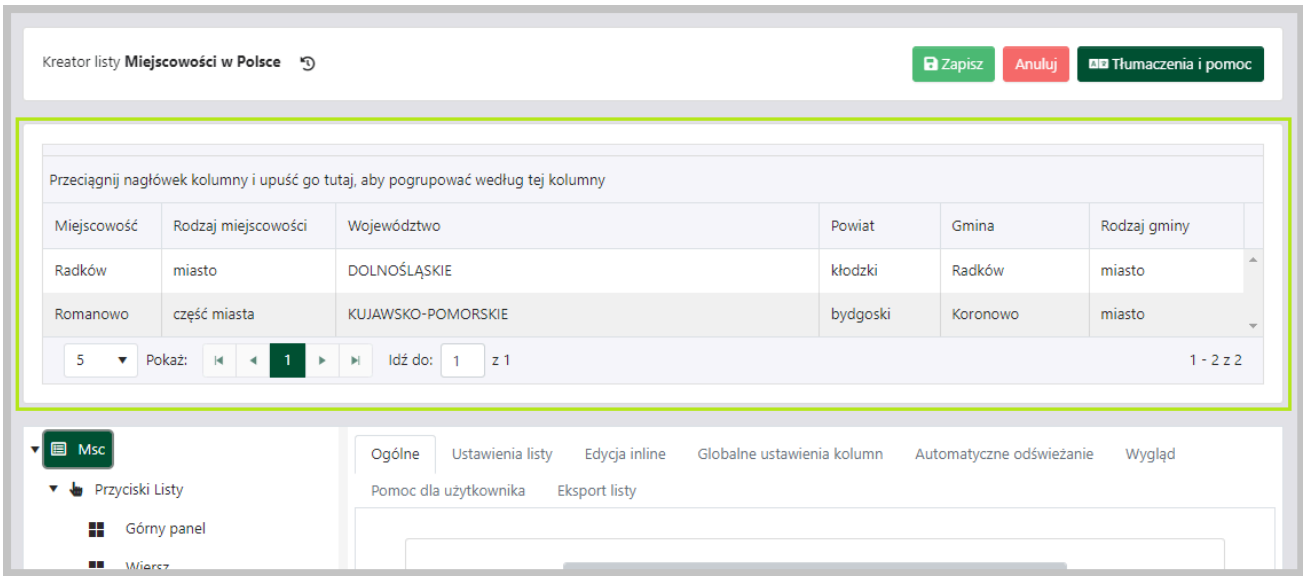
- Widok listy w aplikacji FrontSPA
- Formularze, sekcja typu lista

4.3.1. Okno Kreator listy

Podsumowanie:
Okno `Kreator listy` składa się z kilku głównych paneli: panelu nagłówka z nazwą listy i przyciskami `Zapisz`, `Anuluj` i `Tłumaczenia i pomoc`, panelu podglądu listy u góry, panelu elementów listy w widoku drzewa z lewej strony w dolnej części oraz panelu ustawień właściwych dla elementu wybranego na panelu elementów.

☛ *AdminSPA* > `INTERFEJS UŻYTKOWNIKA` > `Listy` > `Edytuj`

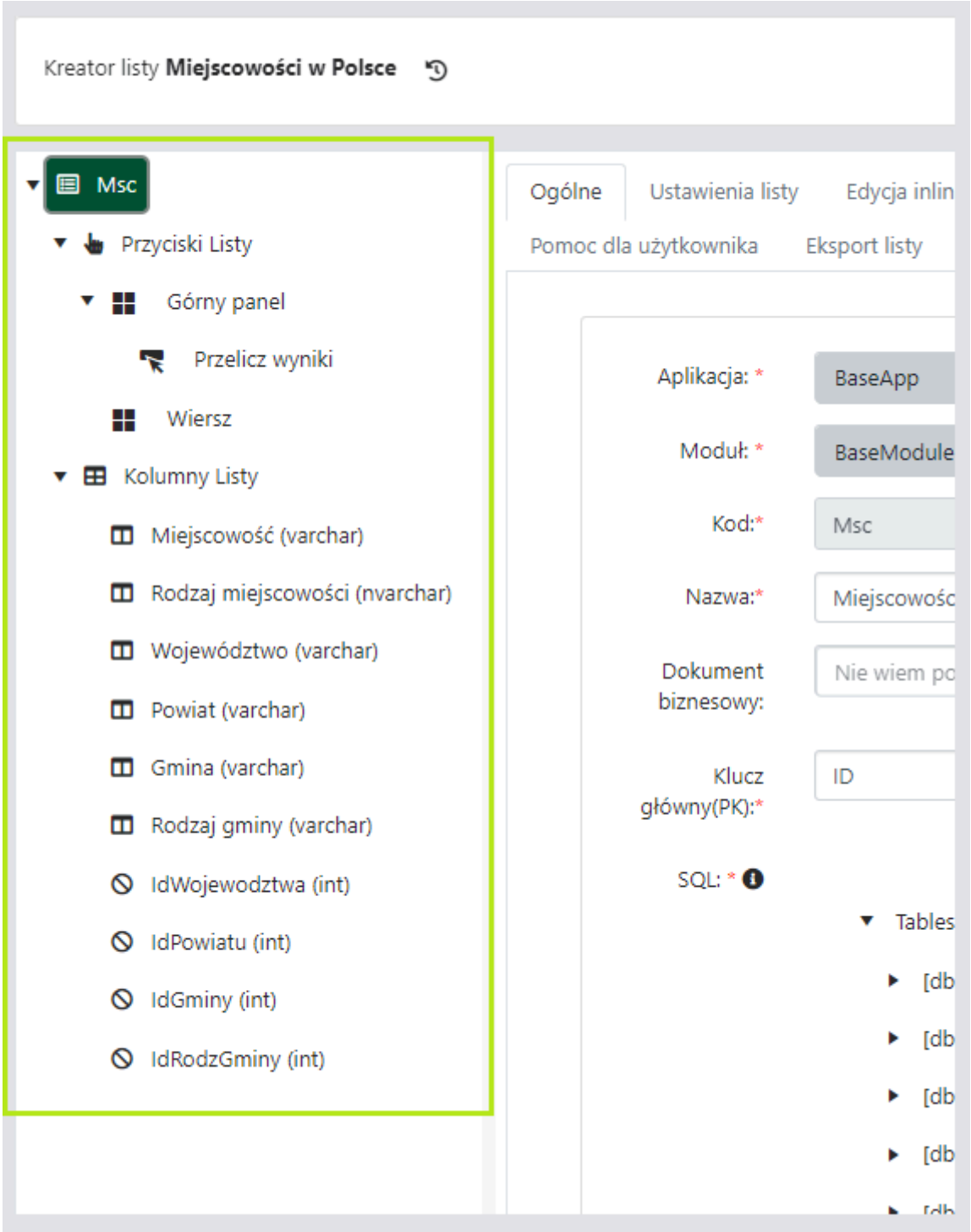
Na panelu podglądu wyświetlane są dwa przykładowe rekordy danych. Podgląd pozwala na bieżąco obserwować zmiany wprowadzane w projekcie listy.



Okno `Kreator listy`, panel podglądu listy.

Poniżej panelu z lewej strony znajduje się panel elementów składających się na strukturę widoku listy, której elementem głównym jest sama lista. Lista na panelu elementów zawiera dwie główne pozycje podrzędne: `Przyciski listy` i `Kolumny listy`. Pierwsza z nich rozwija się na dwie pozycje podrzędne odpowiadające sekcjom listy, w których mogą znajdować się przyciski. Są to `Górny panel` oraz `Wiersz`. Domyślnie na panelu górnym znajduje się przycisk tworzenia nowego rekordu (dokumentu), natomiast w każdym wierszu listy można umieścić przyciski dotyczące operacji na danych z tego wiersza, np. edycji, usunięciu całego wiersza itp.

Pozycja `Kolumny listy` zawiera domyślnie wszystkie kolumny ze źródła danych listy (tabeli lub funkcji tabelarycznej). Dodatkowo lista może zawierać kolumnę specjalną `command`, w której wyświetlane są przyciski zdefiniowane dla sekcji `Wiersz`, oraz inne kolumny specjalne.



Okno `Kreator listy`, panel elementów listy.

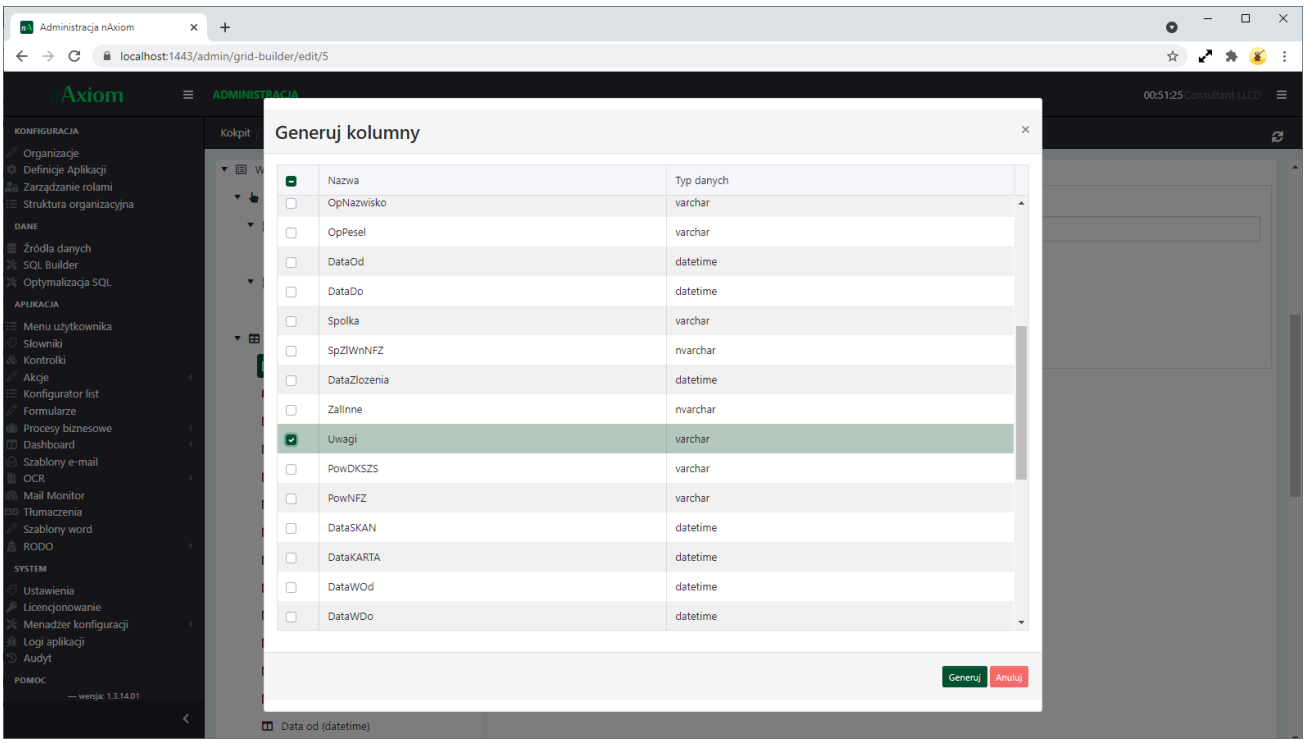
Kliknięcie prawym przyciskiem myszy pozycji na liście powoduje wyświetlenie menu kontekstowego z poleceniami, które opisano w tabeli poniżej.

Pozycja na liście elementów	Polecenie menu kontekstowego	Działanie
`Przyciski listy`, `Górny panel`, `Wiersz`	`Usuń wszystkie przyciski`	Usuwa wszystkie przyciski z listy
`Górny panel`, `Wiersz`	`Dodaj nowy przycisk`	Dodaje nowy przycisk w bieżącej sekcji listy
Przycisk w sekcji `Górny panel` lub `Wiersz`	`Przenieś do góry`	Przenosi dany przycisk o jedną pozycję do góry na panelu elementów, w widoku listy odpowiada to przeniesieniu przycisku o jedną pozycję w lewo
	`Przenieś w dół`	Przenosi dany przycisk o jedną pozycję w dół na panelu elementów, w widoku listy odpowiada to przeniesieniu przycisku o jedną pozycję w prawo
	`Usuń element`	Usuwa dany przycisk z bieżącej sekcji
`Kolumny listy`	`Generuj kolumny`	Otwiera okno dialogowe `Generuj kolumny`, w którym można zaznaczyć dodatkowe kolumny w celu wyświetlenia na liście (na przykład po dodaniu kolumny do źródła danych listy lub po uprzednim usunięciu kolumny z listy); patrz opis poniżej
	`Dodaj kolumnę checkbox`	Dodaje do listy kolumnę z polami wyboru, które pozwalają zaznaczać wiersze na liście; dla tej kolumny można określić właściwość `Zaznacz wszystkie - wszystkie rekordy na stronach` (we właściwościach ogólnych). Decyduje ona, czy zaznaczenie pola wyboru w nagłówku kolumny spowoduje zaznaczenie pól wyboru we wszystkich rekordach listy, czy tylko na bieżącej stronie.
	`Dodaj kolumnę akcji`	Dodaje do listy kolumnę o nazwie `command`, w której są wyświetlane przyciski zdefiniowane w sekcji `Wiersz`

Pozycja na liście elementów	Polecenie menu kontekstowego	Działanie
	`Dodaj kolumnę HTML`	Dodaje do listy kolumnę o nazwie `HTML`, w której można wyświetlać wyrenderowany kod HTML
	`Dodaj kolumnę terminu realizacji`	Dodaje do listy kolumnę o nazwie `Termin`, w której jest wyświetlany termin realizacji bieżącego kroku przetwarzania dokumentu
	`Dodaj kolumnę właściciela zadania`	Dodaje do listy kolumnę o nazwie `Właściciel zadania`, w której są wyświetlane informacje o użytkowniku, który został do dokumentu przypisany jako właściciel zadania
	`Usuń wszystkie kolumny`	Usuwa wszystkie kolumny z listy; aby następnie dodać wybrane kolumny, użyj polecenia `Generuj kolumny`
Kolumna w sekcji `Kolumny listy`	`Przenieś do góry`	Przenosi daną kolumnę o jedną pozycję do góry na panelu elementów, w widoku listy odpowiada to przeniesieniu kolumny o jedną pozycję w lewo
	`Przenieś w dół`	Przenosi daną kolumnę o jedną pozycję w dół na panelu elementów, w widoku listy odpowiada to przeniesieniu kolumny o jedną pozycję w prawo
	`Usuń element`	Usuwa daną kolumnę z listy

GENEROWANIE KOLUMN LISTY

Domyślnie generowana lista zawiera wszystkie kolumny zdefiniowane w źródle danych. Modyfikując listę, można te kolumny usuwać i dodawać. W celu dodania kolumny, której nie ma na liście (na przykład została wcześniej usunięta lub została dodana w definicji źródła danych), kliknij prawym przyciskiem myszy pozycję `Kolumny listy` na panelu elementów i w menu kontekstowym kliknij polecenie `Generuj kolumny`. Zostanie wyświetlone okno `Generuj kolumny`.



Okno dialogowe `Generuj kolumny`.

W tym oknie jest wyświetlana lista wszystkich kolumn zdefiniowanych w źródle danych. Aby dodać kolumnę do listy, zaznacz pole wyboru obok nazwy tej kolumny i kliknij przycisk `Generuj`. Zaznaczona kolumna zostanie dodana do listy.

> **WSKAZÓWKA**
> W jednym kroku można dodać do listy kilka kolumn, a nawet wszystkie, jeśli klikniesz pole wyboru w nagłówku listy kolumn.

W kolejnych tematach opisano właściwości samej listy, kolumn, niektórych kolumn specjalnych i przycisków.

4.3.2. Właściwości listy — karta Ogólne

> **Podsumowanie:**
> Po otwarciu okna `Kreator listy` na panelu elementów domyślnie zaznaczona jest pozycja najwyższego poziomu, to jest lista, a na panelu ustawień wyświetlane są ustawienia listy na karcie `Ogólne`.

> ☛ *AdminSPA* > `INTERFEJS UŻYTKOWNIKA` > `Listy` > `Edytuj` > `Ogólne`

Na tej karcie można ustawić następujące właściwości listy:

Metryka

- `Aplikacja`: aplikacja, z którą jest powiązana lista; bieżąca wartość została ustawiona w procesie generowania modelu procesu biznesowego; patrz Kreator modeli danych, generowanie elementów aplikacji .
- `Moduł`: moduł aplikacji, do którego jest przypisana lista; bieżąca wartość została ustawiona w procesie generowania modelu procesu biznesowego razem z wartością w polu `Aplikacja`.
- `Kod`: (pole tylko do odczytu) wpisywany podczas tworzenia listy ciąg znaków, który identyfikuje listę.
- `Nazwa`: opisowa nazwa listy.

Kontrola uprawnień

- `Dokument biznesowy`: nazwa (w nawiasie kod) dokumentu biznesowego, do którego jest przypisana lista.
- `Dostęp dla posiadaczy ról globalnych`: tylko użytkownicy z rolami wybranymi na tej liście będą mogli wyświetlać listę z poziomu pozycji menu oraz w sekcji formularza i na widgecie. Jeśli lista będzie pusta, dostęp do niej nie będzie ograniczany

- `Weryfikacja uprawnień`: obejmuje dwa pola wyboru:
  - `Według szablonu instancji`: włącza weryfikację uprawnień ACL do rekordów na liście według szablonów ogólnych (bez wskazania statusu).
  - `Według szablonu statusu`: włącza weryfikację uprawnień ACL do rekordów na liście według szablonów z wskazanym statusem.

- `Weryfikacja uprawnień`: włącza lub wyłącza weryfikację uprawnień; przełącznik dostępny tylko w przypadku, gdy w definicji dokumentu biznesowego włączono obsługę uprawnień.

> **UWAGA**
> Włączenie weryfikacji uprawnień ACL powoduje, że zapytanie źródłowe widoku listy musi zwracać kolumny *ACLId* oraz *Status* (w przypadku weryfikacji według szablonu statusu).

Sortowanie zestawu danych listy

- `Klucz główny (PK)`: nazwa kolumny w źródle danych, która jest kolumną klucza głównego.

- `Sortowanie kolumn`: umożliwia skonfigurowanie domyślnego sortowania listy po dowolnych kolumnach zwracanych przez zapytanie źródłowe; wymaga wybrania

kolumn z listy wielowyboru, a następnie ustawienie kierunku oraz kolejności sortowania; konfiguracja sortowania po wielu kolumnach jest zapisywana w systemowej konfiguracji listy; alternatywnie sortowanie po wielu kolumnach można skonfigurować, klikając nagłówki kolumn na podglądzie listy.

> **WSKAZÓWKA**
> W konfiguracji sortowania można także używać kolumn, które nie są wyświetlane na liście (nieaktywne). W takim przypadku, użytkownika w aplikacji FrontSPA nie będzie mógł nadpisać tego sortowania.

Zestaw danych listy

- `SQL`: zapytanie SQL definiujące źródło danych dla listy; zapytanie może odwoływać się do tabeli (tabel) w bazie danych lub funkcji tabelarycznej (ustawienie domyślne dla list generowanych automatycznie)

> **WSKAZÓWKA**
> W przypadku list, które będą osadzone na formularzu w zapytaniu źródłowym można użyć odwołania do wartości pola z sekcji wskazanej we właściwości `Zależność od sekcji`. To odwołanie musi mieć postać*{@_Section_NazwaPola}, gdzie *_Section_* to stały przedrostek, a *NazwaPola* to nazwa pola w kontekście formularza (wartość atrybutu `Nazwa pola w DB` lub `Pole formularza`).
>
> Wymagane jest ustawienie dowolnego warunku zależności, np. *1=1*.

- `Sprawdź składnię SQL`: sprawdza składnię zapytania SQL i wyświetla odpowiedni komunikat.
- `Maksymalizuj edytor`: wyświetla edytor SQL w trybie pełnoekranowym, umożliwiając testowe wykonanie zapytania z ustawionymi wartościami parametrów.
- `Sortowanie kolumn`: umożliwia skonfigurowanie domyślnego sortowania listy po dowolnych kolumnach zwracanych przez zapytanie źródłowe; wymaga wybrania kolumn z listy wielowyboru, a następnie ustawienie kierunku oraz kolejności sortowania; konfiguracja sortowania po wielu kolumnach jest zapisywana w systemowej konfiguracji listy; alternatywnie sortowanie po wielu kolumnach można skonfigurować, klikając nagłówki kolumn na podglądzie listy.

> **WSKAZÓWKA**
> W konfiguracji sortowania można także używać kolumn, które nie są wyświetlane na liście (nieaktywne). W takim przypadku, użytkownika w aplikacji FrontSPA nie będzie mógł nadpisać tego sortowania.

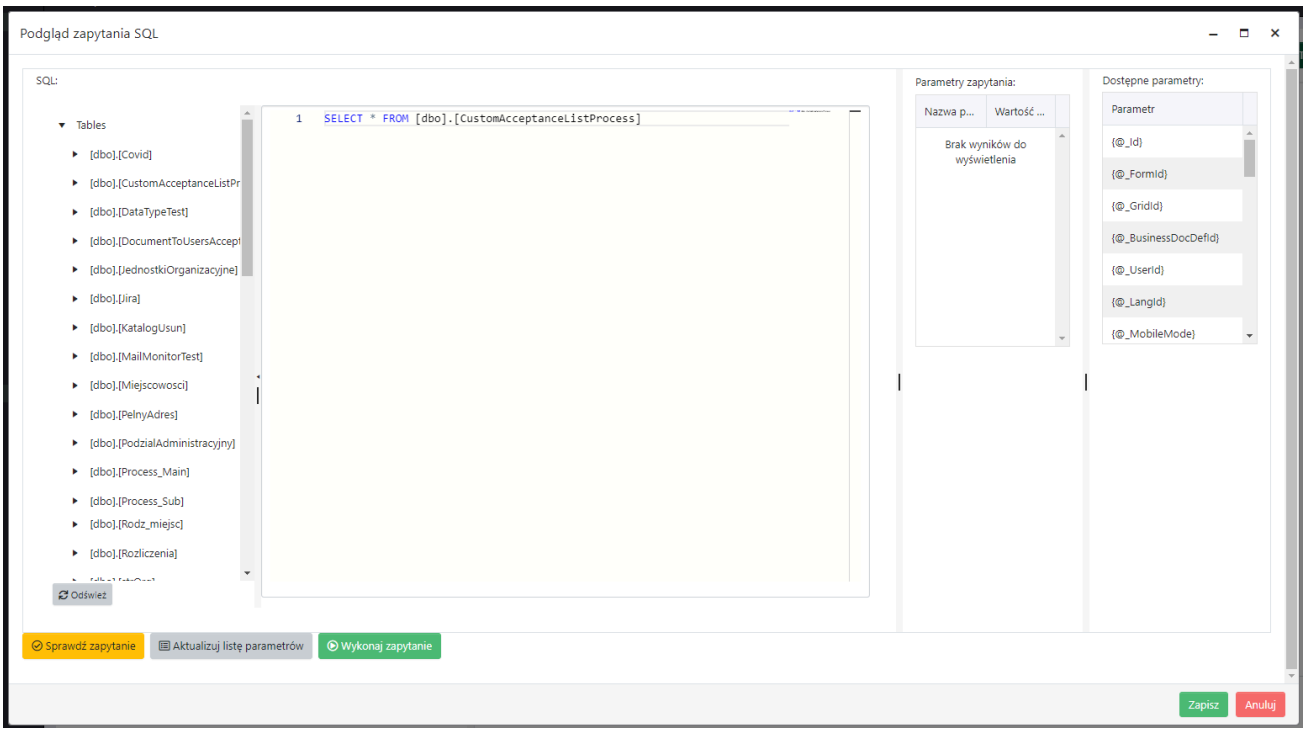
- `Podgląd wynikowego zapytania SQL`: wyświetla podgląd zapytania SQL, które zostanie wykonane na bazie danych w celu zwrócenia rekordów do widoku listy, zależnie od trybu wybranego na karcie `Ustawienia`.
- `Aktualizuj SQL`: aktualizuje zmienione zapytanie SQL.
- `Waliduj kolumny listy`: sprawdza zgodność kolumn używanych w źródle listy (domyślnie funkcja tabelaryczna) z definicja modelu danych; w przypadku wykrycia niezgodności oznacza kolumny w widoku drzewa czerwoną ikoną ostrzegawczą; konsultant musi samodzielnie poprawić właściwości kolumny.

Powiązane tematy:

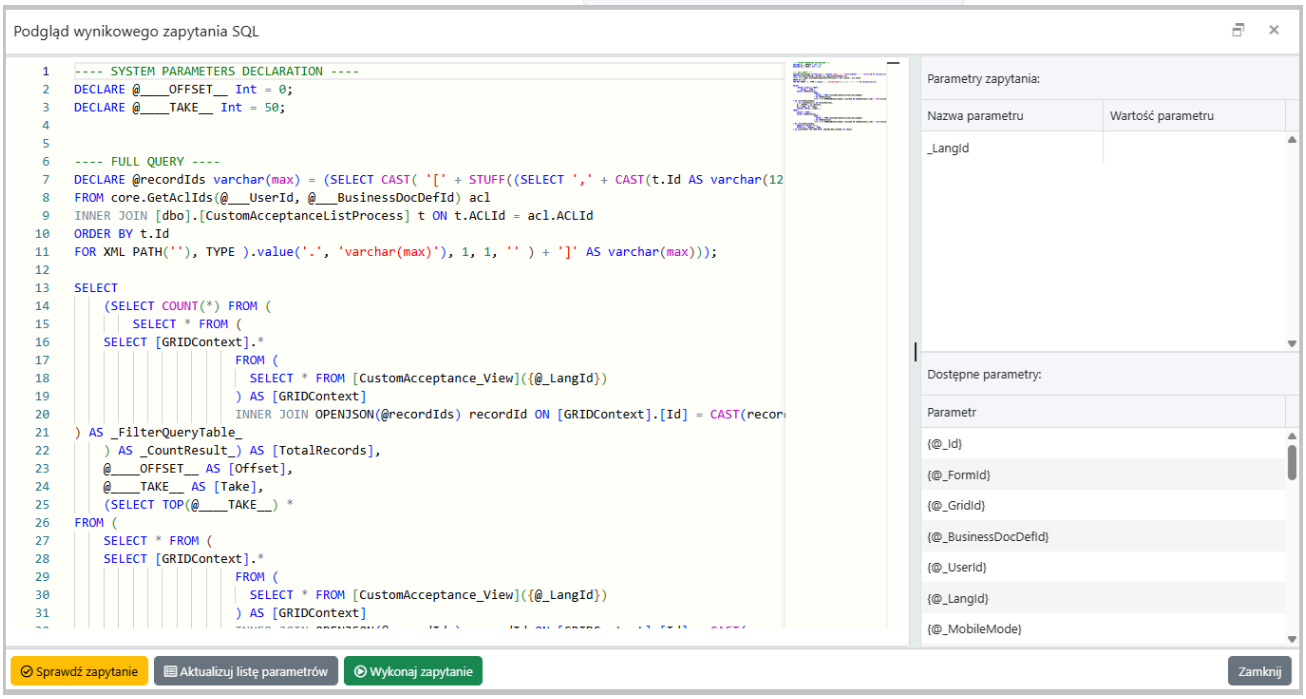
- Uprawnienia ACL do dokumentów

4.3.2.1. PODGLĄD/WYKONANIE ZAPYTANIA SQL

> ☛ *AdminSPA* > `INTERFEJS UŻYTKOWNIKA` > `Formularze` > `Edytuj` > `Podgląd zapytania SQL` `Podgląd wynikowego zapytania SQL` *AdminSPA* > `INTERFEJS UŻYTKOWNIKA` > `Listy` > `Edytuj` > `Podgląd zapytania SQL` `Podgląd wynikowego zapytania SQL`



Okno dialogowe `Podgląd zapytania SQL`



Okno dialogowe `Podgląd wynikowego zapytania SQL` (tryb hybrydowy)

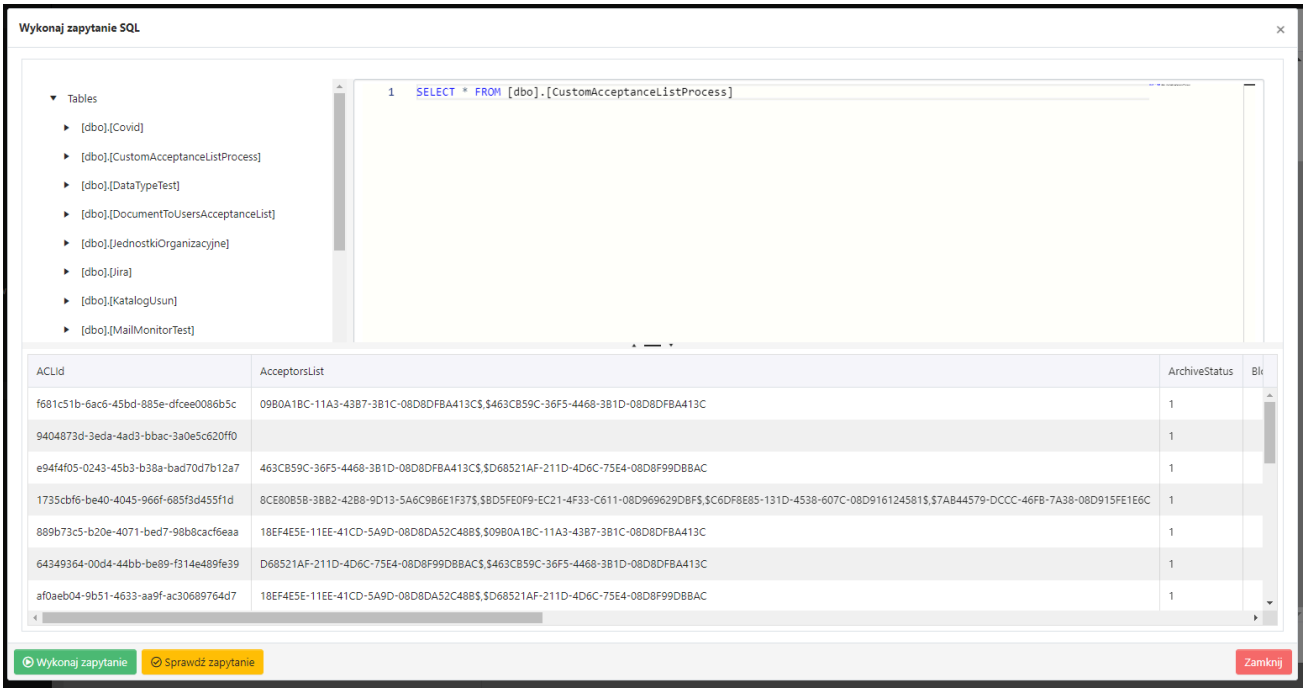
Z prawej strony edytora SQL znajduję się dwie listy:

- `Parametry zapytania`: pozwala podać wartość parametru *{@Nazwa}* użytego w zapytaniu, aby można było wyświetlić zwrócone wyniki.
- `Dostępne parametry`: kliknięcie dowolnego parametru na liście spowoduje wstawienie tego parametru do kodu zapytania SQL z lewej strony w miejscu kursora.

- `Dostępne parametry`: parametry, których można użyć w zapytaniu; po wpisaniu parametru w instrukcji zapytania należy kliknąć przycisk `Aktualizuj parametry` i nadać wartość parametru na liście `Parametry zapytania`.

Przycisk `Sprawdź zapytanie` sprawdza poprawność zapytania, a przycisk `Aktualizuj parametry` aktualizuje listę `Parametry zapytania`.

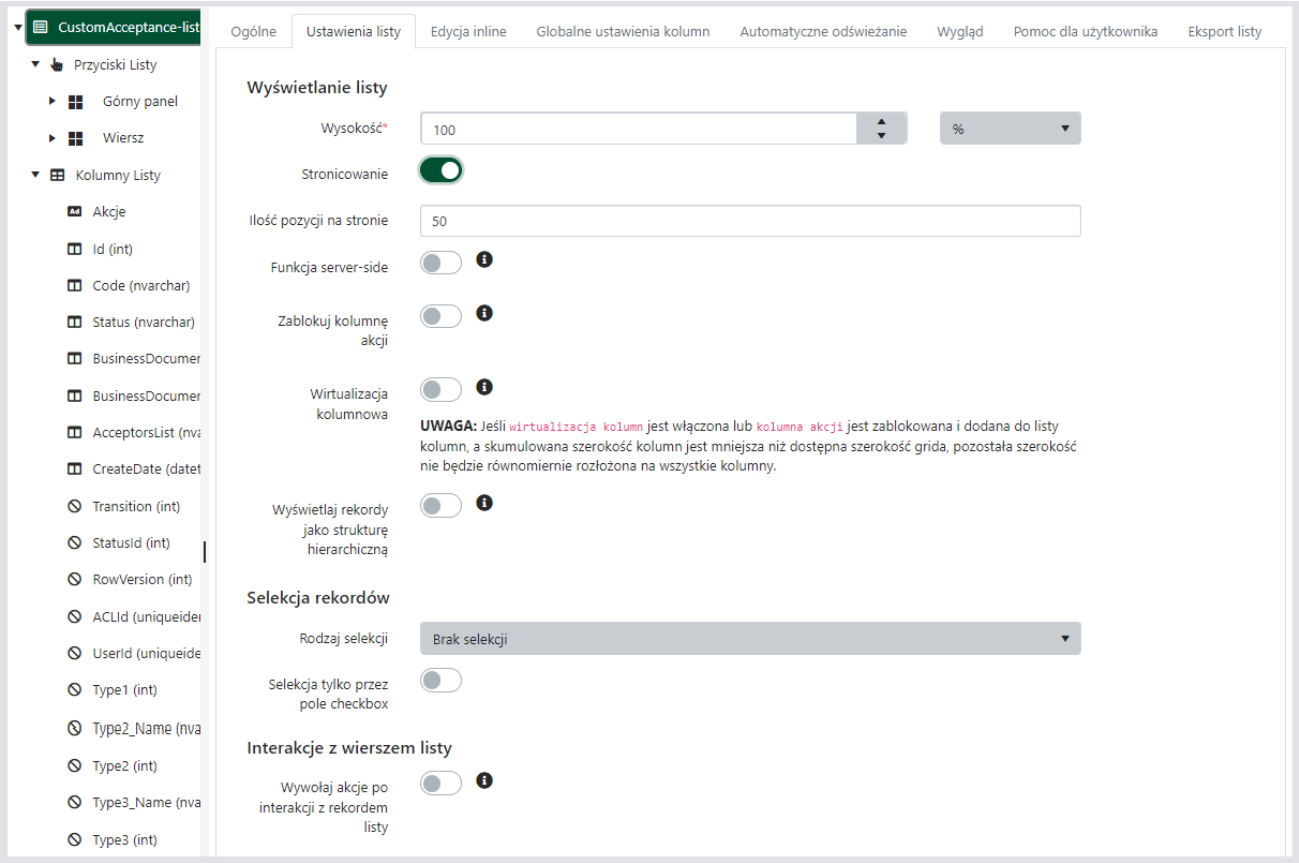
Po podstawieniu wartości parametrów kliknij przycisk `Wykonaj zapytanie`, aby wyświetlić okno `Wykonaj zapytanie SQL`, a następnie kliknij ponownie przycisk o tej samej nazwie, aby wykonać zapytanie. Wyniki zapytania zostaną wyświetlone u dołu okna.



Okno dialogowe `Wykonaj zapytanie SQL`

4.3.3. Właściwości listy — karta Ustawienia listy

> ☛ *AdminSPA* > `INTERFEJS UŻYTKOWNIKA` > `Listy` > `Edytuj` > `Ustawienia listy`



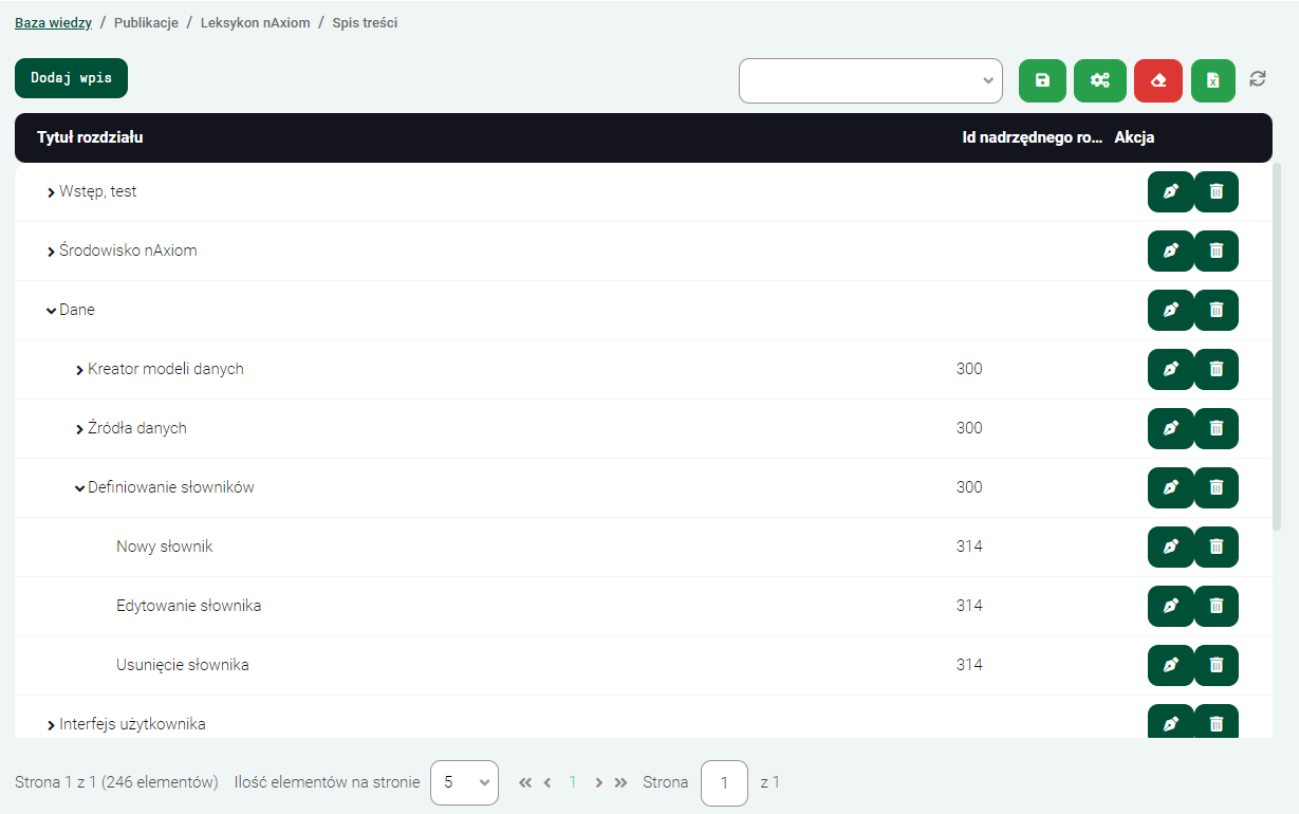
Właściwości listy, karta `Ustawienia listy`.

Na tej karcie można ustawić następujące właściwości listy:

- `Wysokość`: określa wysokość listy w oknie przeglądarki; wartość domyślna to `100%`; wartość można określić w procentach lub w pikselach, jednostkę wybiera się z listy rozwijanej po prawej stronie.
- `Stronicowanie`: przełącznik, którym można określić, czy lista ma być dzielona na strony; domyślnie włączony; jeśli lista ma być stronicowana dodatkowo należy określić liczbę wierszy wyświetlanych na stronie.
- `Ilość pozycji na stronie`: liczba wierszy, które będą wyświetlane na jednej stronie listy po włączeniu stronicowania; wartość domyślna `50`.
- `Funkcja server-side`: po włączeniu powoduje, że operacje filtrowania są wykonywane po stronie serwera, a wynikowe rekordy są przesyłane do przeglądarki; domyślnie wyłączona, włączenie może być konieczne w przypadku list liczących więcej niż kilkaset rekordów.
- `Zablokuj kolumnę akcji`: po włączeniu blokuje przewijania w poziomie kolumny akcji w oknie, dzięki czemu jest ona zawsze widoczna w oknie listy.
- `Wirtualizacja kolumnowa`: po włączeniu renderuje na ekranie tylko te kolumny, które mieszczą się w oknie listy w celu zwiększenia wydajności wyświetlania podczas pracy z ograniczonym zestawem kolumn na liście.
- `Wyświetlaj rekordy jako strukturę hierarchiczną`: po włączeniu i

wskazaniu kolumny hierarchii pozwala wyświetlić listę jako hierarchię rozwijaną (strukturę drzewiastą), to jest taką, w której rekordy nadrzędne grupują rekordy podrzędne, przy czym to grupowanie może być wielopoziomowe. Domyślnie wszystkie poziomy hierarchii są zwinięte, wyświetlenie rekordów podrzędnych wymaga kliknięcia symbolu `>` z lewej strony rekordu nadrzędnego.

- `Kolumna hierarchii`: nazwa kolumny zawierającej identyfikator rekordu nadrzędnego; wartości w tej kolumnie muszą być wartościami z kolumny klucza głównego wskazanej w definicji listy.
- `Rodzaj selekcji`: określa sposób zaznaczania rekordów na liście, dostępne wartości:
  - `Brak selekcji`: nie można zaznaczać rekordów na liście.
  - `Pojedyncza selekcja`: można zaznaczyć jeden rekord na liście (klikając go lewym przyciskiem myszy).
  - `Wybór wielokrotny`: można zaznaczyć wiele rekordów na liście (klikając je lewym przyciskiem myszy z wciśniętym klawiszem `Ctrl`).
  - `Pojedyncza selekcja – z użyciem pola checkbox`: można zaznaczyć jeden rekord na liście (klikając go lewym przyciskiem myszy lub zaznaczając pole wyboru); wymaga dodania do listy kolumny z polami wyboru przy użyciu polecenia `Dodaj kolumnę checkbox` z menu kontekstowego pozycji `Kolumny listy` na panelu elementów z lewej strony.
  - `Wybór wielokrotny – z użyciem pola checkbox`: można zaznaczyć wiele rekordów na liście (klikając je lewym przyciskiem myszy z wciśniętym klawiszem `Ctrl` lub zaznaczając pole wyboru); wymaga dodania do listy kolumny z polami wyboru przy użyciu polecenia `Dodaj kolumnę checkbox` z menu kontekstowego pozycji `Kolumny listy` na panelu elementów z lewej strony.
- `Selekcja tylko przez pole checkbox`: po włączeniu blokuje możliwość zaznaczania rekordów na liście przez klikanie myszą; ta opcja jest niedostępna w przypadku włączenia opcji `Wyświetlaj rekordy jako strukturę hierarchiczną`.
- `Wywołaj akcje po interakcji z rekordem listy`: Po włączeniu wyświetla komponent do wyboru akcji oraz powoduje wykonanie wybranych akcji, gdy użytkownik kliknie aktywną komórkę wiersza (lub naciśnie klawisz Enter). Funkcjonalność nie wspiera komórek kolumny akcji oraz kolumny selekcji rekordów. Włączenie tej funkcji wymusza wyłączenie edycji inline oraz wymusza możliwość selekcji rekordów tylko z użyciem kolumny checkbox. Głównym przeznaczeniem tej funkcjonalności jest otwieranie formularza z danymi bieżącego rekordu w trybie panelu bocznego.



Przykład listy hierarchicznej.

UWAGA

Nieprawidłowo skonstruowany warunek filtrowania listy hierarchicznej po wartościach kolumny hierarchii może powodować, że na liście nie będą wyświetlane żadne rekordy. Na przykład dla warunku:

```
WHERE ParentId = {@Section_ParentId}
```

gdzie *ParentId* to kolumna hierarchii, a *{@Section_ParentId}* to referencja do pola sterującego zależnością,
nie zostanie zwrócony rekord nadrzędny hierarchii, który ma inną wartość w tej kolumnie. Prawidłowy warunek musi zwrócić rekord nadrzędny i rekordy podrzędne:

```
WHERE Id = {@Section_ParentId} OR ParentId = {@Section_ParentId}
```

gdzie *Id* to kolumna klucza głównego.

Na tej karcie można ustawić następujące właściwości listy:

Pobieranie danych

- `Tryb pracy listy`: dostępne wartości:
  - `Baza danych - klasyczny`: Zapytanie wykonywane w całości przez bazę

danych. Każda operacja (pobranie rekordów, policzenie rekordów, wyliczenie funkcji w stopce) generuje oddzielne zapytanie z kontrolą dostępu realizowaną przez *JOIN* do tabeli ACL i weryfikację *ACLId*. Filtrowanie i sortowanie po stronie bazy danych. Zapytanie opakowane funkcjami systemowymi. Jest to tryb domyślny.
    - `Baza danych - CTE`: Wariant trybu klasycznego. Jedyna różnica jest taka, że w tym trybie system opakowuje źródłowe zapytanie SQL listy w wyrażenie *Common Table Expression* (CTE), aby nie powielać go w kodzie. Zapytanie wykonywane w całości przez bazę danych.
    - `Baza danych - hybrydowy`: Wariant trybu klasycznego ze zmienionym sposobem kontroli dostępu ACL. Zapytanie wykonywane w całości przez bazę danych. Każda operacja generuje oddzielne zapytanie z kontrolą dostępu, ale realizowaną przez *JOIN* do danych JSON z tablicą kluczy głównych instancji dokumentów. W tym trybie załadowanie pierwszej strony rekordów oraz oraz przełączanie między stronami zajmuje mniej więcej tyle samo czasu.
    - `Tylko przeglądarka WWW`: Cały zestaw danych pobierany jednorazowo z bazy danych i przekazywany do przeglądarki. Kontrola ACL przez JOIN do tabeli ACL (jedno zapytanie). Filtrowanie i sortowanie realizowane po stronie przeglądarki. Zapytanie opakowane funkcjami systemowymi. Tryb odpowiedni do list z niedużą liczbą rekordów oraz przypadków, gdy akceptowalny jest dłuższy czas ładowania pierwszej strony, a następnie szybkie przełączanie między stronami.
- `Stronicowanie`: przełącznik, którym można określić, czy lista ma być dzielona na strony; domyślnie włączony; jeśli lista ma być stronicowana dodatkowo należy określić liczbę wierszy wyświetlanych na stronie.
- `Liczba pozycji na stronie`: liczba wierszy, które będą wyświetlane na jednej stronie listy po włączeniu stronicowania; wartość domyślna `50`.
- `Wirtualizacja kolumnowa`: po włączeniu renderuje na ekranie tylko te kolumny, które mieszczą się w oknie listy w celu zwiększenia wydajności wyświetlania podczas pracy z ograniczonym zestawem kolumn na liście.
- `Tryb pracy listy`: dostępne wartości:
    - `Server-side`: (domyślnie włączone) operacje filtrowania są wykonywane po stronie serwera nAxiom, a wynikowe rekordy są przesyłane do przeglądarki; tryb zapewniający większą wydajność dla duzych zestawów danych.
    - `Client-side`: pobiera większy zestaw danych do przeglądarki i lokalnie wykonuje operacje filtrowania, sortowania i stronicowania; tryb odpowiedni dla mniejszych zestawów danych.

- `Stronicowanie`: przełącznik, którym można określić, czy lista ma być dzielona na strony; domyślnie włączony; jeśli lista ma być stronicowana dodatkowo należy określić liczbę wierszy wyświetlanych na stronie.
- `Liczba pozycji na stronie`: liczba wierszy, które będą wyświetlane na jednej stronie listy po włączeniu stronicowania; wartość domyślna `50`.

- `Sposób budowy zapytania SQL`: (widoczne tylko w trybie *Server-side*) dostępne wartości:
    - `Podstawowy`: (dotychczasowy) Zapytanie użytkownika jest wbudowywane osobno w każde podzapytanie (pobieranie danych, liczenie rekordów, obliczanie stopki). Najprostszy tryb w konfiguracji, jednak przy rozbudowanych zapytaniach może generować większe obciążenie bazy danych, ponieważ to samo zapytanie jest wykonywane wielokrotnie.
    - `CTE`: Zapytanie użytkownika jest definiowane raz jako wspólne wyrażenie tabelaryczne (CTE) i współdzielone przez wszystkie podzapytania (dane, liczba rekordów, stopka). Zalecany dla złożonych zapytań — eliminuje powtarzanie kodu SQL i pozwala silnikowi bazy danych lepiej zoptymalizować plan wykonania.

    - `Hybrydowy`: Uprawnienia i dane są pobierane w dwóch oddzielnych zapytaniach. W pierwszym kroku system pobiera wyłącznie kolekcję identyfikatorów rekordów dostępnych dla użytkownika zgodnie z jego uprawnieniami (ACL). W drugim kroku pobierane są właściwe dane – z filtrowaniem, sortowaniem i paginacją — ograniczone wyłącznie do dozwolonych rekordów. Zalecany w przypadku zwracania do 50 tys. rekordów, do których użytkownik posiada uprawnienia ACL.

    > **WAŻNE**
    > Dla ustawienia *Hybrydowy* sprawdzanie ACL odbywa się bezpośrednio na tabeli domyślnej dla definicji dokumentu wskazanej w definicji listy. W przypadku użycia w zapytaniu źródłowym listy innej tabeli lub funkcji tabelarycznej zbudowanej na innej tabeli lista nie będzie działać.

- `Wirtualizacja kolumnowa`: po włączeniu renderuje na ekranie tylko te kolumny, które mieszczą się w oknie listy w celu zwiększenia wydajności wyświetlania podczas pracy z ograniczonym zestawem kolumn na liście.

Wyświetlanie danych

- `Filtrowanie`: kliknięcie nagłówka kolumny z włączonym filtrowaniem powoduje wyświetlenie panelu, w którym można filtrować wartości wyświetlane w kolumnie. Sposoby filtrowania konfiguruje się we właściwościach poszczególnych kolumn.

- `Sortowanie`: włącza możliwość sortowania wierszy na liście według wartości wybranej kolumny; wymaga kliknięcia nagłówka kolumny; pierwsze kliknięcie powoduje sortowanie w kolejności rosnącej, drugie — malejącej, a trzecie wyłącza sortowanie.
- `Grupowanie`: włącza możliwość grupowania rekordów na liście według wartości wybranej kolumny; wymaga przeciągnięcia nagłówka kolumny do obszaru bezpośrednio pod panelem przycisków u góry.
- `Domyślnie zwijaj grupy rekordów`: po włączeniu powoduje, że w przypadku korzystania z funkcji grupowania grupy rekordów są domyślnie zwinięte.
- `Struktura danych listy`: dostępne wartości:
    - `płaska`: rekordy są wyświetlane jako lista bez struktury.
    - `hierarchiczna`: rekordy listy są wyświetlane jako hierarchia rozwijana (struktura drzewiasta) — rekordy nadrzędne grupują rekordy podrzędne, przy czym to grupowanie może być wielopoziomowe. Domyślnie wszystkie poziomy hierarchii są zwinięte, wyświetlenie rekordów podrzędnych wymaga kliknięcia symbolu `>` z lewej strony rekordu nadrzędnego; wymaga określenia wartości w polu `Kolumna hierarchii`.

- `Kolumna hierarchii`: nazwa kolumny zawierającej identyfikator rekordu nadrzędnego; wartości w tej kolumnie muszą być wartościami z kolumny klucza głównego wskazanej w definicji listy.

> **UWAGA**
> Nieprawidłowo skonstruowany warunek filtrowania listy hierarchicznej po wartościach kolumny hierarchii może powodować, że na liście nie będą wyświetlane żadne rekordy. Na przykład dla warunku:
>
> ```
> WHERE ParentId = {@Section_ParentId}
> ```
>
> gdzie *ParentId* to kolumna hierarchii, a *{@Section_ParentId}* to referencja do pola sterującego zależnością,
> nie zostanie zwrócony rekord nadrzędny hierarchii, który ma inną wartość w tej kolumnie. Prawidłowy warunek musi zwrócić rekord nadrzędny i rekordy podrzędne:
>
> ```
> WHERE Id = {@Section_ParentId} OR ParentId = {@Section_ParentId}
> ```
>
> gdzie *Id* to kolumna klucza głównego.

- `Wyświetlanie treści rekordów`: określa zachowanie w przypadku, kiedy wartość

w kolumnie nie może zostać wyświetlona w całości; dostępne opcje:
    - `Skracaj i dopisuj wielokropek`
    - `Zawijaj treść`
- `Zablokuj kolumnę akcji`: po włączeniu blokuje przewijania w poziomie kolumny akcji w oknie, dzięki czemu jest ona zawsze widoczna w oknie listy.

Selekcja rekordów

- `Rodzaj selekcji`: określa sposób zaznaczania rekordów na liście, dostępne wartości:
    - `Brak selekcji`: nie można zaznaczać rekordów na liście.
    - `Pojedyncza selekcja`: można zaznaczyć jeden rekord na liście (klikając go lewym przyciskiem myszy).
    - `Wybór wielokrotny`: można zaznaczyć wiele rekordów na liście (klikając je lewym przyciskiem myszy z wciśniętym klawiszem `Ctrl`).
    - `Pojedyncza selekcja – z użyciem pola checkbox`: można zaznaczyć jeden rekord na liście (klikając go lewym przyciskiem myszy lub zaznaczając pole wyboru); wymaga dodania do listy kolumny z polami wyboru przy użyciu polecenia `Dodaj kolumnę checkbox` z menu kontekstowego pozycji `Kolumny listy` na panelu elementów z lewej strony.
    - `Wybór wielokrotny – z użyciem pola checkbox`: można zaznaczyć wiele rekordów na liście (klikając je lewym przyciskiem myszy z wciśniętym klawiszem `Ctrl` lub zaznaczając pole wyboru); wymaga dodania do listy kolumny z polami wyboru przy użyciu polecenia `Dodaj kolumnę checkbox` z menu kontekstowego pozycji `Kolumny listy` na panelu elementów z lewej strony.

W przypadku listy osadzonej na formularzu (w sekcji typu lista) w kontekście formularza zapisywane są identyfikatory tych rekordów. Wartości te znajdują się w tabeli *Selected* i są dostepne poprzez zmienną *{@KodSekcji_Selected}*. Tej zmiennej można używać np. w akcjach SQL przypisanych do przycisku na formularzu.

Innym zastosowaniem tablicy *Selected* jest filtrowanie listy osadzonej na formularzu na podstawie rekordów zaznaczonych na innej liście osadzonej w tym samym formularzu. Służy do tego funkcja zależności od sekcji wraz z warunkiem zależności .

Interakcje z wierszem listy

- `Wywołaj akcje po interakcji z rekordem listy`: Po włączeniu wyświetla komponent do wyboru akcji oraz powoduje wykonanie wybranych akcji, gdy użytkownik kliknie aktywną komórkę wiersza (lub naciśnie klawisz Enter).

Funkcjonalność nie wspiera komórek kolumny akcji oraz kolumny selekcji rekordów. Włączenie tej funkcji wymusza wyłączenie edycji inline oraz wymusza możliwość selekcji rekordów tylko z użyciem kolumny checkbox. Głównym przeznaczeniem tej funkcjonalności jest otwieranie formularza z danymi bieżącego rekordu w trybie panelu bocznego.

Odświeżanie listy

Umożliwia włączenie odświeżania widoku listy wyświetlanego przez użytkowników zalogowanych do FrontSPA. Dostępne tryby odświeżania to:

- `Sterowane przez akcję`: widok listy wyświetlany u wszystkich użytkowników zostanie odświeżony w wyniku uruchomienia akcji Odśwież listę u wszystkich użytkowników , o ile w akcji wybrano tryb wymuszonego odświeżenia, w przeciwnym razie po uruchomieniu akcji w widoku listy będzie pulsował przycisk odświeżania w prawym górnym rogu, informując o konieczności ręcznego odświeżenia listy przez użytkownika.
- `Cykliczne`: powoduje automatyczne odświeżanie widoku listy z podanym interwałem.
    - `Interwał odświeżania (sek.)`: czas w sekundach pomiędzy kolejnymi odświeżeniami widoku listy.
- `Wyłączone`; (ustawienie domyślne) odświeżanie listy jest wyłączone.

UWAGI

- Jeśli użytkownik edytuje inline dane w widoku listy, cykliczne odświeżanie listy zostanie wstrzymane, a odświeżanie automatyczne będzie tylko sygnalizować konieczność odświeżenia pulsowaniem przycisku, bez względu na tryb odświeżania wybrany w akcji.
- W przypadku wywołania akcji w odniesieniu do listy, dla której nie włączono automatycznego odświeżania, to jedynym skutkiem zalogowanie informacji o tym przycisku do logu; nie zostanie zwrócony błąd ani wyjątek.

Opcje listy dla użytkownika końcowego

- `Wyświetlaj pole wyszukiwania`: włącza/wyłącza wyświetlanie pola wyszukiwania.
- `Wyświetlaj zdefiniowane filtry listy`: włącza/wyłącza wyświetlanie pola filtrowania z listą wyboru zdefiniowanych filtrów.
- `Wyświetlaj przycisk odświeżania listy`: włącza/wyłącza wyświetlanie przycisku do odświeżania listy.

- `Wyświetlaj przycisk eksportu listy do Excela`: włącza/wyłącza wyświetlanie polecenia `Eksport do MS Excel` w menu przycisku `Odśwież`.
- `Wyświetlaj konfiguracje listy w nagłówku`: włącza/wyłącza wyświetlanie polecenia `Konfiguracje listy` w menu przycisku `Odśwież`.
- `Widoczność kolumn`: steruje dostępem do opcji `Widoczność kolumn` w menu rozwijanym po kliknięciu strzałki obok przycisku odświeżania w widoku listy na Front SPA.
- `Pomoc dla użytkownika`
    - `Pokaż pomoc, jeśli istnieje`: (ustawienie domyślne) przyciski pomocy są wyświetlane, jeśli zdefiniowano akcje wywoływane przyciskiem pomocy.
    - `Zawsze pokazuj pomoc`; przycisk pomocy jest wyświetlany na formularzu/liście niezależnie od tego, czy skonfigurowano jego obsługę.
    - `Nigdy nie pokazuj pomocy`: ukrywa przycisk pomocy na liście/formularzu; ustawienie przydatne np. w przypadku formularza do wyświetlania pomocy.

Eksport listy

- `Nazwa eksportowanego pliku`: pozwala określić nazwę pliku generowanego po kliknięciu przycisku eksportu danych listy do formatu MS Excel; dostępne wartości:
    - `Taka jak kod listy`: (domyślnie) nazwa wyeksportowanego pliku ma postać *kodListy.xlsx*.
    - `Taka jak nazwa listy`: nazwa wyeksportowanego pliku ma postać *nazwaListy.xlsx*.
    - `Zapytanie SQL`: nazwa wyeksportowanego pliku jest zwracana przez zapytanie SQL, które należy wpisać w wyświetlonym oknie edytora; do wyniku zapytania jest dodawane rozszerzenie *.xlsx*.

4.3.4. Właściwości listy — karta Edycja inline

Podsumowanie:
Na tej karcie można włączyć jeden z dwóch trybów edytowania danych bezpośrednio na liście (*edycja inline*). W jednym z trybów rekordy edytuje się pojedynczo, a przyciski edycji i zapisu znajdują się w kolumnie akcji. Drugi tryb umożliwia edytowanie danych bezpośrednio po kliknięciu w komórkę listy i zbiorcze zapisywanie zmian w wielu rekordach na stronie. W tym drugim trybie nie można dodawać rekordów.

Zakres operacji (dodawanie, edycji, usuwanie) dostępnych w trybie edycji inline definiuje się osobnymi przełącznikami. Włączenie przełączników poszczególnych funkcji edycji na karcie `Edycja inline` powoduje dodanie odpowiednich obiektów w drzewie elementów listy, w gałęzi *Przyciski listy/Wiersz* i/lub *Przyciski listy/Górny panel*. Przyciski `Zapisz` i `Usuń` wymagają zdefiniowania indywidualnych akcji SQL, które będą wykonywane po kliknięciu przycisku.
Ponadto, po włączeniu trybu edycji inline we właściwościach kolumn pojawi się karta `Edycja inline`, na której określa się między innymi typ kontrolki wyświetlanej w danej kolumnie podczas edycji inline, wartość domyślną oraz, zależnie od typu danych kolumny, wartość minimalną i maksymalną.

☛ *AdminSPA* > `INTERFEJS UŻYTKOWNIKA` > `Listy` > `Edytuj` > `Edycja inline`

Na karcie można ustawić następujące właściwości:

Tryb edycji inline

- `Tryb edycji inline`: dostępne wartości:
  - `po kliknięciu w wiersz`: aby edytować dane na liście, należy kliknąć odpowiednią komórkę. W tym trybie można zmieniać wartości w wielu wierszach i zbiorczo zapisać (anulować) zmiany; przyciski zapisu i zanulowania zmian znajdują się na górnym panelu listy. Podczas edycji rekordu w kolumnie akcji wyświetlane są przyciski zatwierdzania i anulowania zmian przed przejściem do innego wiersza. Pozwalają one wyłączyć tryb edycji bieżącego rekordu, ale nie zastępują funkcji przycisków `Zapisz` i `Anuluj` na górnym panelu listy.
  - `po kliknięciu w przycisk edycji`: wejście w tryb edycji wymaga kliknięcia przycisk w kolumnie akcji w danym rekordzie. W tym trybie trzeba zapisać (anulować) zmiany w każdym rekordzie osobno; przyciski zapisu i anulowania znajdują się w każdym wierszu w kolumnie akcji, przycisk dodawania nowego wiersza znajduje się na panelu górnym.
  - `wyłączona`: edycja inline jest wyłączona.

> **UWAGA**
> Nie można używać jednocześnie funkcji `Wywołaj akcje po interakcji z rekordem listy` na karcie `Ustawienia listy` oraz trybu edycji inline `po kliknięciu w wiersz`. Włączenie jednej z tych funkcji spowoduje wyświetlenie komunikatu o konieczności wyłączenia drugiej.

- `po kliknięciu w wiersz`: aby edytować dane na liście, należy kliknąć odpowiednią komórkę. W tym trybie można zmieniać wartości w wielu wierszach i zbiorczo zapisać (anulować) zmiany; przyciski zapisu i zanulowania zmian znajdują się na górnym panelu listy. Podczas edycji rekordu w kolumnie akcji wyświetlane są przyciski zatwierdzania i anulowania zmian przed przejściem do innego wiersza. Pozwalają one wyłączyć tryb edycji bieżącego rekordu, ale nie zastępują funkcji przycisków `Zapisz` i `Anuluj` na górnym panelu listy.
- `po kliknięciu w przycisk edycji`: wejście w tryb edycji wymaga kliknięcia przycisk w kolumnie akcji w danym rekordzie. W tym trybie trzeba zapisać (anulować) zmiany w każdym rekordzie osobno; przyciski zapisu i anulowania znajdują się w każdym wierszu w kolumnie akcji, przycisk dodawania nowego wiersza znajduje się na panelu górnym.

- `wyłączona`: edycja inline jest wyłączona.

> **UWAGA**
> Nie można używać jednocześnie funkcji `Wywołaj akcje po interakcji z rekordem listy` na karcie `Ustawienia listy` oraz trybu edycji inline `po kliknięciu w wiersz`. Włączenie jednej z tych funkcji spowoduje wyświetlenie komunikatu o konieczności wyłączenia drugiej.

Dostępne operacje

- `Dostępne operacje`:
  - `Dodawanie rekordu`: wstawia przycisk `Dodaj` na górnym panelu listy; kliknięcie tego przycisku dodaje nowy, pusty wiersz do listy z włączonym trybem edycji inline; ten przycisk jest wyświetlany na górnym panelu listy z lewej strony.
  - `Edycja rekordu`: zależnie od trybu edycji inline, wstawia w kolumnie akcji przycisk `Edycja`, którego kliknięcie włącza tryb edycji danych w bieżącym wierszu listy, lub aktywuje włączanie trybu edycji po kliknięciu w komórkę danych na liście.
  - `Usuwanie rekordu`: wstawia przycisk `Usuń` w kolumnie akcji; kliknięcie tego przycisku usuwa bieżący wiersz; we właściwościach tego przycisku

należy skonfigurować zapytanie SQL dla właściwości `Akcja`.

- `Dodawanie rekordu`: wstawia przycisk `Dodaj` na górnym panelu listy; kliknięcie tego przycisku dodaje nowy, pusty wiersz do listy z włączonym trybem edycji inline; ten przycisk jest wyświetlany na górnym panelu listy z lewej strony.
- `Edycja rekordu`: zależnie od trybu edycji inline, wstawia w kolumnie akcji przycisk `Edycja`, którego kliknięcie włącza tryb edycji danych w bieżącym wierszu listy, lub aktywuje włączanie trybu edycji po kliknięciu w komórkę danych na liście.
- `Usuwanie rekordu`: wstawia przycisk `Usuń` w kolumnie akcji; kliknięcie tego przycisku usuwa bieżący wiersz; we właściwościach tego przycisku należy skonfigurować zapytanie SQL dla właściwości `Akcja`.

- `Dodawanie elementu podrzędnego`: przełącznik dostępny tylko dla listy hierarchicznej; po włączeniu umożliwia dodawanie rekordów podrzędnych w trybie edycji inline.

- `Dynamiczne przeliczanie stopki`: po włączeniu powoduje, że wartości funkcji agregacyjnych w stopkach są automatycznie przeliczane po zmianie wartości w trybie edycji inline.

> **WSKAZÓWKA**
> Włączenie przełącznika `Dodawanie` i/lub `Edycja` powoduje także dodanie w kolumnie akcji przycisków `Zapisz` i `Anuluj`. Dla przycisku zapisu należy zdefiniować akcję zapisu w jego właściwościach.

Obsługa błędów

- `Identyfikator rekordu do obsługi błędów`: dowolna kolumna listy, np. *Id*, *Code* itp., z której wartość będzie używana w wyświetlanych komunikatach o błędach.

Zbiór akcji edycji inline

- `Zbiór akcji do wywołania na otwarcie i zmianę wartości rekordu w edycji inline`: zestaw akcji Javascript, których celem powinno być konfigurowanie możliwości edycji i wymagalności poszczególnych komórek w trybie edycji inline. Akcje są wykonywane po wejściu w tryb edycji oraz po wprowadzeniu zmian w komórkach rekordu (zmiana kontekstu rekordu).
  W akcjach można korzystać z obiektu *rowContext*, który zawiera wartości z poszczególnych pól rekordu oraz dodatkowo wartości pomocnicze zwracane przez listy rozwijane używane w polach rekordu. Do odwoływania się do poszczególnych komórek służy funkcja `GetControlByColumnId(columnName)`, gdzie *columnName*

to wartość atrybutu `Nazwa SQL` danej kolumny. Funkcja zwraca obiekt reprezentujący kontrolkę używaną w danej kolumnie lub *null*, jeśli kolumna o podanej nazwie nie istnieje.
Dla zwróconego obiektu kontrolki są dostępne następujące metody:

- *setRequired(isRequired: boolean)*: ustawia właściwość *iSRequired* (wymagalność),
- *setDisabled(isDisabled: boolean)*: ustawia właściwość *isDisabled* (edytowalność),
- *setValue(value)*: ustawia wartość kontrolki; argument *value* może być wyrażeniem odwołującym sie do wartości w innych polach danego rekordu.

Przykłady:

```
const ctrlEnvType = getControlByColumn('RodzajKoperty');
if (rowContext.RodzajWysylki === 'list'){
ctrlEnvType
  .setDisabled(false)
  .setRequired(true);
}
```

```
const columnA = getControlByColumn('kolumna_A');
columnA.setValue(rowContext.kolumna_B * rowContext.kolumna_C);
```

gdzie parametry *kolumna_A* i *kolumna_B* to wartości właściwości `Nazwa SQL` odpowiednich kolumn listy.

UWAGI

- Nie jest obsługiwane jednoczesne włączenie dla kontrolki obu właściwości (kontrolka tylko do odczytu nie może być wymagalna).
- W przypadku kontrolki typu checkbox ustawienie wymagalności oznacza, że pole wyboru w danym rekordzie musi być zaznaczone.
- Warto zwrócić uwagę na różnice w działaniu operatorów `==` i `===` w JavaScript.

- `Dodawanie elementu podrzędnego`: przełącznik dostępny tylko dla listy hierarchicznej; po włączeniu umożliwia dodawanie rekordów podrzędnych w trybie edycji inline.

- `Dynamiczne przeliczanie stopki`: po włączeniu powoduje, że wartości funkcji agregacyjnych w stopkach są automatycznie przeliczane po zmianie wartości w trybie edycji inline.

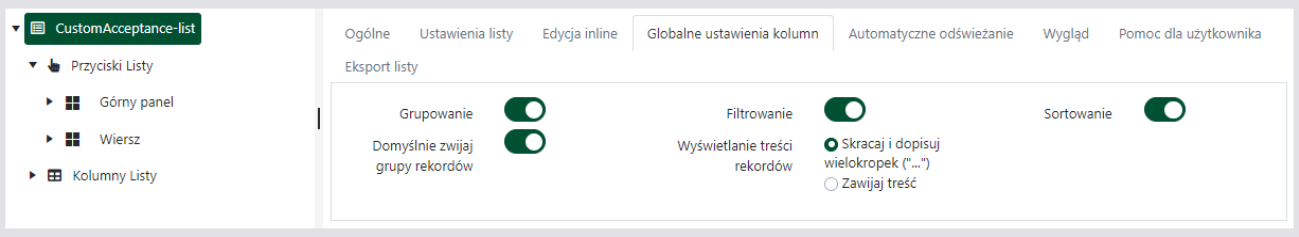
Powiązane tematy:

- Kolumna listy — karta Edycja inline
- Przyciski listy do edycji inline

4.3.5. Właściwości listy — karta Globalne ustawienia kolumn

Podsumowanie:
Właściwości określane na tej karcie są globalnymi ustawieniami dla kolumn. Wszystkie te ustawienia można określać indywidualnie we właściwościach poszczególnych kolumn. Przy czym jeśli dana właściwość jest włączona globalnie, można ją wyłączyć indywidualnie dla kolumny. Jeśli właściwość jest wyłączona globalnie, włączenie jej dla kolumny nie odniesie skutku.

☛ *AdminSPA* > `INTERFEJS UŻYTKOWNIKA` > `Listy` > `Edytuj` > `Globalne ustawienia kolumn`



Właściwości listy, karta `Globalne ustawienia kolumn`.

Na karcie można określić następujące właściwości listy:

- `Grupowanie`: włącza możliwość grupowania rekordów na liście według wartości wybranej kolumny; wymaga przeciągnięcia nagłówka kolumny do obszaru bezpośrednio pod panelem przycisków u góry.
- `Filtrowanie`: kliknięcie nagłówka kolumny z włączonym filtrowaniem powoduje wyświetlenie panelu, w którym można filtrować wartości wyświetlane w kolumnie. Sposoby filtrowania konfiguruje się we właściwościach poszczególnych kolumn.
- `Sortowanie`: włącza możliwość sortowania wierszy na liście według wartości wybranej kolumny; wymaga kliknięcia nagłówka kolumny; pierwsze kliknięcie powoduje sortowanie w kolejności rosnącej, drugie — malejącej, a trzecie wyłącza sortowanie.
- `Domyślnie zwijaj grupy rekordów`: po włączeniu powoduje, że w przypadku korzystania z funkcji grupowania grupy rekordów są domyślnie zwinięte.
- `Wyświetlanie treści rekordów`: określa zachowanie w przypadku, kiedy wartość w kolumnie nie może zostać wyświetlona w całości; dostępne opcje:

- `Skracaj i dopisuj wielokropek`
- `Zawijaj treść`

4.3.6. Właściwości listy — karta Automatyczne odświeżanie

> ☛ *AdminSPA* > `INTERFEJS UŻYTKOWNIKA` > `Listy` > `Edytuj` > `Automatyczne odświeżanie`

Umożliwia włączenie odświeżania widoku listy wyświetlanego przez użytkowników zalogowanych do FrontSPA. Dostępne tryby odświeżania to:

- `Automatyczne`: widok listy wyświetlany u wszystkich użytkowników zostanie odświeżony w wyniku uruchomienia akcji Odśwież listę u wszystkich użytkowników , o ile w akcji wybrano tryb wymuszonego odświeżenia, w przeciwnym razie po uruchomieniu akcji w widoku listy będzie pulsował przycisk odświeżania w prawym górnym rogu, informując o konieczności ręcznego odświeżenia listy przez użytkownika.
- `Cykliczne`: powoduje automatyczne odświeżanie widoku listy z podanym interwałem.
  - `Interwał odświeżania (sek.)`: czas w sekundach pomiędzy kolejnymi odświeżeniami widoku listy.
- `Wyłączone`; odświeżanie listy jest wyłączone.

UWAGI

- Jeśli użytkownik edytuje inline dane w widoku listy, cykliczne odświeżanie listy zostanie wstrzymane, a odświeżanie automatyczne będzie tylko sygnalizować konieczność odświeżenia pulsowaniem przycisku, bez względu na tryb odświeżania wybrany w akcji.
- W przypadku wywołania akcji w odniesieniu do listy, dla której nie włączono automatycznego odświeżania, to jedynym skutkiem zalogowanie informacji o tym przycisku do logu; nie zostanie zwrócony błąd ani wyjątek.

4.3.7. Właściwości listy - karta Wygląd

Podsumowanie:
Ustawienia na karcie `Wygląd` dotyczą rozmiaru i koloru czcionki oraz koloru tła używanych i wyrównania wartości w poszczególnych obszarach listy, a także elementów wyświetlanych na panelu górnym listy w aplikacji *FrontSPA*. Te ustawienia zostały pogrupowane na czterech kartach podrzędnych: `Ustawienia globalne grida`, `Nagłówek`, `Wiersze` i `Stopka`.

☛ *AdminSPA* > `INTERFEJS UŻYTKOWNIKA` > `Listy` > `Edytuj` > `Wygląd`

Na karcie `Ustawienia globalne grida` można włączyć/wyłączyć następujące przełączniki dotyczące wyświetlania poszczególnych elementów w panelu górnym listy w aplikacji *FrontSPA*:

- `Wyświetlaj pole wyszukiwania`: włącza/wyłącza wyświetlanie pola wyszukiwania.
- `Wyświetlaj zdefiniowane filtry listy`: włącza/wyłącza wyświetlanie pola filtrowania z listą wyboru zdefiniowanych filtrów.
- `Wyświetlaj konfiguracje listy w nagłówku`: włącza/wyłącza wyświetlanie polecenia `Konfiguracje listy` w menu przycisku `Odśwież`.
- `Wyświetlaj przycisk eksportu listy do Excela`: włącza/wyłącza wyświetlanie polecenia `Eksport do MS Excel` w menu przycisku `Odśwież`.
- `Wyświetlaj przycisk odświeżania listy`: włącza/wyłącza wyświetlanie przycisku do odświeżania listy.

Na kartach `Nagłówek`, `Wiersze` i `Stopka` są dostępne następujące ustawienia:

- `Rozmiar czcionki (rem)`: rozmiar czcionki w jednostkach rem (względem czcionki głównego elementu html); wartość domyślna *1*.
- `Tło`: kolor tła wiersza; ustawienie dostępne dla nagłówka i stopki.
- `Kolor czcionki`: kolor czcionki; ustawienie dostępne dla nagłówka i stopki.
- `Tło (wiersz parzysty)`: kolor tła wierszy parzystych (tylko sekcja `Wiersze`).
- `Tło (wiersz nieparzysty)`: kolor tła wierszy nieparzystych; (tylko sekcja `Wiersze`).
- `Kolor czcionki (wiersz nieparzysty)`: kolor czcionki w wierszach nieparzystych (tylko karta `Wiersze`).
- `Kolor czcionki (wiersz parzysty)`: kolor czcionki w wierszach parzystych

(tylko karta `Wiersze`).

- `Pogrubienie czcionki`: włącza pogrubienie czcionki używanej w stopce listy.
- `Pochylenie czcionki`: włącza pochylenie czcionki używanej w stopce listy.
- `Wyrównanie`: określa wyrównanie wartości w danej sekcji listy; dostępne wartości: `do lewej`, `do środka`, `do prawej`.

> **WSKAZÓWKA**
> W przypadku kolumn specjalnych (akcje, checkbox, HTML, właściciel, termin) ich właściwości mogą mieć priorytet przed ustawieniami globalnymi, np. ustawienia koloru tła dla wierszy parzystych i nieparzystych.

Czcionka

W sekcjach **Nagłówek**, **Wiersze** i **Stopka** są dostępne następujące ustawienia:

- `Pogrubienie czcionki`: włącza pogrubienie czcionki używanej w stopce listy.
- `Pochylenie czcionki`: włącza pochylenie czcionki używanej w stopce listy.
- `Wyrównanie`: określa wyrównanie wartości w danej sekcji listy; dostępne wartości: `do lewej`, `do środka`, `do prawej`.

Styl

Opisane poniżej ustawienia są dostepne osobno dla predefiniowanych motywów na kartach `Jasny`, `Ciemny` i `Wysoki kontrast`.

- `Ustawienia lokalne dla wszystkich kolumn`: gdy ustawienie jest wyłączone, wygląd listy dla danego motywu jest określany przez styl tego motywu; po włączeniu można zdefiniować poniższe ustawienia dla każdego motywu, osobno dla sekcji nagłówka, wierszy i stopki:
  - `Rozmiar czcionki (rem)`: rozmiar czcionki w jednostkach rem (względem czcionki głównego elementu html); wartość domyślna *1*.
  - `Tło`: kolor tła wiersza; ustawienie dostępne dla nagłówka i stopki.
  - `Kolor czcionki`: kolor czcionki; ustawienie dostępne dla nagłówka i stopki.
  - `Tło (wiersz parzysty)`: kolor tła wierszy parzystych (tylko sekcja `Wiersze`).
  - `Tło (wiersz nieparzysty)`: kolor tła wierszy nieparzystych; (tylko sekcja `Wiersze`).
  - `Kolor czcionki (wiersz nieparzysty)`: kolor czcionki w wierszach

nieparzystych (tylko karta `Wiersze`).

- `Kolor czcionki (wiersz parzysty)`: kolor czcionki w wierszach parzystych (tylko karta `Wiersze`).

Powiązane tematy:

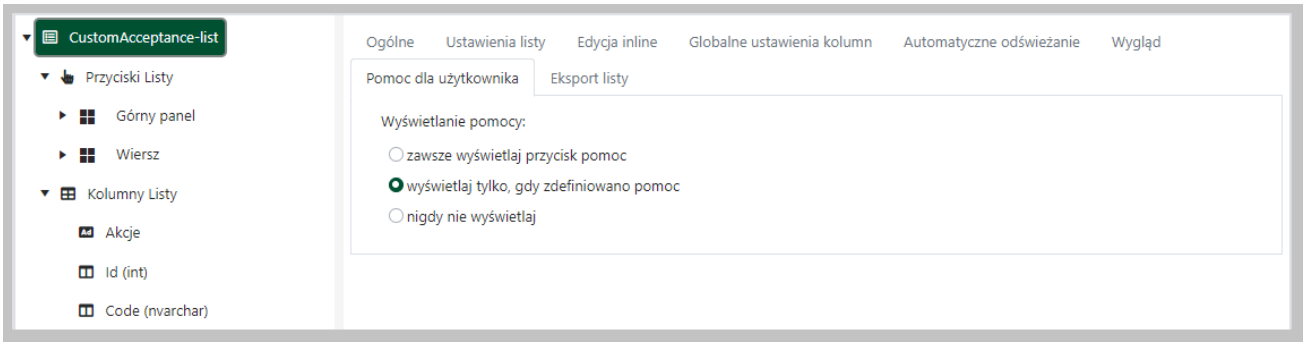
- FrontSPA, elementy widoku listy

4.3.8. Właściwości listy — karta Pomoc dla użytkownika

Podsumowanie:

Na tej karcie można zdefiniować opcje wyświetlania przycisku pomocy dla widoku listy.

☛ *AdminSPA* > `INTERFEJS UŻYTKOWNIKA` > `Listy` > `Edytuj` > `Pomoc dla użytkownika`



Na karcie są dostępne opcje:

- `Zawsze wyświetlaj przycisk pomocy`; przycisk pomocy jest wyświetlany na formularzu/liście niezależnie od tego, czy skonfigurowano jego obsługę.
- `Wyświetlaj tylko, gdy zdefiniowano pomoc`: (ustawienie domyślne) przyciski są wyświetlane, jeśli zdefiniowano akcje wywoływane przyciskiem pomocy.
- `Nigdy nie wyświetlaj`: ukrywa przycisk pomocy na liście/formularzu; ustawienie przydatne np. w przypadku formularza do wyświetlania pomocy.

Więcej informacji zawiera rozdział Definiowanie pomocy w aplikacjach biznesowych

4.3.9. Właściwości listy — karta Eksport listy

> **Podsumowanie:**
> Na tej karcie można określić sposób tworzenia nazwy pliku Excel tworzonego po kliknięciu przycisku eksportu w aplikacji *FrontSPA*.

Na karcie są dostępne następujące opcje w grupie `Pobieraj nazwę eksportowanego pliku z`:

`Kod listy`: nazwa wyeksportowanego pliku ma postać *kodListy.xlsx*.

`Nazwa listy`: nazwa wyeksportowanego pliku ma postać *nazwaListy.xlsx*.

`Zapytanie SQL`: nazwa wyeksportowanego pliku jest zwracana przez zapytanie SQL, które należy wpisać w wyświetlonym oknie edytora; do wyniku zapytania jest dodawane rozszerzenie *.xlsx*.

4.3.10. Kolumna listy — karta Ogólne

> **Podsumowanie:**
> Na tej karcie konfiguruje się ogólne właściwości kolumny listy. Dostępne parametry zależą od rodzaju kolumny oraz typu danych.

Na karcie `Ogólne` można ustawić następujące właściwości kolumny:

Baza danych

- `Nazwa SQL`: (tylko do odczytu) nazwa kolumny w źródle danych listy.
- `Typ danych`: (tylko do odczytu) typ danych kolumny zdefiniowany w źródle danych.

Wyświetlanie kolumny listy

- `Aktywna`: wyłączenie powoduje, że kolumna nie jest wyświetlana w widoku listy.
- `Nazwa kolumny tabeli`: tekst wyświetlany w nagłówku kolumny.
- `Kolejność`: numer kolejny kolumny na liście, licząc od lewej strony; kolejność kolumn można zmieniać poleceniami `Przenieś do góry` i `Przenieś w dół` z menu kontekstowego kolumny na panelu elementów.
- `Szerokość automatyczna kolumny`: włączenie powoduje, że dana kolumna będzie mogła zostać wyświetlona z większą szerokością, aby dostosować wyświetlanie listy do szerokości ekranu; w takiej sytuacji szerokość zadeklarowana w

polu `Szerokość` (`Szerokość minimalna`) będzie traktowana jako szerokość minimalna. Jeśli ten przełącznik nie zostanie włączony dla żadnej kolumny, podczas zapisu widoku listy będzie wyświetlany komunikat z informacją, że szerokości kolumn zostaną przeliczone proporcjonalnie.

- `Szerokość/Minimalna szerokość`: szerokość listy w pikselach; jeśli włączono przełącznik `Szerokość automatyczna`, określa minimalną szerokość kolumny.
- `Format wyświetlania danych`: format danych typów numerycznych oraz daty i godziny. W sekcji Formatowanie danych na listach przedstawiono kilka przykładów formatowania. Pełny wykaz można znaleźć na stronie zewnętrznej po kliknięciu przycisku `Info` obok pola `Format`.
- `Wyświetlanie treści rekordów`: określa sposób wyświetlania w kolumnie wartości, która jest dłuższa niż szerokość kolumny; dostępne wartości:
  - `Skracaj i dopisuj wielokropek`: Wyświetla część wartości, która mieści się kolumnie i dodaje na końcu wielokropek (...), aby wskazać, że wartość została skrócona.
  - `Zawijaj treść`: wyświetla całą wartość w kilku liniach.
- `Wyświetlaj dane jako zbiór wartości`: określa sposób wyświetlania wartości ustawianych za pomocą kontrolki *Checkbox (grupa)*. Po włączeniu w kolumnie są wyświetlane wartości rozdzielane przecinkami bez symbolu *$*.

Operacje na kolumnie listy

- `Sortowanie`: włącza/wyłącza obsługę sortowania indywidualnie według wartości danej kolumny; sortowanie wymaga kliknięcia nagłówka kolumny; pierwsze kliknięcie powoduje sortowanie w kolejności rosnącej, drugie — malejącej, a trzecie wyłącza sortowanie.
- `Filtrowanie`: włącza/wyłącza kontrolki filtrowania dla danej kolumny, to jest pole do wpisania wartości oraz przycisk (`Operator filtrowania`), który pozwala wybrać operator dla wyrażenia filtrowania.
- `Grupowanie`: włącza/wyłącza możliwość grupowania rekordów na liście według wartości danej kolumny; wymaga przeciągnięcia nagłówka kolumny do obszaru bezpośrednio pod panelem przycisków u góry.
- `Metoda wyliczania listy wartości`: określa sposób wyliczania listy/zakresu wartości na potrzeby filtrowania rekordów w widoku listy; to ustawienie jest niedostępne dla kolumny typu *uniqueidentifier*; dostępne wartości:
  - `Dynamiczne`: umożliwia filtrowanie rekordów widoku listy według wartości występujących w danej kolumnie; użytkownik zaznacza pola wyboru obok wartości, które chce wyświetlić; dodatkowo są dostępne pozycje `Zaznacz wszystko` (zaznacza pola wyboru obok wszystkich wartości, umożliwiając

łatwe wykluczenie pojedynczych wartości) i `Puste` (po zaznaczeniu wyświetlane są tylko rekordy, które mają w tej kolumnie wartość *null*); zależnie od typów danych dostępne są opcje umożliwiające określenie np. zakresu wartości, predefiniowanych okresów, itp.

- `Przez słownik`: umożliwia filtrowanie rekordów według wartości ze słownika wybranego z listy rozwijanej; wzorzec wyszukiwania zależy od formatu pozycji słownikowej określonego po kliknięciu przycisku `Format`; metoda dostępna tylko dla typów danych *varchar* i *nvarchar*.

  > **UWAGA**
  > Typ danych kolumn, którym przypisano słownik w kreatorze modeli danych, jest w funkcji źródłowej widoku listy zamieniany na *nvarchar*, ponieważ w widoku listy w tej kolumnie jest wyświetlana wartość tekstowa. Ta konwersja ma miejsce w przypadku list generowanych poleceniem `Generuj elementy aplikacji`. W przypadku przypisania słownika na późniejszym etapie konieczne jest ręczne odświeżenie typu kolumny, aby wartości słownikowe były poprawnie wyświetlane na liście.

  - `Słownik dla listy`: wybierz słownik, do użycia w filtrze
  - przycisk z kluczem otwiera okno dialogowe, w którym można określić format wyświetlania wpisów ze słownika. Format określa się na liście wielowyboru z możliwością edycji, wybierając predefiniowane atrybuty wpisu do słownika, w tym *code*, *name*, *description*, *numericDescription* i inne, oraz wpisując dodatkowe znaki i/lub tekst, które poprawią czytelność wyświetlanych danych (np. nawiasy, myślnik, dwukropek, ukośnik itp.).
  - `Tylko aktywne pozycje`: w okienku filtrowania dostępne będą tylko aktywne pozycje słownika
  - `Wszystkie pozycje`: w okienku filtrowania dostępne będą wszystkie pozycje słownika (aktywne i nieaktywne).
- `Zaawansowany`: po kliknięciu nagłówka kolumny w widoku listy wyświetla okienko, w którym można skonstruować kryterium filtrowania w formie *operator wartość and/or operator wartość*; dostępne operatory i sposób wyboru wartości różni się, zależnie od typu danych kolumny; jest to domyślny i jedyny sposób filtrowania wartości w kolumnie typu *uniqueidentifier*.
- `Zapytanie SQL`: wyświetla okno edytora SQL w celu zdefiniowania zapytania zwracającego listę wartości filtrujących.

Ustawienia stopki

- `Funkcja stopki`: funkcja agregująca używana w stopce; wartość domyślna `Brak`; dostępne funkcje (odpowiednio do typu danych kolumny):
    - `Suma`: zwraca sumę wartości w kolumnie (dla numerycznych typów danych).
    - `Średnia`: oblicza wartość średnią z wartości w kolumnie (dla numerycznych typów danych).
    - `Liczba`: zwraca liczbę rekordów z wartością w danej kolumnie (dla wszystkich typów danych).
    - `Max`: zwraca największą wartość w kolumnie (dla numerycznych typów danych).
    - `Min`: zwraca najmniejszą wartość w kolumnie (dla numerycznych typów danych).

> **UWAGA**
>
> - Wartość funkcji w stopce jest wyliczana dla całego zestawu rekordów, z uwzględnieniem filtrowania i sortowania, bez uwzględniania grupowania i paginacji.
> - Rekordy z brakiem wartości w danym polu (*null*) nie są uwzględniane w wyliczanej wartości.

- `Treść stopki`: tekst wyświetlany w stopce; aby wyświetlić wartość funkcji wybranej z listy `Funkcja stopki`, należy użyć zmiennej `{@FunctionResult}`.

FORMATY DANYCH TYPU DATA/GODZINA

Parametr formatu	Alternatywna składnia	Opis formatu	Przykład
`d`	`d.M.y`	Data krótka	7.6.2021
`D`	`EEEE, d MMMM y`	Data długa	poniedziałek, 7 czerwca 2021
`F`	`EEEE, d MMMM y HH:mm`	Pełna data i godzina	poniedziałek, 7 czerwca 2021 16:34
`g`	`d.M.y H:mm`	Krótka data i godzina	7.6.2021 16:34
`G`	`d.M.y HH:mm:ss`	Krótka data i pełna godzina	7.6.2021 16:34:18
`M`	`d MMMM`	Dzień i nazwa miesiąca	7 czerwca
`m`	`d MMM`	Dzień i skrót miesiąca	7 cze
`y`	`MMM y`	Skrót miesiąca i rok	cze 2021
`t`	`HH:mm`	Krótka godzina	16:34
	`hh:mm`	Krótka godzina, zegar 12 godzinny	04:34
`T`	`HH:mm:ss`	Pełna godzina	16:34:18
`s`	`"yyyy-MM-dd HH:mm:ss`	Uniwersalny, sortowalny format daty i godziny	2021-06-07 16:34
`u`	`"yyyy-MM-dd HH:mm:ss Z`	Uniwersalny, sortowalny format daty i godziny UTC	2021-06-07 16:34 +0100

Formaty danych typów numerycznych

Parametr formatu	Alternatywna składnia	Opis formatu	Przykład
`n`		Formatuje wartości numeryczne jako liczby dziesiętne z trzema miejscami po przecinku	`n` — 1 234 567,890
			`n2` — 1 234 567,89
`c`		Formatuje wartości numeryczne jako kwoty, zgodnie z ustawieniami narodowymi	1 234 567,89 zł
`p`		Formatuje wartości numeryczne jako wartości procentowe, domyślnie bez miejsc po przecinku	`p` — 37%
			`p1` — 37,5%

4.3.11. Kolumna listy — karta Wygląd

Podsumowanie:
Na karcie `Wygląd` można ustawiać indywidualnie właściwości wyglądu dla kolumn, nadpisując właściwości określone globalnie dla wszystkich kolumn listy.

Właściwości wyglądu podzielono na trzy karty podrzędne odpowiadające sekcjom listy: `Nagłówek`, `Wiersze` i `Stopka`, przy czym te karty są wyświetlane dopiero wtedy, gdy zostanie włączony przełącznik nadpisywania ustawień dla odpowiedniej sekcji.

Na karcie `Wygląd` można ustawić następujące właściwości kolumny:

- `Nadpisz ustawienia wyglądu nagłówka`: pozwala ustawić właściwości wyglądu nagłówka bieżącej kolumny inne niż dla pozostałych kolumn na liście; po włączeniu zostaje wyświetlona karta podrzędna `Nagłówek`, na której można zmieniać właściwości wyglądu.
- `Nadpisz ustawienia wyglądu grida`: pozwala ustawić właściwości sekcji grid dla bieżącej kolumny inne niż dla pozostałych kolumn na liście; po włączeniu zostaje wyświetlona karta podrzędna `Grid`, na której można zmieniać właściwości wyglądu.
- `Nadpisz ustawienia wyglądu stopki`: pozwala ustawić właściwości wyglądu stopki bieżącej kolumny inne niż dla pozostałych kolumn na liście; po włączeniu zostaje wyświetlona karta podrzędna `Stopka`, na której można zmieniać właściwości wyglądu.

Na kartach `Nagłówek`, `Wiersze` i `Stopka` są dostępne następujące ustawienia:

- `Rozmiar czcionki (rem)`: rozmiar czcionki w jednostkach rem (względem czcionki głównego elementu html); wartość domyślna *1*.
- `Tło`: kolor tła wiersza; ustawienie dostępne dla nagłówka i stopki.
- `Kolor czcionki`: kolor czcionki; ustawienie dostępne dla nagłówka i stopki.
- `Tło (wiersz parzysty)`: kolor tła wierszy parzystych (tylko sekcja `Wiersze`).
- `Tło (wiersz nieparzysty)`: kolor tła wierszy nieparzystych; (tylko sekcja `Wiersze`).
- `Kolor czcionki (wiersz nieparzysty)`: kolor czcionki w wierszach nieparzystych (tylko karta `Wiersze`).
- `Kolor czcionki (wiersz parzysty)`: kolor czcionki w wierszach parzystych (tylko karta `Wiersze`).
- `Pogrubienie czcionki`: włącza pogrubienie czcionki używanej w stopce listy.
- `Pochylenie czcionki`: włącza pochylenie czcionki używanej w stopce listy.
- `Wyrównanie`: określa wyrównanie wartości w danej sekcji listy; dostępne wartości: `do lewej`, `do środka`, `do prawej`.

WSKAZÓWKA
W przypadku kolumn specjalnych (akcje, checkbox, HTML, właściciel, termin) ich właściwości mogą mieć priorytet przed ustawieniami globalnymi, np. ustawienia koloru tła dla wierszy parzystych i nieparzystych.

- `Margines wewnętrzny lewy/prawy`: określa w pikselach *padding* w poziomie (od lewej i od prawej krawędzi kontrolki).

Powiązane tematy:
- Właściwości listy - karta Wygląd

4.3.12. Kolumna listy — karta Widoczność

Podsumowanie:
Ta karta służy do konfigurowania warunków wyświetlania danej kolumny. Warunek definiuje się w formie zapytania SQL, które musi zwrócić wartość liczbową. Aby ukryć kolumnę, zapytanie musi zwrócić *0*. Warunek widoczności można zdefiniować dla każdej kolumny listy, w tym dla kolumny akcji (*command*).

W warunku widoczności można używać następujących zmiennych:

- *@_LangId*: identyfikator języka zalogowanego użytkownika.
- *@_UserId*: identyfikator zalogowanego użytkownika.
- *@_BusinessDocDefId*: identyfikator dokumentu biznesowego, z którym jest powiązana lista.
- *@_GridId*: identyfikator listy.
- *@_MobileMode*: tryb pracy aplikacji; w aplikacji mobilnej przyjmuje wartość "FLUTTER" (nie dotyczy trybu PWA), w pozostałych przypadkach przyjmuje wartość pustego ciągu znaków ("").

Obsługiwane są również odwołania do pola formularza w formie *{@*KodFormularza.PoleFormularza}*. Pozwala to sterować widocznością kolumn w listach osadzonych na formularzu, zależnie od wartości w polach formularza.

> **UWAGA**
> Odwołanie do pola formularza można stosować wyłącznie w listach osadzonych na formularzu. Próba wyświetlenia widoku listy zawierającej takie odwołanie spowoduje, że wszystkie kolumny z ustawionym warunkiem widoczności zostaną ukryte.

Zdefiniowane zapytanie zostanie opakowane w inne zapytanie, dlatego można w nim używać wyłącznie konstrukcji SQL obsługiwanych w podzapytaniach. W szczególności nie można używać instrukcji *IF-ELSE* (zamiast niej można użyć *CASE WHEN*), *FOR JSON PATH*, pętli itp.

Przykłady zapytań:

```
SELECT CASE {@_MobileMode}
       WHEN 'FLUTTER'
       THEN 1
       ELSE 0
       END
```

```
SELECT COUNT(*)
FROM [dbo].[Maszyny]
WHERE [DataZlomowania] IS NOT NULL
```

4.3.13. Kolumna listy — karta Eksportowanie danych

> **Podsumowanie:**
> Na tej karcie należy zdefiniować format danych z bieżącej kolumny na potrzeby eksportu do pliku Excel. Zapewni to prawidłową identyfikację formatu eksportowanych danych w programie Excel.

Na karcie `Eksportowanie danych` są dostępne następujące opcje:

`Liczbowy (dokładność do setnych)`: eksportuje wartość jako liczbę i wyświetla z dwoma miejscami po przecinku.

`Procentowy (dokładność do całości)`: eksportuje wartość i wyświetla jako procenty bez miejsc po przecinku.

`Tekstowy`: eksportuje wartość jako tekst.

`Data (dd.MM.rrrr)`: eksportuje wartość jako datę w podanym formacie.

`Czas (gg:mm:ss)`: eksportuje wartość jako czas w podanym formacie.

`Ogólny`: eksportuje wartość w formacie ogólnym (ustawienie domyślne).

`Niestandardowy`: wymaga podania schematu niestandardowego formatu zgodnie z konwencją programu Excel, np *0* to format liczbowy bez miejsc po przecinku odpowiedni dla liczb całkowitych. W przypadku formatowania daty i czasu należy używać skrótów w wersji angielskojęzycznej, na przykład *dd.mm.yyyy hh:mm* zamiast *dd.mm.rrrr gg:mm*.

4.3.14. Kolumna listy — karta Edycja inline

> **Podsumowanie:**
> Ta karta jest dostępna we właściwościach kolumny, jeśli we właściwościach listy włączono edycję inline. Na tej karcie można skonfigurować ustawienia dotyczące pola kolumny wyświetlanej w trybie edycji inline, takie jak rodzaj kontrolki, wartość domyślna, ustawienia dostępności i wymagalności oraz komunikat o błędzie wyświetlany, jeśli pole wymagane nie zostanie wypełnione. W przypadku wyboru jako kontrolki zdefiniowanej osobno listy wyboru można także określić mapowanie wartości z kolumn zwracanych przez tę kontrolkę do kolumn widoku listy.

Na karcie można określić następujące właściwości:

- `Typ pola`: wybierz kontrolkę wyświetlaną w danej kolumnie podczas edycji inline; kontrolka powinna odpowiadać wartości w polu `Typ danych` powyżej; dostępne

wartości:

- `Pole tekstowe`
- `Lista wyboru (SQL)`: patrz opis dodatkowej konfiguracji poniżej
- `Pole numeryczne`
- `Checkbox pojedynczy`
- `Datepicker`
- `Datepicker i pole czasu`
- `Pole czasu`
- `Pole daty`

> **WSKAZÓWKA**
>
> - Kontrolka do ustawiania czasu (autonomiczna i w datepickerze) umożliwia ustawianie tych jednostek, które zostaną określone we właściwości `Format`. Na przykład dla formatu *HH:mm:ss*, możliwe będzie ustawienie godzin, minut i sekund.
> - W edycji inline można wprowadzać daty z zakresu od *01-02-1900* do *31-12-9999*, natomiast w formularzu można zapisywać daty od *01-01-1753*. Jeśli edytowany inline rekord będzie zawierał datę spoza zakresu (wcześniejszą niż minimalna), nie będzie można go zapisać.

- `Wartość domyślna`: wartość domyślna wyświetlana w tej kolumnie w nowym rekordzie; w wyrażeniu można odwołać się do wartości z innej kolumny listy (*{@NazwaSQL}*); w przypadku listy osadzonej na formularzu obsługiwane jest odwołanie się do wartości pola formularza w postaci *{@*KodFormularza.PoleFormularza}*.
- `Wartość minimalna/maksymalna`: minimalna/maksymalna wartość, jaką użytkownik może wpisać w danej komórce; dotyczy tylko kontrolek dla typów liczbowych oraz daty/godziny. Wpisanie wartości spoza zakresu spowoduje podświetlenie komórki na czerwono oraz wyświetlenie informacji o błędzie.
- `Długość minimalna/maksymalna`: minimalna/maksymalna liczba znaków tekstu, jaki użytkownik może wpisać w danej komórce; dotyczy tylko kontrolki typu pole tekstowe. Wpisanie wartości spoza zakresu spowoduje podświetlenie komórki na czerwono oraz wyświetlenie informacji o błędzie.

> **UWAGA**
> W przypadku niezgodności typu kontrolki i typu danych kolumny, np. użycia pola tekstowego do wprowadzania liczby lub daty, walidacja jest pomijana.

- `Tylko do odczytu`: wyłącza możliwość edycji inline w danej kolumnie.

- `Wymagane uzupełnienie komórki`: włączenie tego przełącznika spowoduje, że bez wpisania wartosci w tym polu rekordu, nie będzie można go zapisać.
- `Komunikat o błędzie`: tekst wyświetlany w przypadku, gdy wymagane pole rekordu nie zostanie wypełnione.

Aby korzystać z listy wyboru do edycji inline, należy określić dodatkowe ustawienia:

- `Zdefiniowana lista wyboru`: wybierz predefiniowaną listę wyboru (`INTERFEJS UŻYTKOWNIKA` > `Kontrolki`).
- `Lista zależności kolumna - kolumna`: umożliwia wypełnianie innych pól w bieżącym rekordzie danymi zwracanymi przez zapytanie źródłowe używanej listy wyboru.

  > **UWAGA**
  > Można definiować zależności do innych kolumn widoku listy, w których używane są kontrolki listy rozwijanej. W takim przypadku należy przekazać wartość odpowiadającą wartości *[key]* z kontrolki w kolumnie zależnej. Taka konfiguracja będzie działać poprawnie, to znaczy wybór wartości w kontrolce nadrzędnej spowoduje prawidłowe ustawienie wartości w kontrolce zależnej. Nie zaleca się natomiast definiowania zależności między jedną z kolumn zwracanych przez kontrolkę, a kolumną widoku listy, w której znajduje się ta kontrolka.

- `Tryb aktualizacji pól zależnych`: dostępne wartości:
  - `Tylko po zmianie na liście wyboru`: aktualizacja wartości w polach zależnych odbywa się tylko po zmianie wartości na liście wyboru.
  - `Tylko gdy wskazane pole jest puste`: wartości zwrócone przez listę wyboru są przekazywane do pól zależnych tylko wtedy, kiedy te pola są puste.

KONTROLKA LISTY WYBORU

W podstawowym scenariuszu zakłada się, że lista wyboru zostanie użyta w kolumnie pomocniczej używanej do wyświetlania danych, a w tabeli modelu danych będzie zapisywana wartość klucza zwracana przez listę. Aby skonfigurować korzystanie z listy wyboru podczas edycji inline w tym scenariuszu, wykonaj następujące czynności:

1. Zdefiniuj kontrolkę listy wyboru (*AdminSPA* > `Interfejs użytkownika` > `Kontrolki`):

```
SELECT [Id] as [key],
    CONCAT(FirstName, ' ', Lastname) as [value],
```

```
    [Email] as [useremail]
FROM [core].[UserProfiles]
```

2. Dodaj pomocniczą kolumnę w zapytaniu źródłowym widoku listy:

```
SELECT v.*, CONCAT(up.FirstName, ' ', up.Lastname) as UserFullName
FROM [mytable_View]({@_LangId}) v
LEFT JOIN core.UserProfiles up ON up.Id = v.EmployeeId
```

> **WAŻNE**
> Zwróć uwagę, aby wartości zwracane w kolumnie *[value]* były identyczne jak wartości w kolumnie dodawanej do źródła widoku listy. W przeciwnym razie kontrolka nie będzie działać poprawnie.

3. W kolumnie pomocniczej (w przykładzie *UserFullName*) użyj zdefiniowanej listy wyboru jako kontrolki do edycji inline i zdefiniuj zależność, aby wstawić wartość *key* zwracaną przez kontrolkę do odpowiedniej kolumny listy (w przykładzie *[key]* -> *EmployeeId*). Ustawienie zależności jest konieczne tylko wtedy, gdy kolumna docelowa jest widoczna i użytkownik powinien widzieć podczas edycji inline, że wybór wartości z listy wyboru powoduje zmianę w kolumnie docelowej zależności. Jednak w takim przypadku w akcji zapisu wartości w tej kolumnie należy odwołać się do do nazwy kolumny z kontrolką, dodając przyrostek *_key*.
4. Odpowiednio zdefiniuj akcję zapisu inline, aby zapisywała dane z kolumny klucza obcego, np.

```
IF ({@Id} IS NULL)
  INSERT INTO [dbo].[MyTable] (Column1, Column2, EmployeeId)
  VALUES ({@Column1},{@Column2}, {@EmployeeId})
ELSE
  UPDATE [dbo].[MyTable]
  SET Column1 = {@Column1},
   Column2 = {@Column2},
   EmployeeId = {@EmployeeId}
  WHERE Id = {@Id}
```

Alternatywnie w akcji zapisu można użyć podstawienia *EmployeeId = {@UserFullName_key}*, aby odwołać się bezpośrednio do wartości z kolumny *[key]* zwróconej przez listę wyboru.

EDYCJA INLINE W KOLUMNACH ZE SŁOWNIKIEM

Listy wyboru można także użyć do zasilania kolumny typu *int*, której w kreatorze modeli danych przypisano słownik. W takim przypadku w definicji widoku listy jest dostępna kolumna o takiej samej nazwie, tylko typu *varchar*. W tej kolumnie jest wyświetlana wartość słownikowa wybierana na formularzu. Konfiguracja możliwości edycji inline tej wartości wymaga wykonania następujących kroków:

1. W definicji źródła danych widoku listy dodaj oryginalną kolumnę z tabeli (typu *int*):

```
SELECT v.*,
mt.MyKey AS MyKeyColumn
FROM [MyTable_View]({@_LangId}) v
INNER JOIN [dbo].[MyTable] mt ON v.Id = mt.Id
```

2. Zdefiniuj kontrolkę, która będzie zwracać odpowiednie wartości słownikowe:

```
SELECT dv.Id as [key], t.Value as [value]
FROM core.TranslationDefinitions td
INNER JOIN core.Translations t ON td.Id = t.TranslationDefinitionId
INNER JOIN core.DictionaryValues dv ON td.RecordRowGuid = dv.RowGuid
WHERE [DictionaryId] = {&MyApp.Dictionaries.MyDict} AND LanguageId = {@_LangId}
```

3. W definicji widoku listy, we właściwościach kolumny z wartością słownikową dodaj tę kontrolkę dla trybu edycji inline i zdefiniuj mapowanie klucza kontrolki na dodaną kolumnę typu *int* (w powyższym przykładzie *[key] -> MyKeyColumn*).
4. Zdefiniuj akcję zapisu inline, odpowiednio zestawiając kolumnę docelową w tabeli danych z kolumną widoku, w której jest ustawiany klucz wartości słownikowej.

PROSTE UŻYCIE LISTY WYBORU W EDYCJI INLINE

W prostym scenariuszy można używać listy wyboru w trybie edycji inline do zapisywania wyświetlanych wartości w kolumnie, w której jest osadzona. W tym celu należy zdefiniować kontrolkę i ustawić ją w kolumnie typu *varchar*. Bez żadnej dodatkowej konfiguracji i dla standardowej akcji zapisu typu:

```
UPDATE [dbo].[MyTable]
SET MyColumn = {@MyColumn}
WHERE Id = {@Id}
```

gdzie *MyColumn* to kolumna typu *varchar* z ustawioną listą wyboru, w tej kolumnie będą zapisywane wartości tekstowe wybierane z listy wyboru.

W takim przypadku także możliwe jest definiowanie zależności między kolumnami zwracanymi przez kontrolkę a innymi kolumnami widoku listy.

Powiązane tematy:

- Właściwości kontrolek (patrz Wartość domyślna)
- Przyciski listy do edycji inline

4.3.15. Kolumna HTML

Podsumowanie:
Kolumna HTML to specjalny rodzaj kolumny, którą można dodać do widoku listy. W komórkach takiej kolumny można renderować markup HTML zdefiniowany na kartach właściwości kolumny. W kodzie HTML można odwoływać się do wartości z pozostałych kolumn, używając składni *{@nazwakolumny}*. Funkcje sortowania, filtrowania i grupowania dla listy HTML wymagają podania nazwy kolumny źródłowej (tylko jednej), której wartości będą uwzględniane.

Kolumna HTML jest obsługiwana także w aplikacji mobilnej.

W TYM TEMACIE

- Karta Ogólne
- Karta Formatowanie, podkarta Stylizacja
- Karta Formatowanie, podkarta Formatowanie warunkowe

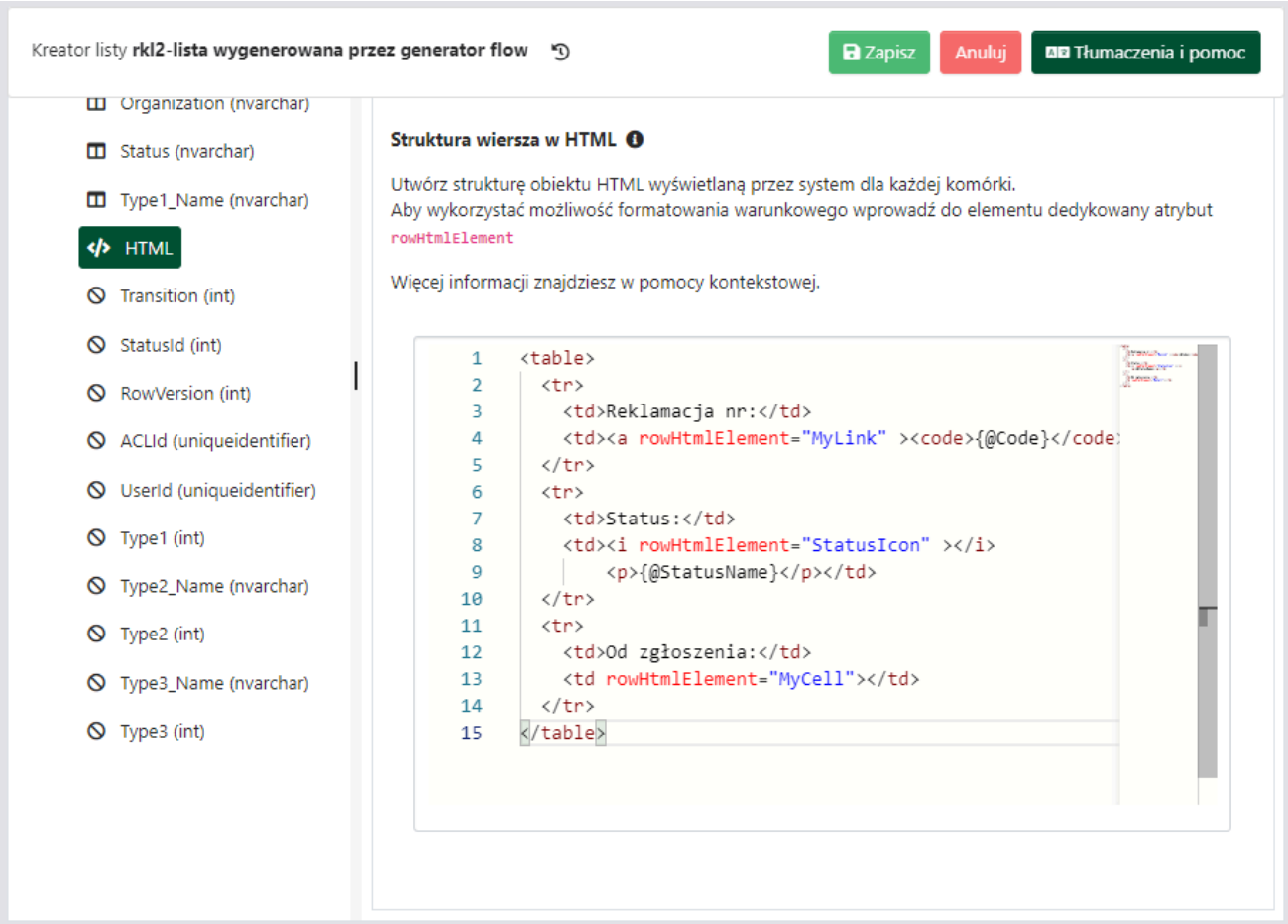
Kolumna HTML może agregować informacje z innych kolumn w widoku listy, co ma szczególne zastosowanie w urządzeniach mobilnych. Poniższa ilustracja przedstawia kolumnę HTML, w której zamieszczono wartości z innych kolumn w danym rekordzie. Dodatkowo zastosowano formatowanie warunkowe, zależnie od wartosci z innych kolumn lub wartości wyrażeń wyliczanych na podstawie wartości z innych kolumn. Konfigurację właściwości kolumny opisano w przykładach zamieszczonych w tym temacie.

Widok listy ograniczony do kolumny HTML.

INFO

W odwołaniach do stylu — w markupie HTML, kodzie Javascript i definicjach CSS można korzystać z selektorów platformy Bootstrap 5 (https://getbootstrap.com/docs/5.0/).

KARTA OGÓLNE



Kolumna HTML, karta `Ogólne`.

Oprócz takich samych właściwości jak inne kolumny, kolumna HTML ma na karcie `Ogólne` właściwości:

- `Kolumna wiodąca`: wybierz z listy rozwijanej kolumnę, której wartości zostaną użyte do sortowania, filtrowania i/lub grupowania rekordów w widoku listy w przypadku użycia tych funkcjonalności dla kolumny HTML; kolumna HTML nie musi zawierać odwołania do wartosci z kolumny wiodącej.
- `Struktura wiersza w HTML`: kod HTML, który zostanie wyrenderowany w każdym wierszu widoku listy w kolumnie HTML; kolumna pozwala prezentować wartości z innych pól danego rekordu (w tym także z pól ukrytych), formatować je (w tym stosować formatowanie warunkowe), wyświetlać ikony, obrazy itp.; na przykład:

  ```
  <i class="fas fa-check" style="font-size:36px;color:lightblue;">
  {@StatusName}
  </i>
  ```

  W odwołaniu do wartości z pola rekordu należy używać składni *{@NazwaSQL}*, gdzie *Nazwa SQL* to nazwa kolumny z pola `Nazwa SQL` na karcie `Ogólne`.

PRZYKŁAD

W każdej komórce kolumny HTML będzie renderowana tabela złożona z dwóch kolumn i trzech wierszy. W kolumnie z lewej strony znajdują się opisy wartości wyświetlanych w sąsiedniej kolumnie. Są to:

- `Nr reklamacji`: wartość z kolumny *Code* w formie linku, którego kliknięcie wyświetli formularz z danymi reklamacji. Adres URL (wartość atrybutu *href*) jest ustawiany przez instrukcję Javascript.
- `Status`: ikona i pomocniczo nazwa statusu dokumentu; ikona jest ustawiana instrukcją JavaScript, zależnie od wartości identyfikatora statusu.
- `Dni od zgłoszenia`: liczba dni, które upłynęły od utworzenia dokumentu (*CreateDate*) do daty bieżącej; wartość wyliczana instrukcją SQL; dodatkowo, kiedy ta wartość jest większa od 14 stosowany jest skrypt zdefiniowany na karcie `Formatowanie warunkowe`, podkarta `Stylizacja`.

```
<table>
  <tr>
    <td>Reklamacja nr:</td>
    <td><a rowHtmlElement="MyLink" ><code>{@Code}</code></a></td>
  </tr>
  <tr>
    <td>Status:</td>
    <td><i rowHtmlElement="StatusIcon" ></i>
        <p>{@StatusName}</p></td>
  </tr>
  <tr>
    <td>Od zgłoszenia:</td>
    <td rowHtmlElement="MyCell"></td>
  </tr>
</table>
```

KARTA FORMATOWANIE, PODKARTA STYLIZACJA

Na karcie `Stylizacja` można zdefiniować style CSS, które zostaną użyte do formatowania markupu HTML zdefiniowanego na karcie `Ogólne`. Definicje stylów mają charakter lokalny, to jest są stosowane tylko w obrębie bieżącej kolumny HTML.

Ogólne | Wygląd | Widoczność | Formatowanie warunkowe

Ustawienia formatowania

Formatowanie warunkowe pozwala na zdefiniowanie stylu, który system ma użyć do oznaczenia treści komórek spełniających założone kryteria.

Warunki formatowania | Stylizacja

Skonfiguruj własne style z użyciem składni CSS. Zdefiniowane style będą stosowane tylko w obrębie tej kolumny.

Więcej informacji znajdziesz w pomocy kontekstowej.

```
1   td:nth-child(odd) {
2     background-color: #85cc98;
3     text-align: right;
4     vertical-align: top;
5   }
6   td:nth-child(even) {
7     background-color: #a7d9b4;
8     text-align: center;
9     column-width: 200px;
10    max-width: 200px;
11  }
12
13  /* Styl stosowany przez instrukcję JavaScript
14  po spełnieniu warunku: */
15
16  .warn14 {
17    color: orangered;
18    font-size: 1.4rem;
19  }
```

Kolumna HTML, karta `Formatowanie warunkowe, Stylizacja`.

PRZYKŁAD

Formatowanie kolumn tabeli renderowanej w komórkach kolumny html:

```
td:nth-child(odd) {
  background-color: #85cc98;
```

```
  text-align: right;
  vertical-align: top;
}
td:nth-child(even) {
  background-color: #a7d9b4;
  text-align: center;
  column-width: 200px;
  max-width: 200px;
}
```

Styl stosowany przez instrukcję JavaScript po spełnieniu warunku:

```
.warn14 {
  color: orangered;
  font-size: 1.4rem;
}
```

KARTA FORMATOWANIE, PODKARTA FORMATOWANIE WARUNKOWE

Na karcie `Warunki formatowania` należy wpisać instrukcje JavaScript formatujące elementy w kolumnie HTML, w zależności od podanych warunków. Instrukcje JavaScript mogą odwoływać się do wartości z innych kolumn w danym rekordzie przy użyciu składni *RowDataContext.NazwaKolumny*, gdzie *NazwaKolumny* to wartość właściwości `Nazwa SQL` danej kolumny, czyli nazwa kolumny w źródle danych listy. Z kolei odwołania do formatowanych elementów HTML muszą mieć postać: *RowHtmlElements.NazwaElementu*, gdzie *NazwaElementu* to wartość atrybutu o tej nazwie zdefiniowanego w oknie edytora HTML na karcie `Ogólne`.

Ogólne | Wygląd | Widoczność | Formatowanie warunkowe

Ustawienia formatowania

Formatowanie warunkowe pozwala na zdefiniowanie stylu, który system ma użyć do oznaczenia treści komórek spełniających założone kryteria.

Warunki formatowania | Stylizacja

Użyj składni JavaScript aby zdefiniować warunek formatowania.

- W warunku formatowania dostęp do danych opisujących rekord uzyskasz wyłuskując wybrany atrybut z obiektu `RowDataContext`.
- W poleceniu warunku dostęp do atrybutów HTML ustawionych dla elementów uzyskasz wyłuskując je z obiektu `RowHtmlElements`.
- Do wyboru stylu możesz posłużyć się nazwami klas z biblioteki Bootstrap 5 lub zdefiniować własne klasy w zakładce Stylizacja.

Więcej informacji znajdziesz w pomocy kontekstowej.

```
1   var
2   anchor = 'http://localhost:4200/front/form/59CEB6A4-
3            + RowDataContext.Id
4
5   RowHtmlElements.MyLink.href = anchor;
6
7   switch (RowDataContext.Status) {
8     case 1:
9       RowHtmlElements.StatusIcon.className =
10          'fas fa-pen fa-4x text-success';
11      break;
12    case 30:
13      RowHtmlElements.StatusIcon.className =
```

Kolumna HTML, karta `Formatowanie warunkowe, Warunki formatowania`.

PRZYKŁAD

Link do edycji dokumentu

W markupie HTML zdefiniowano element *<a>* z atrybutem *rowHtmlElement* o wartości *MyLink*. Poniższa instrukcja ustawia dla tego elementu wartość atrybutu *href*, dodając do adresu URL formularza identyfikator dokumentu. Jako tekst linku wyświetlany jest numer dokumentu (wartość z kolumny *Code*).

```
var
anchor = 'http://localhost:4200/front/form/59CEB6A4-0DAD-4E0D-9C0D-E017357CDE12/'
          + RowDataContext.Id

RowHtmlElements.MyLink.href = anchor;
```

Wybór ikony w zależności od statusu

W markupie HTML zdefiniowano element *<i>* z atrybutem *rowHtmlElement* o wartości *StatusIcon*. Poniższa instrukcja ustawia dla tego elementu nazwę klasy (ikonę z zestawu Font Awesome) odpowiednio do wartości z kolumny *Status* (liczba całkowita identyfikująca status dokumentu).

```
switch (RowDataContext.Status) {
  case 1:
    RowHtmlElements.StatusIcon.className =
                          'fas fa-pen fa-4x text-success';
    break;
  case 30:
    RowHtmlElements.StatusIcon.className =
                           'fas fa-pen fa-4x text-success';
    break;
  case 2:
    RowHtmlElements.StatusIcon.className =
                            'fas fa-hourglass fa-4x text-info';
    break;
  case 3:
     RowHtmlElements.StatusIcon.className =
                            'far fa-hourglass fa-4x text-warning';
    break;
  case 4:
    RowHtmlElements.StatusIcon.className =
                              'fas fa-check fa-4x text-success';
    break;
  case 5:
    RowHtmlElements.StatusIcon.className =
                               'fas fa-ban fa-4x text-danger';
    break;
  case 6:
    RowHtmlElements.StatusIcon.className =
                                'fas fa-ban fa-4x';
    break;
```

```
  case 7:
    RowHtmlElements.StatusIcon.className =
                                'fas fa-check fa-4x';
};
```

Liczba dni od utworzenia dokumentu

W markupie HTML zdefiniowano element *<td>* z atrybutem *rowHtmlElement* o wartości *MyCell*. Poniższa instrukcja ustawia dla tego elementu wartość (liczbę dni, które upłynęły od utworzenia dokumentu) i wartość atrybutu *class*, jeśli liczba dni jest większa od 14.

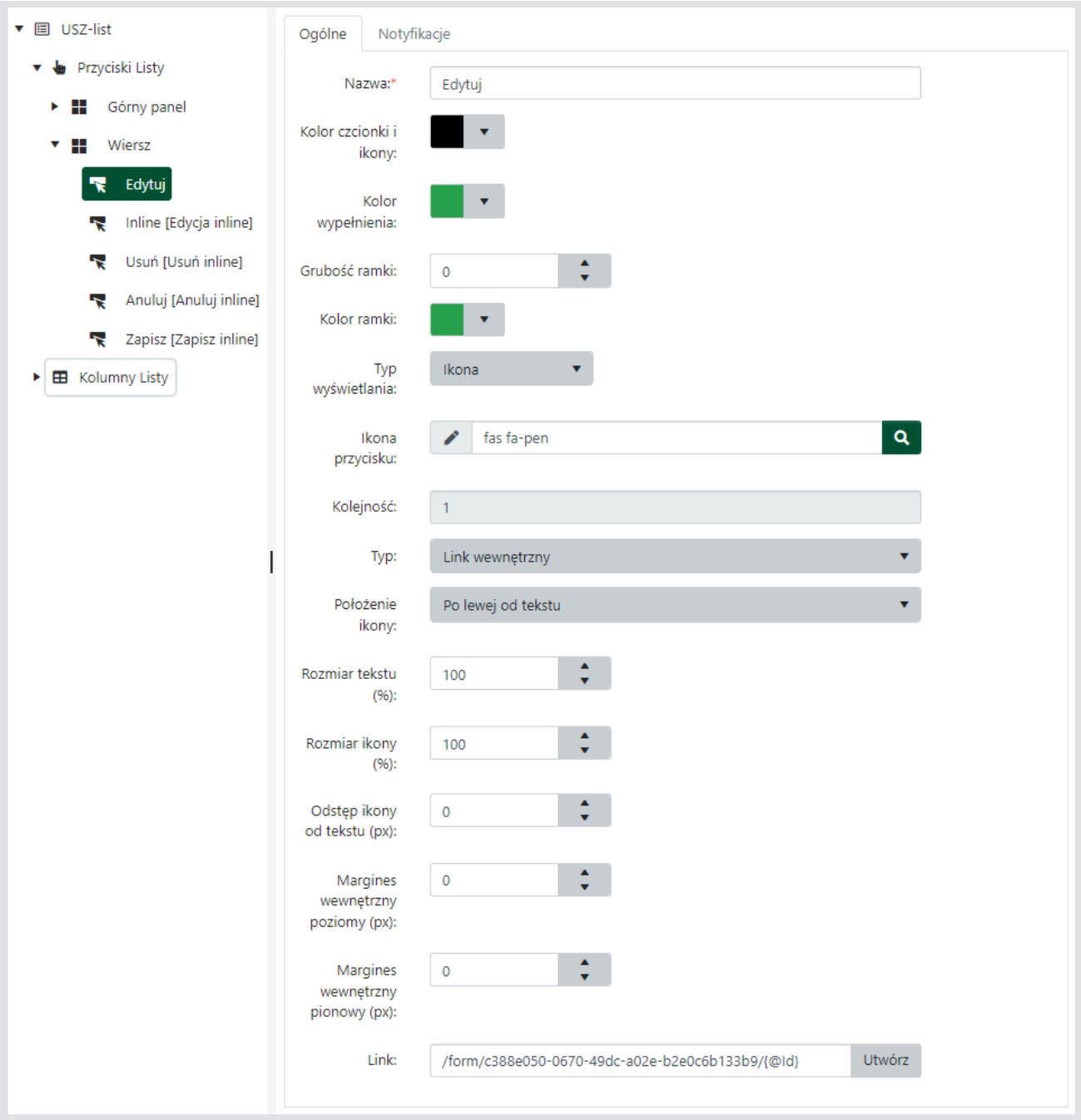
```
var dzisiaj = new Date();
var utworzono = new Date(RowDataContext.CreateDate).setHours(0);
var dni = Math.floor(Math.abs(dzisiaj-utworzono)/(1000 * 3600 * 24));

RowHtmlElements.MyCell.innerHTML = dni + ' dni';

if (dni > 14 ) {
    RowHtmlElements.MyCell.className = 'warn14';
};
```

4.3.16. Właściwości przycisku — karta Ogólne

> **Podsumowanie:**
> Przyciski w widoku listy mogą znajdować się w sekcji `Górny panel` (tam gdzie standardowo znajduje się przycisk `Nowy dokument`) oraz w poszczególnych wierszach listy, w kolumnie akcji (`command`). Do tych przycisków można przypisywać różne akcje i zestawy akcji.

Aby wyświetlić właściwości przycisku, rozwiń odpowiednią sekcję (`Górny panel` lub `Wiersz`) gałęzi `Przyciski listy` na panelu elementów i kliknij nazwę przycisku. Właściwości zostaną wyświetlone na panelu właściwości, na kartach `Ogólne` i `Notyfikacje`.

Aby dodać przycisk, kliknij nagłówek sekcji `Górny panel` lub `Wiersz` i wybierz polecenie `Dodaj nowy przycisk`.



Właściwości przycisku, karta `Ogólne`.

Na karcie `Ogólne` znajdują się następujące ustawienia:

- `Nazwa`: tekst wyświetlany na przycisku.
- `Kolor czcionki i ikony`: określa kolor ikony (o ile zostanie wybrana) i tekstu etykiety przycisku.
- `Kolor wypełnienia`: kolor tła przycisku.
- `Grubość ramki`: grubość obramowania przycisku w pikselach.
- `Kolor ramki`: kolor obramowania przycisku.
- `Rodzaj wyświetlania`: określa sposób wyświetlania przyciski; dostępne wartości:

    - `Ikona i tekst`: wyświetla na przycisku ikonę i tekst (z pola `Nazwa`).
    - `Ikona`: wyświetla na przycisku ikonę.
    - `Tekst`: wyświetla na przycisku tekst.

- `Ikona przycisku`: pozwala wybrać ikonę przycisku; kliknięcie przycisku (`Szukaj`) otwiera okno dialogowe wyboru ikony.
- `Kolejność`: kolejność przycisku na górnym panelu listy; wartość `1` odpowiada skrajnemu przyciskowi z lewej strony.
- `Typ`: rodzaj zdarzenia wyzwalanego kliknięciem przycisku; dostępne wartości:
    - `Link wewnętrzny`: (wartość domyślna) link do formularza lub listy w nAxiom; należy kliknąć przycisk `Utwórz` obok pola `Link`, aby wygenerować link do formularza lub listy, wyświetlanej po kliknięciu definiowanego przycisku; patrz opis w sekcji Nowa pozycja menu .
    - `Link zewnętrzny`: zewnętrzny adres URL; należy go wpisać w pole `Link`.
    - `Akcja`: pozwala przypisać do przycisku jedną lub kilka globalnych akcji zdefiniowanych w nAxiom, korzystając z wyświetlonego na karcie okna dialogowego wyboru akcji; patrz opis w sekcji Właściwości przycisku na formularzu . Przycisk na liście może także wyzwalać wykonanie następujących akcji systemowych:
        - Otwórz formularz
        - Odśwież listę
        - Odśwież formularz
        - Odśwież wskazany formularz
        - Zamknij formularz

> **WSKAZÓWKA**
> Akcje systemowe dotyczące formularzy są wykonywane tylko w przypadku list osadzonych w sekcji formularza, w przypadku list wyświetlanych w widoku listy, te akcje są ignorowane.

- `Położenie ikony`: określa położenie ikony względem tekstu; dostępne wartości:
    - `Po lewej od tekstu`
    - `Po prawej od tekstu`
    - `Nad tekstem`
    - `Pod tekstem`
- `Kolor tekstu i ikony dla krycia tła przycisku 0% lub braku tła`: określa kolor tekstu i ikony wyświetlany po najechaniu kursorem myszy na przycisk z

przezroczystym tłem; domyślnie jest to `Kolor akcji` ze stylu ustawionego w kreatorze stylów. Do ustawiania kolorów służy selektor, który zawiera następujące elementy:

1. Paleta odcieni, 2. Wyłączenie koloru, 3. Suwak koloru, 4. Suwak krycia, 5. Szesnastkowy kod koloru, 6. Wartości R, G i B koloru, 7. Wartość krycia, 8. Paleta kolorów standardowych.

Selektor koloru

Paleta kolorów standardowych w pierwszych trzech kolumnach zawiera: kolory kreatora stylów (główny, zależne A, B i C, akcji i zależny A), standardowe odcienie szarości oraz kolory z biblioteki *Bootstrap* (btn-success, btn-danger, btn-warning, btn-info, btn-light, btn-dark).

- `Rozmiar tekstu`: określa w procentach rozmiar tekstu względem ustawienia w kreatorze stylów (`INTERFEJS UŻYTKOWNIKA` > `Kreator stylów systemu`); wartość maksymalna to 100%.
- `Rozmiar ikony`: określa w procentach rozmiar ikony względem ustawienia w kreatorze stylów; wartość maksymalna to 100%.
- `Odstęp ikony od tekstu`: określa w pikselach odległość między ikoną i tekstem przycisku.
- `Margines wewnętrzny poziomy`: określa w pikselach *padding* w poziomie (od lewej i od prawej krawędzi przycisku).

- `Margines wewnętrzny pionowy`: określa w pikselach *padding* w pionie (od górnej i od dolnej krawędzi przycisku).

Powiązane tematy:

- Generowanie linku wewnętrznego
- Przypisywanie akcji do przycisku

4.3.17. Właściwości przycisku — karta Dostępność

Podsumowanie:
Na tej karcie można określić warunki dostępności (widocznosci i aktywności) przycisków dostępnych w sekcjach `Górny panel` oraz w kolumnie akcji (`command`). Dotyczy to zarówno przycisków otwierania w formularzu, jak i przycisków do edycji inline. Ta karta nie jest dostępna w przypadku przycisków do obsługi trybu edycji inline.

Warunki dostępności przycisków listy określa się zapytaniem SQL. Zapytanie musi zwrócić wartości *IsVisible* (widoczność przycisku) i/lub *IsReadOnly* (aktywność przycisku) typu *bit*.

W zapytaniu można odwoływać się do wartości z pól formularza, w którym lista jest osadzona jako sekcja, służy do tego standardowa składnia *{@*KodFormularza.NazwaPola}*.

UWAGA
Aby warunki widoczności zostały przeliczone po zmianie wartości w kontekście formularza, konieczne jest użycie akcji Odśwież sekcję formularza.

Powiązane tematy:

- Sekcja typu lista
- Akcja Odśwież sekcję formularza

4.3.18. Właściwości przycisku — karta Notyfikacje

Podsumowanie:
Przyciski w widoku listy mogą znajdować się w sekcji `Górny panel` (tam gdzie standardowo znajduje się przycisk `Nowy dokument`) oraz w poszczególnych wierszach listy, w kolumnie akcji (`command`). Do tych przycisków można przypisywać różne akcje i zestawy akcji oraz określać powiadomienia wyświetlane po wykonaniu tych akcji.

Na karcie `Notyfikacje` można określać następujące ustawienia dotyczące powiadomień wyświetlanych po wykonaniu akcji przypisanych do przycisku:

`Wyłącz pozytywne notyfikacje`: wyłącza wyświetlanie komunikatów po poprawnym wykonaniu akcji przypisanej do przycisku.

`Wyłącz błędne notyfikacje`: wyłącza wyświetlanie komunikatów w przypadku niewykonania akcji przypisanej do przycisku.

`Wiadomość po poprawnym wykonaniu akcji`: tekst komunikatu wyświetlanego po poprawnym wykonaniu akcji.

`Pozycja pozytywnej notyfikacji`: miejsce wyświetlania na ekranie okienka z komunikatem o powodzeniu; dostępne wartości:

- `Prawa góra`: (wartość domyślna) prawy górny róg ekranu.
- `Domyślny (z ustawień globalnych aplikacji`:
- `Środkowa góra`: na środku u góry ekranu.
- `Lewa góra`: lewy górny róg ekranu.
- `Prawy dół`: prawy dolny róg ekranu.
- `Środkowy dół`: na środku u dołu ekranu.
- `Lewy dół`: lewy dolny róg ekranu.

`Czas wyświetlania pozytywnej notyfikacji (w sekundach)`: czas w sekundach, po którym komunikat zostanie wygaszony; wartość 0 powoduje, że wygaszenie komunikatu wymaga kliknięcia go przez użytkownika.

`Wiadomość po niewykonanej akcji`: tekst komunikatu wyświetlanego, kiedy akcja nie zostanie wykonana.

`Pozycja błędnej notyfikacji`: miejsce wyświetlania na ekranie okienka z komunikatem o niepowodzeniu; dostępne wartości:

- `Prawa góra`: (wartość domyślna) prawy górny róg ekranu.
- `Domyślny (z ustawień globalnych aplikacji`:
- `Środkowa góra`: na środku u góry ekranu.
- `Lewa góra`: lewy górny róg ekranu.
- `Prawy dół`: prawy dolny róg ekranu.
- `Środkowy dół`: na środku u dołu ekranu.
- `Lewy dół`: lewy dolny róg ekranu.

`Czas wyświetlania negatywnej notyfikacji (w sekundach)`: czas w sekundach, po którym komunikat zostanie wygaszony; wartość 0 powoduje, że wygaszenie komunikatu wymaga kliknięcia go przez użytkownika.

4.3.19. Przyciski listy do edycji inline

> **Podsumowanie:**
> Włączenie opcji edycji inline we właściwościach listy powoduje dodanie przycisków do edycji inline w kolumnie akcji i/lub na górnym panelu. W definicji listy te przyciski są dodawane w drzewie elementów listy, w gałęzi *Przyciski listy/Wiersz* i/lub *Przyciski listy/Górny panel*. Niektóre z tych przycisków wymagają skonfigurowania odpowiedniej akcji.

Niektóre przyciski do edycji inline, oprócz standardowych właściwości przycisków, wymagają skonfigurowania dodatkowej akcji. W zależności od włączonego trybu edycji inline oraz operacji w tym trybie na liście mogą występować następujące przyciski:

- Tryb edycji inline `po kliknięciu w wiersz`:
  - *Przyciski listy* > *Górny panel*:
    - `Anuluj [Anuluj inline]`: anuluje zmiany wprowadzone we wszystkich wierszach od ostatniego zapisu.
    - `Zapisz [Zapisz inline]`: zapisuje zmiany wprowadzone we wszystkich wierszach od ostatniego zapisu; wymaga skonfigurowania akcji zapisu.
  - *Przyciski listy* > *Wiersz* (kolumna akcji):
    - `Usuń [Usuń inline]`: usuwa bieżący wiersz; wymaga skonfigurowania akcji usuwania.
- Tryb edycji inline `po kliknięciu w przycisk edycji`:
  - *Przyciski listy* > *Górny panel*:
    - `Dodaj [Dodaj inline]`: dodaje nowy wiersz u góry widoku listy.
  - *Przyciski listy* > *Wiersz* (kolumna akcji):
    - `Edycja [Edycja inline]`: przełącza pola bieżącego wiersza w tryb edycji inline.
    - `Zapisz [Zapisz inline]`: zapisuje zmiany wprowadzone w bieżącym wierszu od ostatniego zapisu; wymaga skonfigurowania akcji zapisu.
    - `Anuluj [Anuluj inline]`: anuluje zmiany wprowadzone w bieżącym wierszu od ostatniego zapisu.
    - `Usuń [Usuń inline]`: usuwa bieżący wiersz; wymaga skonfigurowania akcji usuwania.

ZAPIS INLINE

Przycisk zapisu w trybie inline wymaga skonfigurowania akcji SQL do bazy systemowej. Ponieważ dla edycji inline definiuje się tylko jedną akcję zapisu, powinna ona uwzględniać przypadki zapisu nowego wiersza (*INSERT*) i zapisu zmian w wierszu (*UPDATE*) w tym celu stosuje się następujący warunek:

```
IF ({@Id} IS NULL)
  INSERT INTO [dbo].[TabelaDefinicjiDokumentu]
        (Kolumna1)
  VALUES
        ({@Kolumna1})
ELSE
UPDATE [dbo].[TabelaDefinicjiDokumentu]
SET Kolumna1 = {@Kolumna1}
WHERE Id = {@Id}
```

Nazwy używane w zmiennych *{@Nazwa}* to właściwości `Nazwa SQL` odpowiednich kolumn.

W przypadku list, dla których tabela źródłowa jest oparta na modelu standardowym, zapis nowego wiersza wymaga podania wartości dla kolumn systemowych (np. *Status*). Poniżej przykład zapytania *INSERT*, które zapisuje pełny zestaw danych dla nowego dokumentu (nie uwzględniono kolumn definiowanych przez użytkownika):

```
IF ({@Id} IS NULL)
  INSERT INTO [dbo].[DataTypeTest]
          ([Transition]
          ,[Status]
          ,[Type1]
          ,[Type2]
          ,[Type3]
          ,[Code]
          ,[Organization]
          ,[RowVersion]
          ,[UserId]
          ,[CreateDate]
          ,[UpdatedBy]
          ,[UpdatedDate]
          ,[ACLId]
          ,[ArchiveStatus]
          ,[Blocked]
```

```
            ,[BlockUser]
            ,[LastEntryDate]
            ,[IsAnonimized])
      VALUES
            ([transition-id]
            ,[status-id]
            ,NULL
            ,NULL
            ,NULL
            ,[code]
            ,NULL
            ,1
            ,{@_UserId}
            ,GETDATE()
            ,{@_UserId}
            ,GETDATE()
            ,NEWID()
            ,1
            ,0
            ,NULL
            ,NULL
            ,0)
ELSE
  UPDATE ...
```

gdzie *transition-id* to numer przejścia na diagramie procesu biznesowego, *status-id* identyfikator pierwszego statusu (domyślnie jest to status *Utworzony*), a *code* to kod nowej instancji dokumentu.

USUWANIE INLINE

Przycisk usuwania w trybie inline wymaga skonfigurowania akcji SQL do bazy systemowej z następującym zapytaniem:

```
DELETE FROM [dbo].[TabelaDefinicjiDokumentu]
WHERE Id = {@Id}
```

WAŻNE
W przypadku kiedy lista jest osadzona jako sekcja na formularzu, w akcjach przycisków zapisu i usuwania inline można odwoływać się do wartości z kontekstu tego formularza, stosując składnię:

```
{@*KodFormularza.NazwaPolaWKontekście}
```

Powiązane tematy:

- Właściwości listy — karta Edycja inline
- Kolumna listy — karta Edycja inline

4.4. PULPITY APLIKACJI

Podsumowanie:
Pulpit to grupa widgetów wyświetlana na stronie startowej. W nAxiom można definiować dwa rodzaj pulpitów:

- użytkownika
- aplikacji

  Pulpity aplikacji, w odróżnieniu od pulpitów użytkownika, można migrować.

☛ *AdminSPA* > `INTERFEJS UŻYTKOWNIKA` > `Pulpity`

Aby zadeklarować nowy pulpit, zaloguj się jako konsultant do *AdminSPA* i wybierz `INTERFEJS UŻYTKOWNIKA` > `Pulpity`. Kliknij `Nowy pulpit` i podaj następujące właściwości:

- `Kod`
- `Nazwa`
- `Opis`
- `Aplikacja`: wybierz aplikację, z którą definicja pulpitu będzie migrowana.
- `Moduł`
- `Aktywny`: po wyłączeniu pulpit nie będzie widoczny na liście pulpitów we FrontSPA, ale będzie wyświetlany dla aplikacji, dla których jest pulpitem domyślnym.

- `Domyślny dla aplikacji`: wyświetla listę aplikacji, dla których wybrano ten pulpit jako domyślny.

Konfiguracja widżetów na pulpicie aplikacji odbywa się tak samo jak w przypadku pulpitu użytkownika:

1. Zaloguj sie we FrontSPA.
2. Rozwiń listę wyboru pulpitu i wybierz z niej polecenie `Lista pulpitów`.
3. Kliknij przycisk `Designer` w wierszu pulpitu, który chcesz edytować.

UWAGA
Edycja pulpitów aplikacji wymaga odpowiedniego uprawnienia PBA

4.5. WIDŻETY

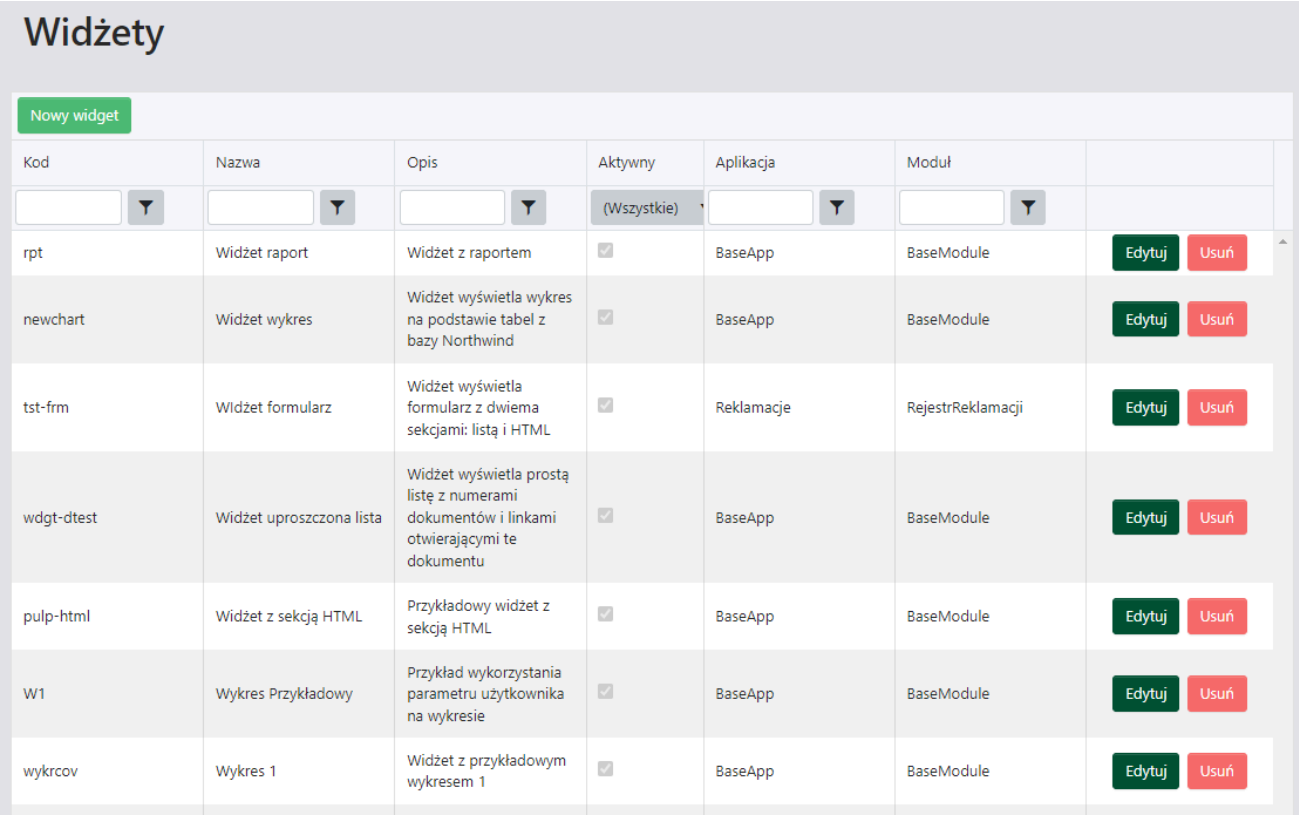
Podsumowanie:
Widżety to elementy interfesju użytkownika wyświetlane w aplikacji *FrontSPA* na *pulpitach* (*dashboard*), na stronie startowej. Dostępnych jest kilka rodzajów widżetów: uproszczona lista, sekcja HTML, wykres, raport i formularz. Konfiguracja widżetów (oprócz widżetu formularza) jest identyczna jak konfiguracja odpowiadających im sekcji formularza. Dodatkowo, w konfiguracji widżetów można zdefiniować parametry, zarówno lokalne, jak i globalne dla widżetów

☛ (*AdminSPA*) `INTERFEJS UŻYTKOWNIKA` > `Widżety`

W TYM TEMACIE

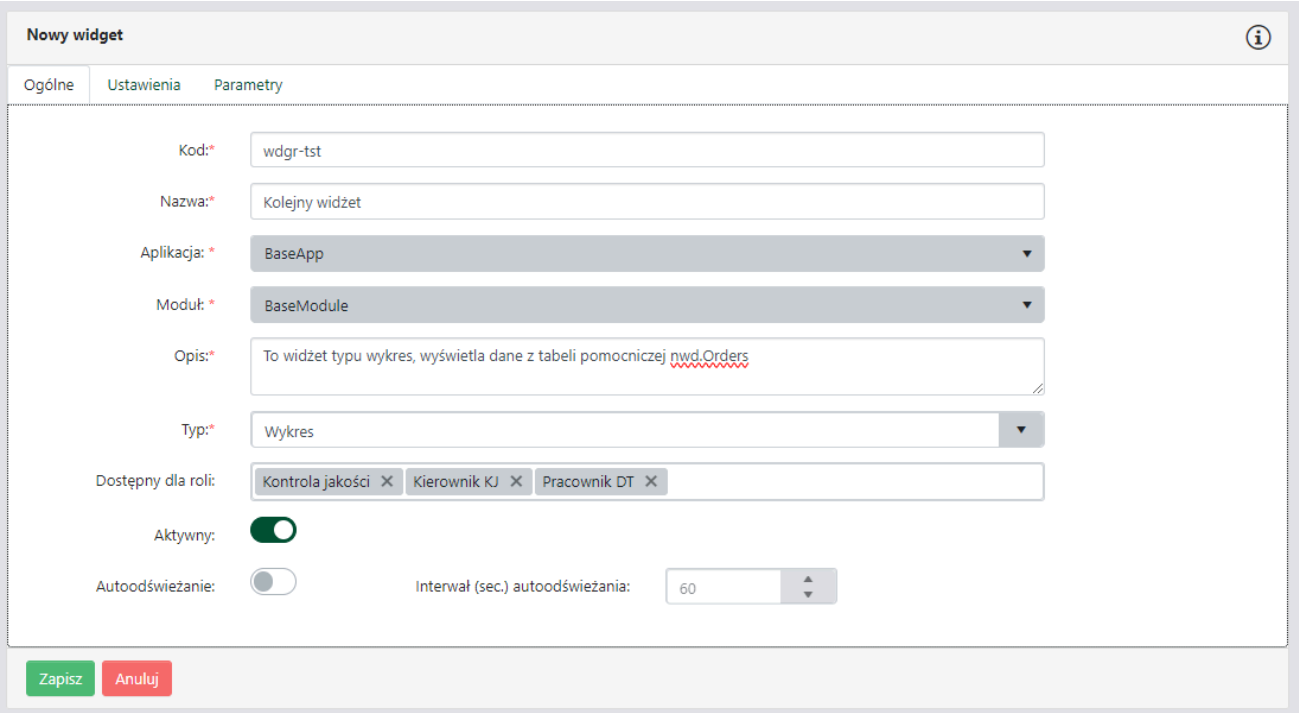
- Zarządzanie widżetami
- Konfigurowanie widżetów
- Parametry widżetów

ZARZĄDZANIE WIDŻETAMI



Lista widżetów

Przyciski obok definicji każdego widżetu pozwalają edytować widżety lub je usuwać. Aby utworzyć nowy widżet kliknij przycisk `Nowy widget` i uzupełnij podstawowe właściwości widżetu.



Właściwości widżetu

- `Kod`: kod widżetu.

- `Nazwa`: nazwa widżetu.
- `Aplikacja`: aplikacja, z którą jest powiązany widżet.
- `Moduł`: moduł aplikacji , z którą jest powiązany widżet.
- `Opis`: opis widżetu.
- `Typ`: rodzaj widżetu; dostępne wartości:
    - `Uproszczona lista`
    - `Sekcja HTML`
    - `Wykres`
    - `Raport`
    - `Formularz`
- `Dostępny dla roli`: lista wielowyboru ról biznesowych; po wybraniu ról widżet będzie dostępny do ustawienia na pulpicie tylko dla użytkowników, którzy mają przypisaną co najmniej jedną z wybranych ról; domyślnie widżety są dostępne dla wszystkich użytkowników; te ustawienia są zapisywane w tabeli *core.WidgetRoles*.
- `Minimalna wysokość widżetu w unitach`: dostępne wartości *1 unit*, *2 unity*, *bez ograniczeń*. 1 unit odpowiada mniej więcej 1/3 wysokości obszaru pulpitu w aplikacji *FrontSPA* dla rozdzielczości FullHD.
- `Pokazuj toolbar widżetu`: pozwala wyłączyć pasek z nazwą widżetu i ikoną do zmiany parametrów (o ile widżet ma zdefiniowane parametry); w przypadku widżetu formularz jest to sekcja toolbar.
- `Aktywny`: widżetów nieaktywnych nie można ustawiać na pulpiciew aplikacji *FrontSPA*.
- `Autoodświeżanie`: po włączeniu powoduje automatyczne odświeżanie danych wyświetlanych przez widżet z podanym interwałem.
- `Interwał autoodświeżania`: okres w sekundach, z jakim dane wyświetlane w widżecie będą odświeżane.

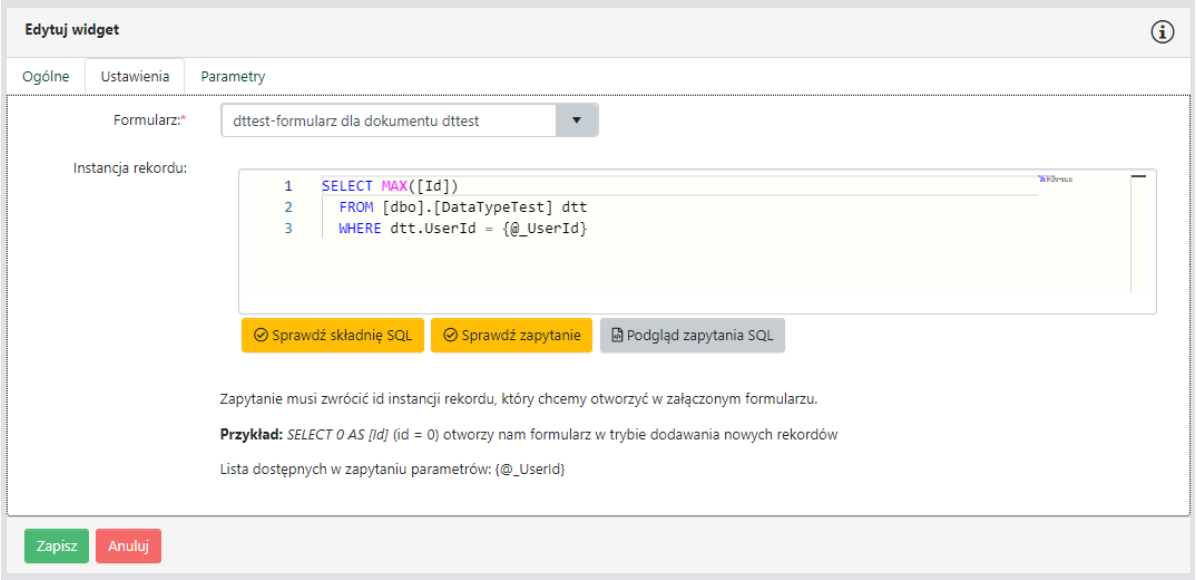
Karty `Ustawienia` i `Parametry` są wyświetlane dopiero po wybraniu rodzaju widżetu (dla raportu karta `Parametry` nie jest dostępna).

Definicje widżetów są zapisywane w tabeli *core.Widgets*.

KONFIGUROWANIE WIDŻETÓW

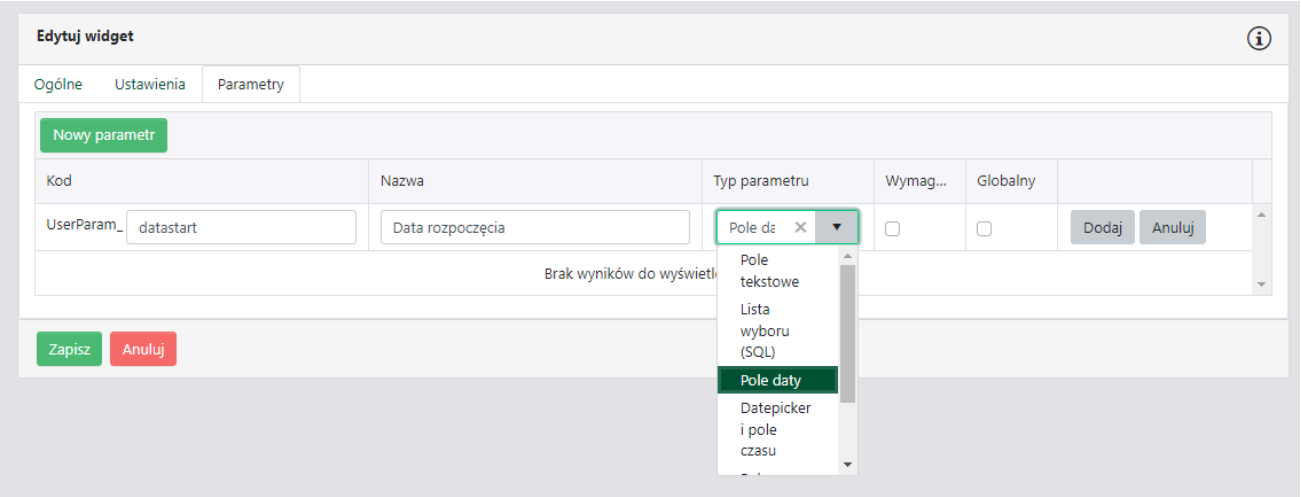
Po zdefiniowaniu nowego widżetu należy przejść do jego konfiguracji na karcie `Ustawienia`. W większości przypadków ta konfiguracja jest taka sama jak w przypadku analogicznych sekcji formularza:

- widżet uproszczona lista odpowiada sekcji typu prosta lista,
- widżet sekcja HTML odpowiada sekcji HTML; w tym widżecie obsługiwana jest bibliotek chart.js, przy użyciu której można definiować wykresy, została ona opisana w kolejnym temacie: Obsługa biblioteki chart.js .
- widżet wykres odpowiada sekcji wykres,
- widżet raport odpowiada sekcji podgląd raportu,
- widżet formularz wymaga jedynie wybrania formularza z listy oraz utworzenia zapytania wybierającego identyfikator dokumentu, który ma zostać otwarty w formularzu (podobnie jak w akcji otwierania formularza).



Konfiguracja widżetu formularz

PARAMETRY WIDŻETÓW



Parametry widżetu

Prawie wszystkie rodzaje widżetów (wyjątkiem jest raport) umożliwiają definiowanie parametrów, które można ustawiać w aplikacji *FrontSPA*, aby wpływać na zakres danych wyświetlanych w widżetach. Parametry definiuje się na karcie `Parametry` po klimnięciu przycisku `Nowy parametr`. Należy kolejno określić:

- `Kod`: kod parametru; w zapytaniu źródłowym dla widżetu odwołanie do parametru ma składnię *{@UserPAram_kod}*.
- `Nazwa`: zrozumiała dla użytkownika nazwa parametru.
- `Typ parametru`: wybierz z listy kontrolkę, która będzie wyświetlana w oknie dialogowym w celu określenia wartości parametru; dostępne pozycje:
  - `Pole tekstowe`
  - `Lista wyboru SQL`: należy skonfigurować zapytanie SQL dla listy wyboru; zapytanie musi zwrócić kolumny *[Key]* (wartość parametru) i *[Value]* (wartość wyświetlana na liście)
  - `Pole daty`
  - `Datepicker i pole czasu`
  - `Pole numeryczne`
  - `Checkbox (pojedynczy)`
- `Wymagany`: określa, czy podanie wartości dla parametru jest wymagane.
- `Globalny`: określa, czy wartość parametru ustawione dla dane widżetu ma obowiązywać również w innych widżetach, w których zdefiniowano taki sam parametr.

Określanie wartości parametru w aplikacji FrontSPA

Definicje parametrów widżetów są zapisywane w tabeli *core.WidgetParameters*

Powiązane tematy:

- Sekcje formularza, prosta lista
- Sekcje formularza, HTML
- Sekcje formularza, wykres
- Sekcje formularza, raport
- Pulpity w aplikacji FrontSPA

4.5.1. Obsługa biblioteki chart.js

Podsumowanie:
W wersji nAxiom 1.12.0.0 obsługiwana jest biblioteka Chart.js w wersji 4.4.0 (licencja MIT). Kod biblioteki został włączony do platformy nAxiom, dzięki czemu można z niej korzystać jak z natywnej funkcjonalności.

Biblioteka umożliwia renderowanie wykresów różnego typu z użyciem JavaScript oraz zasilanie ich danymi pochodzącymi z bazy danych lub kontekstu formularza. Konfiguracja wykresów jest przedstawiona w dokumentacji projektu Chart.js na stronie projektu.

Wykresy można definiować w sekcjach HTML (na formularzach oraz widgetach typu sekcja HTML), w kontrolkach blok HTML oraz w kolumnie HTML w widoku listy.

W TYM TEMACIE

- Wstęp
- Ogólny przykład konfiguracji wykresu
- Przykład kodu JavaScript dla wykresu
- Zalecenia
- Przykład konfiguracji dwóch wykresów w jednej sekcji
- Przykład wykresu liniowego w kolumnie HTML

WSTĘP

W tym temacie opisano sposób definiowania wykresu w sekcji HTML. Opis ten pozostaje właściwy także dla kontrolki blok HTML. W przypadku kolumny HTML w widoku listy dane do wykresu muszą pochodzić z bieżącego rekordu widoku listy (*RowDataContext.NazwaKolumny*, gdzie *NazwaKolumny* to właściwość *Nazwa SQL*); patrz

także Zalecenia poniżej. Wykres jest generowany kodem JavaScript jako element HTML *canvas*. Konfiguracja wykresu w sekcji HTML obejmuje trzy części:

- Zapytanie SQL: zwraca dane do wykresu, jeden rekord jest jedną serią danych, nazwy kolumn to etykiety wartości, a zwrócone dane to same wartosci.
- Szablon HTML: musi zawierać deklarację elementu *canvas* z identyfikatorem, który zostanie użyty w kodzie JavaScript:

```
<canvas id="niepowtarzalny-identyfikator"></canvas>
```

> Aby poprawnie wyświetlić kilka wykresów na stronie nAxiom, ich elementy *canvas* muszą mieć niepowtarzalne identyfikatory.

- Kod JavaScript: definiuje podstawowe elementy wykresu, takie jak rodzaj wykresu, serie danych, opcje konfiguracyjne. W kodzie należy także zadeklarować zmienne, które będą uzywane w seriach danych jako etykiety i/lub wartości i powiązać je z danymi zwróconymi przez zapytanie, wskazując nazwę kolumny i numer rekordu (pierwszy rekord ma numer 0).

OGÓLNY PRZYKŁAD KONFIGURACJI WYKRESU

Zapytanie SQL:

```
select
     () as InProgress
    ,() as Delayed
```

Szablon HTML:

```
<div>
    <canvas id="chart1"></canvas>
</div>
```

Kod JavaScript:

W przykładzie zadeklarowano zmienne używane w seriach danych jako wartości. Za zmienne są podstawiane dane z zapytania SQL — wskazana kolumna i numer rekordu.

W przypadku sekcji HTML na formularzu, w tym na widżecie typu formularz, referencja do wyniku zapytania SQL wymaga użycia kodu formularza:

```
var DanaDoWykresu = window["KodFormularza_QueryModel"][indeksWierszaZapytaniaSQL].NazwaKolumnyZapytaniaSQL;
```

W przypadku użycia widgetu typu sekcja HTML, należy użyć w referencji kodu pulpitu, na którym ma być używany widżet:

```
window["kodPulpitu_QueryModel"][indeksWierszaZapytaniaSQL].NazwaKolumnyZapytaniaSQL
```

PRZYKŁAD KODU JAVASCRIPT DLA WYKRESU

```
//deklaracja zmiennych i przypisywanie wartości
var val_InProgress = window["MyDashboardCode_QueryModel"][0].InProgress;
var val_Delayed = window["MyDashboardCode_QueryModel"][0].Delayed;

//deklarowanie wykresu
var ctx = document.getElementById('chart1').getContext('2d');

//definiowanie serii danych
var data = {
  // właściwa konfiguracja serii danych - tutaj używa się zmiennych zadeklarowanych na początku
};

//definiowanie konfiguracji wykresu
var config = {
      // właściwa konfiguracja wykresu
};

//utworzenie wykresu
new Chart(ctx, config);
```

ZALECENIA

1. W celu poprawnego renderowania kilku wykresów na stronie nAxiom, elementy *canvas* każdego wykresu muszą mieć niepowtarzalne wartości atrybutu *id*.
2. Dla kolumny HTML zaleca się stosowanie automatycznej wysokości wykresu i definiowanie jedynie szerokości zgodnej z szerokością kolumny. Pozwoli to na indywidualne dostosowanie wysokości wykresu w każdym rekordzie.
3. Kod HTML w kolumnie HTML musi zawierać co najmniej jeden element z atrybutem *rowHtmlElement* o dowolnej wartości, która jednak nie może zawierać cyfr. W przeciwnym razie kod Javascript nie zostanie wykonany.

4. Aby kontrolować rozmieszczenie wykresów w obrębie sekcji zaleca się stosowanie techniki *Flexbox Layout*.

PRZYKŁAD KONFIGURACJI DWÓCH WYKRESÓW W JEDNEJ SEKCJI

Zapytanie SQL:

```
select
     (select count(Id) from [dbo].[RecruitmentProcess]
        where [RecruitmentDeadline] >= getdate()
        and Worksite = {&HRDevelopment.Dictionaries.Worksite.Design}
        and Status != {&HRDevelopment.BusinessStatuses.RecruitmentProcess.Rec_Finished}
        ) as WorksiteDesignInTimeNotFinished
    ,(select count(Id) from [dbo].[RecruitmentProcess]
        where [RecruitmentDeadline] < getdate()
        and Worksite = {&HRDevelopment.Dictionaries.Worksite.Design}
        and Status != {&HRDevelopment.BusinessStatuses.RecruitmentProcess.Rec_Finished}
        ) as WorksiteDesignOutOfTimeNotFinished
    ,(select count(Id) from [dbo].[RecruitmentProcess]
        where [RecruitmentDeadline] >= getdate()
        and Worksite = {&HRDevelopment.Dictionaries.Worksite.RoadConstruction}
        and Status != {&HRDevelopment.BusinessStatuses.RecruitmentProcess.Rec_Finished}
        ) as WorksiteRoadConstructionInTimeNotFinished
    ,(select count(Id) from [dbo].[RecruitmentProcess]
        where [RecruitmentDeadline] < getdate()
        and Worksite = {&HRDevelopment.Dictionaries.Worksite.RoadConstruction}
        and Status != {&HRDevelopment.BusinessStatuses.RecruitmentProcess.Rec_Finished}
        ) as WorksiteRoadConstructionOutOfTimeNotFinished
```

Szablon HTML:

```
<QueryRowTemplate>
    <div
        class="widgetFullContainer"
        style="display:flex; flex-wrap: wrap; flex-direction: row; align-items: center; justify-content: spa
        <div
            class="chart"
            style="width: 40%">
            <canvas id="chart1"></canvas>
            <p style="text-align:center">Dział projektów</p>
```

```
        </div>
        <div
            class="chart"
            style="width: 40%">
            <canvas id="chart2"></canvas>
            <p style="text-align:center">Dział budowy dróg</p>
        </div>
    </div>
</QueryRowTemplate>
```

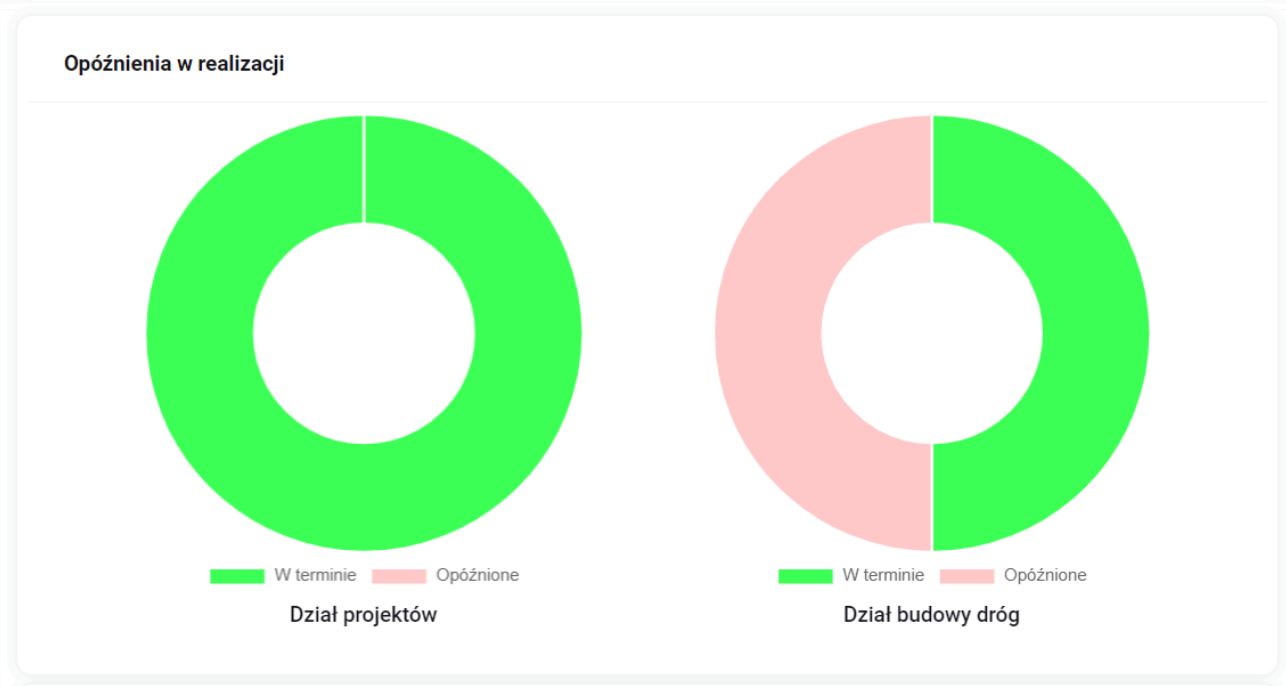
Kod JavaScript:

```
var ValueWorksiteDesignInTimeNotFinished = window["HRDDeadlinesByWorkstite_QueryModel"][0].WorksiteDesignInT
var ValueWorksiteDesignOutOfTimeNotFinished = window["HRDDeadlinesByWorkstite_QueryModel"][0].WorksiteDesigr
var ValueWorksiteRoadConstructionInTimeNotFinished = window["HRDDeadlinesByWorkstite_QueryModel"][0].Worksit
var ValueWorksiteRoadConstructionOutOfTimeNotFinished = window["HRDDeadlinesByWorkstite_QueryModel"][0].Work

var ctx = document.getElementById('chart1').getContext('2d');

var data = {
  labels: ['W terminie', 'Opóźnione'],
  datasets: [
    {
      label: 'Dataset 1',
      data: [ValueWorksiteDesignInTimeNotFinished, ValueWorksiteDesignOutOfTimeNotFinished],
      backgroundColor: [
        'rgba(0, 255, 34, 0.75)',
        'rgba(255, 183, 183, 0.75)',
      ],
    }
  ]
};

var config = {
  type: 'doughnut',
  data: data,
    options: {
        responsive: true,
        plugins: {
        legend: {
            position: 'bottom',
```

```
        },
        title: {
            display: false,
        }
        }
    },
    };
}
new Chart(ctx, config);

//

var ctx = document.getElementById('chart2').getContext('2d');

var dataWorksiteRoadConstruction = {
  labels: ['W terminie', 'Opóźnione'],
  datasets: [
    {
      label: 'Dataset 2',
      data: [ValueWorksiteRoadConstructionInTimeNotFinished, ValueWorksiteRoadConstructionOutOfTimeNotFinish
      backgroundColor: [
        'rgba(0, 255, 34, 0.75)',
        'rgba(255, 183, 183, 0.75)',
      ],
    }
  ]
};

var configWorksiteRoadConstruction = {
  type: 'doughnut',
  data: data,
  options: {
    responsive: true,
    plugins: {
      legend: {
        position: 'bottom',
      },
      title: {
        display: false,
      }
    }
  },
```

```
};
new Chart(ctx, config);
```

Wynik końcowy:



Przykład wykresu

PRZYKŁAD WYKRESU LINIOWEGO W KOLUMNIE HTML

Szablon HTML:

```
<div rowHtmlElement="thisChart" style="width: 300px;">
  <canvas id="doughnutChartForm_{@Id}"></canvas>
</div>
```

Kod Javascript:

```
var chartId = 'doughnutChartForm_' + RowDataContext.Id;
var ctx = document.getElementById(chartId).getContext('2d');
var data = {
  labels: ['Q1', 'Q2', 'Q3', 'Q4'],
  datasets: [
    {
      label: 'Sales by quater,
      data: [
```

```
        RowDataContext.Column1_SQLName,
        RowDataContext.Column2_SQLName,
        RowDataContext.Column3_SQLName,
        RowDataContext.Column4_SQLName
      ]
    }
  ]
};

var config = {
  type: 'line',
  data: data,
  options: {
    responsive: true,
    plugins: {
      legend: {
        position: 'top',
      },
      title: {
        display: true,
        text: 'Chart.js Chart'
      }
    }
  },
};

new Chart(ctx, config);
```

Powiązane tematy:

- Formularz, sekcja HTML

4.6. KONTROLKI

Podsumowanie:
Kontrolki to elementy interfejsu użytkownika używane na formularzach oraz w widoku listy w trybie edycji inline. nAxiom oferuje możliwość osobnego definiowania kontrolek typu lista wyboru, np. w celu korzystania z tej samej listy w różnych formularzach i widokach listy.

☛ *AdminSPA* > `INTERFEJS UŻYTKOWNIKA` > `Kontrolki`

Na stronie `Kontrolki` można definiować nowe kontrolki oraz modyfikować i usuwać zdefiniowane kontrolki.

Aby skorzystać ze zdefiniowanej kontrolki:

- na formularzu: dodaj do formularza kontrolkę typu lista wyboru (`Lista wyboru (SQL)`, `Lista wyboru (SQL - async)` i `Lista wielowyboru (SQL)`), przejdź do jej konfiguracji w oknie dialogowym `Konfiguracja listy wyboru`, wyłącz przełącznik `Konfiguracja poprzez SQL`, a następnie wybierz zdefiniowaną kontrolkę z listy `Zdefiniowana lista wyboru`.
- w widoku listy: włącz tryb edycji inline dla widoku listy, przejdź do właściwości wybranej kolumny, wyświetl kartę `Edycja inline`, z listy `Typ pola` wybierz pozycję `Lista wyboru SQL`, z listy `Zdefiniowana lista wyboru` wybierz nazwę kontrolki, której chcesz użyć; przykład użycia opisano w jednym z powiązanych tematów.

DEFINIOWANIE KONTROLKI

W celu utworzenia nowej kontrolki kliknij przycisk `Nowa kontrolka` z lewej strony u góry listy `Kontrolki` i wypełnij pola tego okna dialogowego zgodnie z opisem poniżej. Następnie kliknij przycisk `Zapisz`. Na liście kontrolek w każdym rekordzie są dostępne przyciski `Edytuj` i `Usuń`, ktorymi można odpowiednio edytować definicję kontrolki lub ją usunąć nie można usunąć kontrolki, która jest używana na formularzu.

Aby zdefiniować kontrolkę, określ następujące właściwości:

`Kod kontrolki`: krótki ciąg znaków identyfikujący kontrolkę.

`Nazwa kontrolki`: nazwa kontrolki.

`Aplikacja`: nazwa aplikacji, do której zostanie przypisany tworzony obiekt; wszystkie obiekty przypisane do aplikacji można wraz z nią wyeksportować, używając polecenia `Migrator aplikacji` w sekcji `ADMINISTRACJA` menu głównego.

`Moduł`: nazwa modułu aplikacji, do którego zostanie przypisany tworzony obiekt.

`Typ kontrolki`: rodzaj tworzonej kontrolki; dostępne wartości:

- `Lista wyboru`: lista wartości, z której można wybrać jedną po rozwinięciu listy.

`Źródło danych`: źródło danych dla listy; dostępne wartości:

- `Dokument biznesowy`: źródłem danych jest tabela powiązana z wybraną definicją dokumentu; dodatkowo określ właściwości:
  - `Dokument biznesowy`: nazwa definicji dokumentu.
  - `Wartość zwracana`: nazwa kolumny tabeli, z której wartości będą zapisywane po wybraniu wartości w kontrolce.
  - `Wartość wyświetlana`: nazwa kolumny tabeli, z której wartości będą wyświetlane na liście rozwijanej.
  - `Dodaj filtr`: otwiera okno, w którym można skonstruować warunek filtrowania dla pozycji zwracanych przez listę wyboru; w warunku można używać parametrów:
    - *{@Nazwa}*: wartość z pola formularza lub pola rekordu w widoku listy.
    - *{@*KodFormularza.Nazwa}*: wartość z pola formularza o podanym kodzie; odwołanie dotyczy sytuacji, kiedy kontrolka jest używana w widoku listy osadzonej w formularzu.
- `Zapytanie SQL`: źródłem danych jest zapytanie SQL; dodatkowo określ właściwości:
  - `SQL`: kliknięcie przycisku `Zapytanie` otwiera okno `Edytor kodu`, w którym należy wpisać instrukcje SQL zwracające kolumny *key* i *value*; jeśli zapytanie zwróci inne kolumny, można będzie ich użyć do wypełnienia innych pól na formularzu (`Lista zależności: kolumna - pole formularza`) lub w rekordzie widoku listy (`Lista zależności: kolumna - kolumna`); wyniki zapytania można filtrować, używając takich samych parametrów, jak opisano powyżej.

Powiązane tematy:

- Lista wyboru (SQL)
- Lista wyboru (SQL - Async)
- Lista wielowyboru (SQL)
- Przykład: lista wyboru w trybie edycji inline

4.7. MENU UŻYTKOWNIKA

Podsumowanie:
Menu użytkownika to menu poleceń dostępne w aplikacji *FrontSPA*, za pomocą którego użytkownicy końcowi mogą korzystać z aplikacji biznesowych budowanych w nAxiom. Po zdefiniowaniu sekcji menu i dostępnych w nim poleceń należy nadać rolom biznesowym uprawnienia do tych poleceń. Zdefiniowane polecenia menu użytkownika są widoczne tylko dla użytkowników z przypisaną rolą uprawnioną do tego polecenia.

☛ *AdminSPA* > `INTERFEJS UŻYTKOWNIKA` > `Menu użytkownika`

W TYM TEMACIE

- Wprowadzenie
- Sekcje menu
- Pozycje menu

WPROWADZENIE

Menu może mieć strukturę wielopoziomową i dzieli się na sekcje, które zawierają pozycje menu. W nAxiom można wyróżnić dwa rodzaje pozycji menu: zwykłe (domyślne) i nadrzędne (wymagane włączenie przełącznika `Pozycja nadrzędna`). Zarówno sekcje, jak i nadrzędne oraz zwykłe pozycje menu służą do wyświetlania przypisanych formularzy lub widoków listy.

Zawartość menu we FrontSPA można filtrować według aplikacji, do których są przypisywane sekcje menu. Jest to tzw. tryb pojedynczej aplikacji. Oprócz menu wybranej aplikacji, w menu użytkownika wyświetlane są sekcje `USTAWIENIA SYSTEMOWE` z pozycjami menu zależnymi od uprawnień PBA użytkownika, oraz tzw. sekcja współdzielona, w której wyświetlane są pozycje menu z włączonym ustawieniem `Wyświetlaj w sekcji współdzielonej`.

Na listach sekcji/pozycji menu w każdym wierszu znajdują się przyciski:

- `Edytuj`: otwiera okno dialogowe z właściwościami i pozycjami sekcji/pozycji;
- `Kopiuj`: tworzy nową sekcję o tej samej nazwie i z tymi samymi pozycjami menu;
- `Usuń`: usuwa sekcję wraz z jej pozycjami lub usuwa pojedynczą pozycję.

Ponadto, w kolumnie `Kolejność` znajdują się przyciski , za pomocą których można zmieniać kolejność wyświetlania sekcji w menu użytkownika.

SEKCJE MENU

Kliknięcie przycisku `Dodaj sekcję menu` z lewej strony u góry otwiera okno dialogowe `Dodanie sekcji menu`. W tym oknie należy określić następujące właściwości:

- `Kod`: wpisz dowolny ciąg znaków, który będzie identyfikował sekcję menu.
- `Nazwa`: wpisz nazwę sekcji menu; nazwa sekcji będzie wyświetlana w menu użytkownika.
- `Aplikacja`: wybierz aplikację, do której zostanie przypisana sekcja menu.
- `Moduł`: wybierz moduł aplikacji, do którego zostanie przypisana sekcja menu.
- `Kolejność`: pole tylko do odczytu, które informuje o tym, w jakiej kolejności (licząc od góry) będzie wyświetlana dana sekcja menu; wartość `auto` wyświetlana dla nowej sekcji oznacza, że będzie ona domyślnie wyświetlana jako ostatnia, co można zmienić, patrz .
- `Zwijanie sekcji`: włącza możliwość zwijania sekcji menu.
- `Domyślnie rozwiń pozycje`: jeśli włączono zwijanie sekcji, ten przełącznik pozwala określić, czy sekcja ma być domyślnie rozwinięta.
- `Kod formatki`: kliknij przycisk `Utwórz`, aby wyświetlić okno dialogowe `Generowanie kodu formatki...` i wybrać formularz lub widok listy, który będzie wyświetlany po kliknięciu danej pozycji menu; elementy dostępne w tym oknie opisano poniżej.
- `wybierz parametry i wartości`: wyświetla okno dialogowe, w którym można przypisać polom otwieranego formularza określone wartości, analogicznie jak w przypadku akcji otwierania formularza; można przypisywać wartości stałe lub zwracane przez zapytanie SQL, można korzystać ze zmiennych systemowych, np. *{@_UserId}* oraz ze składni smart numbers.
- `Wyświetlanie nazw elementów menu`: określa sposób wyświetlania nazw pozycji menu, które są zbyt długie, aby w całości zmieścić się na panelu menu; dostępne opcje:
  - `Wyświetlaj jeden wiersz i dopisuj wielokropek (...)`
  - `Zawijaj na białym znaku`
- `Wyświetlana etykieta`: zapytanie SQL, które musi zwrócić trzy parametry etykiety wyświetlanej w menu, z prawej strony danego elementu menu:
  - **LabelText** (*nvarchar*): tekst wyświetlany na etykiecie.

- **LabelTextColor**: szesnastkowy kod koloru tekstu na etykiecie.
- **LabelBackground**: szesnastkowy kod koloru tła etykiety.

> **INFO**
> W zapytaniu są obsługiwane zmienne systemowe (*@_UserId*, *@_LangId*, *@_TenantId* i *@_TenantBaseUrl*), zmienne aplikacji oraz składnia smart numbers.

Wyboru obiektu wyświetlanego po kliknięciu danej pozycji/sekcji menu dokonuje sie w oknie wyświetlanym po kliknięciu przycisku `Utwórz` obok właściwości `Kod formatki`.

1. Wybierz z listy formularz, który ma być otwierany po kliknięciu definiowanej pozycji menu, i kliknij przycisk `Utwórz`.
   W polu `Kod formatki` zostanie wpisany kod formularza w formie ciągu `/n/0`, gdzie `n` to identyfikator GUID formularza, a `0` oznacza nowy rekord.
2. Aby kliknięcie definiowanej pozycji menu powodowało wyświetlenie widoku listy, w oknie `Generowanie kodu formatki...` włącz przełącznik `Wybieraj wśród zdefiniowanych list`.
   Zostanie wyświetlona lista widoków list zdefiniowanych w nAxiom.
3. Wybierz z listy widok listy, który ma być otwierany po kliknięciu definiowanej pozycji menu.
4. (Opcjonalnie) Z listy `Filtr` wybierz predefiniowany filtr, który będzie stosowany po wyświetleniu listy. Z listy `Konfiguracja` wybierz konfigurację wyświetlania listy.
5. Kliknij przycisk `Utwórz`.
   W polu `Kod formatki` zostanie wpisany kod widoku listy w formie ciągu `/grid/edit/n`, gdzie `n` to identyfikator widoku listy.

POZYCJE MENU

Aby zarządzać pozycjami menu, należy przejść do edycji wybranej sekcji. W oknie dialogowym `Edycja sekcji menu` u dołu jest wyświetlana lista pozycji menu zdefiniowanych dla tej sekcji.

W ogólności pozycje mają takie same właściwości jak sekcje menu i tak samo się nimi zarządza. Jedna z różnic dotyczy możliwości definiowania zagnieżdżonych pozycji menu. Wymaga to skonfigurowania następujących właściwości:

- `Pozycja nadrzędna`: włączenie tego przełącznika powoduje, że dana pozycja menu może zostać wskazana jako nadrzędna dla innych pozycji w tej sekcji menu.
- `Menu nadrzędne`: lista wyboru pozycji nadrzędnej dla pozycji menu.

- `Domyślnie rozwiń pozycje`: po włączeniu wyświetla pozycje nadrzędne z widocznymi pozycjami podrzędnymi; domyślnie pozycje nadrzędne są zwinięte.
- `Wyświetlaj w sekcji współdzielonej`: (domyślnie wyłączone) pozwala zdecydować, że dana pozycja będzie wyświetlana w sekcji współdzielonej menu, która jest dostępna niezależnie od ustawień trybu pojedynczej aplikacji. W przypadku jeśli w trybie pojedynczej aplikacji będzie wybrana aplikacja macierzysta danej pozycji menu, ta pozycja będzie wyświetlana w „normalnej" sekcji menu, a nie w sekcji współdzielonej. Sekcja współdzielona nie ma wyświetlonej nazwy.

Powiązane tematy:

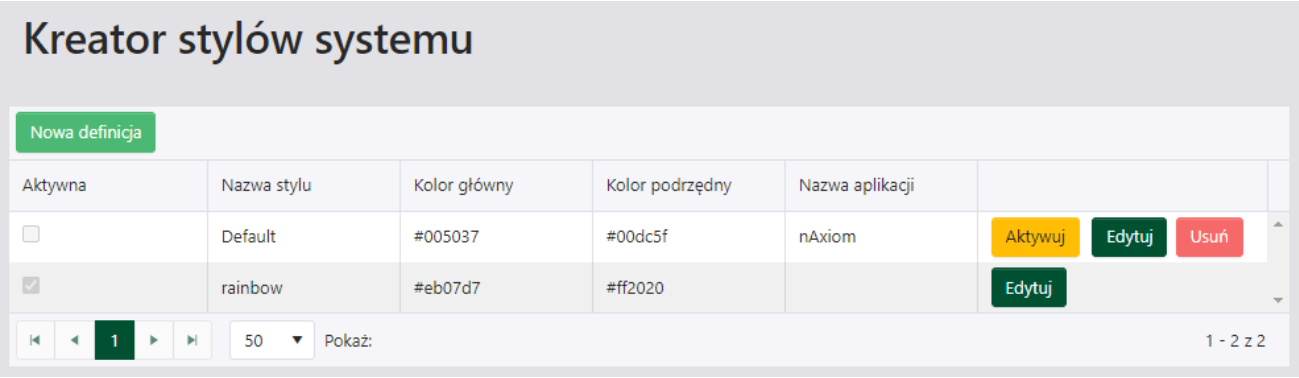
- Role biznesowe w nAxiom
- Tryb pojedynczej aplikacji

4.8. KREATOR STYLÓW SYSTEMU

Podsumowanie:
W nAxiom można tworzyć definicje stylu określające wartości niektórych atrybutów wizualnych elementów interfejsu widocznych we FrontSPA. Ustawienia te obejmują między innymi logo, kolor główny, kolory zależne, odstępy między elementami i rozmiary czcionek. Niektóre atrybuty określane w definicji stylu można nadpisywać w definicji indywidualnych obiektów. Na przykład kolor główny określa domyślny tła przycisku, ale kolor tła można definiować indywidualnie dla przycisków.

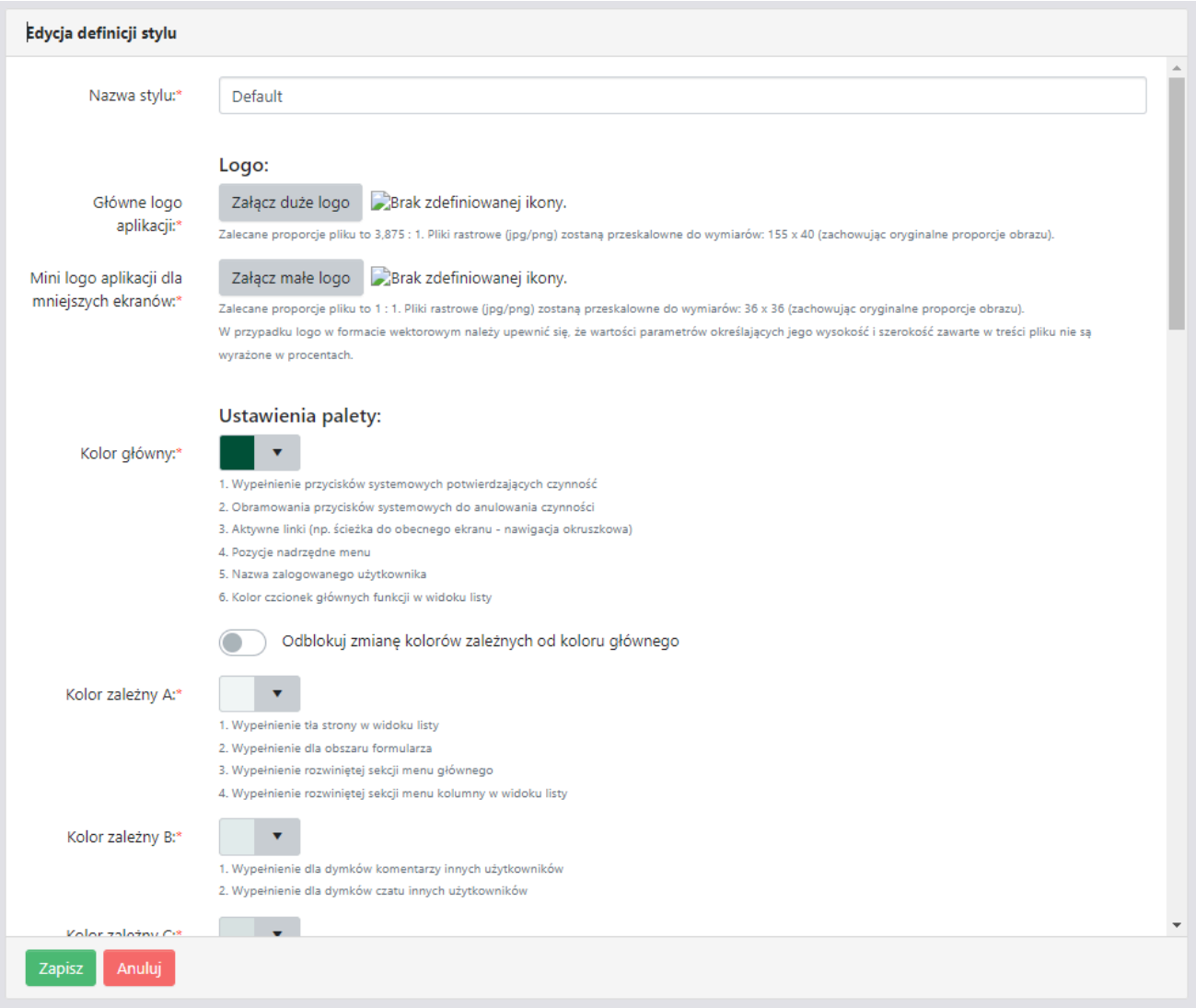
☛ *AdminSPA* > `INTERFEJS UŻYTKOWNIKA` > `Kreator stylów`

Kliknięcie polecenia `Kreator stylów` wyświetla listę zdefiniowanych stylów. Standardowo na liście znajduje się definicja stylu o nazwie *Default*. Po zdefiniowaniu nowej definicji stylu, w kolumnie przycisków dla tego stylu są widoczne przyciski `Aktywuj` i `Usuń`. Aby zastosować definicję we FrontSpa, należy kliknąć przycisk `Aktywuj` w wierszu tej definicji. Zmiany wprowadzone w aktywnej definicji są widoczne po odświeżeniu strony w przeglądarce.



Lista definicji stylów

Kliknięcie przycisku `Nowa definicja` wyświetla stronę ustawień definicji stylu. Niektóre z tych ustawie są oznaczone jako obowiązkowe i muszą zostać ustawione, aby można było zapisać definicję.



Edycja definicji stylu

Aby utworzyć nową definicję stylu, należy określić następujące ustawienia:

- `Nazwa stylu`: nazwa definicji stylu.

- `Główne logo aplikacji`: logo wyświetlane w lewym górnym rogu FrontSPA i AdminSPA; oryginalne logo nAxiom znajduje się w lokalizacji */folder-instalacyjny-naxiom/**front/assets/img/brand/logo.svg***.
- `Mini logo aplikacji dla mniejszych ekranów`: logo wyświetlane w lewym górnym rogu FrontSPA i AdminSPA na ekranach o szerokości poniżej 992 px; oryginalne logo nAxiom znajduje się w lokalizacji */folder-instalacyjny-naxiom/**front/assets/img/brand/mini-logo.svg***.
- `Kolor główny`: podstawowy kolor używany dla wielu elementów interfejsu, między innymi jako domyślne tło przycisków.
- `Odblokuj zmianę kolorów zależnych...`: po włączeniu pozwala na dowolne zmiany kolorów zależnych; domyślnie wyłączony.
- `Kolor zależny A/B/C`: kolory uzyskiwane z koloru głównego przez zastosowanie do nich wartości *krycia* odpowiednio: 4%, 10% i 15%.
- `Kolor akcji`: między innymi kolor linków z aktywnym fokusem.
- `Wyświetlanie kontrolek`: pozwala określić kolor wypełnienia i kolor ramki osobno w trybie edytowalnym i osobno w trybie tylko do odczytu; dodatkowo dla trybu tylko do odczytu można określić kolor czcionki.
- `Wygląd przycisków w trybie tylko do odczytu na listach i formularzach`: kolor wypełnienia, czcionki i ikony dla nieaktywnych przycisków.
- `Zaokrąglenia elementów`: określa promień zaokrąglenia elementów interfejsu użytkownika; obsługiwane są wszystkie jednostki używane w CSS; wartość tej właściwości można określić osobno dla następujących grup elementów:
    - Przyciski, kontrolki edycyjne, elementy list i sekcji systemowych,
    - Etykieta akronimu dokumentu biznesowego
    - Pozostałe elementy

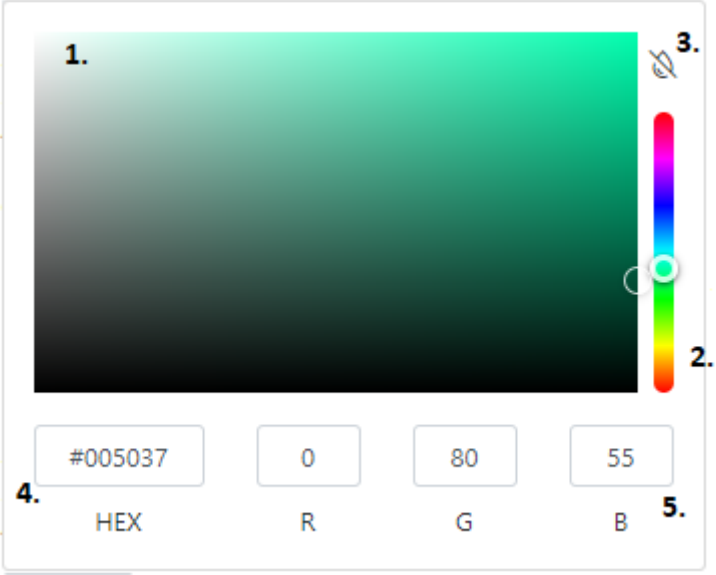
    > **UWAGA**
    > Nie zaleca się stosowania promieni zaokrągleń przycisków większych niż 0,5 rem (lub równoważnych w innych jednostkach) ze względu na nieprawidłowe wyświetlanie innych elementów interfejsu użytkownika, np. list rozwijanych.

- `Odstępy między elementami`: pozwala określić odstępy między rekordami listy, polami formularza, pozycjami menu, a także odstępy w poziomie i w pionie od tekstu na przycisku (paddingi).

- `Ustawienia czcionek`: pozwala wybrać podstawową i pomocniczą rodzinę czcionek oraz określić rozmiary czcionek dla etykiet pól formularza, wartości w kontrolkach i polach edycyjnych, nazwy menu, pozycji nadrzędnej menu i pozycji menu, napisu na przycisku oraz; w tej sekcji można także określić domyślny rozmiar ikon na przyciskach.

SELEKTOR KOLORU

Do ustawiania koloru w Kreatorze stylów służy selektor koloru, który składa się z elementów opisanych poniżej.



Selektor koloru

1. Paleta odcieni
2. Suwak kolorów
3. Ikona braku koloru
4. Szesnastkowy kod koloru
5. Wartości *R*, *G* i *B* koloru.

5. PROCESY BIZNESOWE

Platforma nAxiom została zbudowana i jest rozwijana z myślą o tworzeniu aplikacji, które służą do obsługi procesów w organizacjach. Przez proces rozumie się uporządkowany ciąg czynności prowadzących do osiągnięcia zdefiniowanego celu. Przedmiotem procesu może być dowolny obiekt materialny lub abstrakcyjny: maszyna, materiał, projekt techniczny, faktura. W aplikacji do obsługi procesu taki przedmiot jest reprezentowany przez zestaw danych zapisywanych w specjalnie zdefiniowanej tabeli (patrz DANE). Taki zestaw danych dla konkretnego przedmiotu procesu to *dokument*. Czynności wykonywane przez

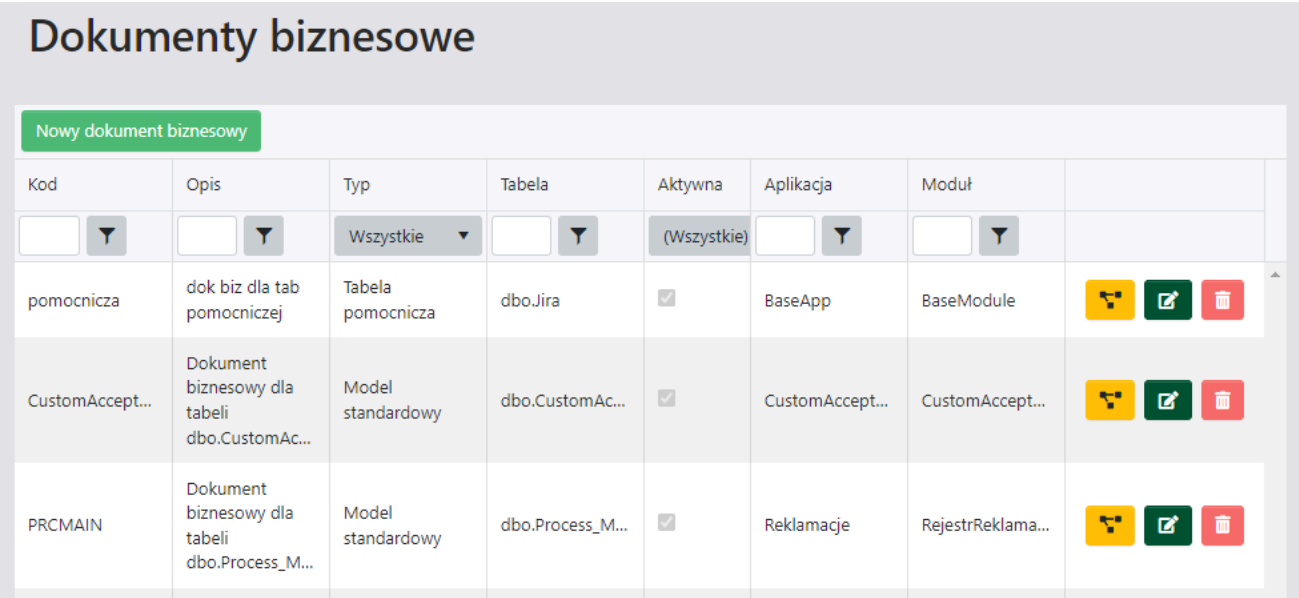
operatorów w procesie powodują zmianę stanu przedmiotu procesu. W aplikacji nAxiom te stany są opisywane przez *statusy*. Natomiast kolejność wykonywania tych czynności określają *przejścia statusów*, które definiują *diagram procesu* (tzw. *workflow*), to jest model przebiegu procesu. Elementem, który łączy te wszystkie składniki — model danych, statusy i ich przejścia — w całość jest w nAxiom *definicja dokumentu*. Można powiedzieć, że definicja dokumentu biznesowego to inaczej definicja klasy obiektów będących przedmiotami danego procesu.

W kolejnych rozdziałach tej części opisano szczegółowo wszystkie składniki używane w nAxiom do modelowania procesów zgodnie z układem poleceń menu `PROCESY BIZNESOWE` w aplikacji *AdminSPA*.

5.1. DOKUMENTY BIZNESOWE (DEFINICJE DOKUMENTÓW)

Podsumowanie:
Definicja dokumentu to w nAxiom deklaracja klasy dokumentów; atrybuty obiektów tej klasy są przechowywane w tabeli dokumentu biznesowego. W definicji dokumentu, oprócz tabeli źródłowej, określa się między innymi szablony uprawnień ACL dla nowych dokumentów, opcje blokowania edytowanych dokumentów, audytowane zdarzenia w toku przetwarzania dokumentów, powiadomienia o zbliżającym się lub przekroczonym terminie wykonania kroku procesu dla dokumentu, akcje wykonywane dla określonych zdarzeń w toku przetwarzania dokumentu i grupy interesariuszy.

☛ *AdminSPA* > `PROCESY BIZNESOWE` > `Dokumenty biznesowe`



Lista Dokumenty biznesowe

W standardowym procesie budowy modelu procesu w nAxiom definicja dokumentu jest tworzona automatycznie poleceniem `Generuj elementy aplikacji` uruchamianym dla wybranego modelu danych w Kreatorze modeli danych. W określonych przypadkach konsultant może samodzielnie utworzyć taką definicję Polecenie `PROCESY BIZNESOWE` > `Dokumenty biznesowe` wyświetla okna `Dokumenty biznesowe`. W tym oknie znajduje się lista definicji dokumentów. Na liście, w wierszu każdej definicji z prawej strony znajdują się następujące ikony:

wyświetla diagram procesu powiązany z definicją dokumentu w edytorze diagramów procesów.

wyświetla okno dialogowe `Edycja definicji dokumentu`.

służy do usunięcia definicji dokumentu; nie można usunąć definicji, dla której istnieją powiązane obiekty (widok listy, formularz itp.) i/lub dokumenty.

Nad listą z lewej strony znajduje się przycisk `Nowy dokument biznesowy`, którego kliknięcie wyświetla okno dialogowe `Tworzenie definicji dokumentu`. Właściwości dostępne na poszczególnych kartach tego okna opisano w kolejnych tematach:

- Definicja dokumentu, Ogólne
- Definicja dokumentu, Blokady
- Definicja dokumentu, Audyt
- Definicja dokumentu, Uprawnienia statusów
- Definicja dokumentu, Uprawnienia — brak odczytu
- Definicja dokumentu, Eskalatory procesu
- Definicja dokumentu, Zdarzenia
- Definicja dokumentu, Interesariusze
- Definicja dokumentu, Pomoc dla użytkownika

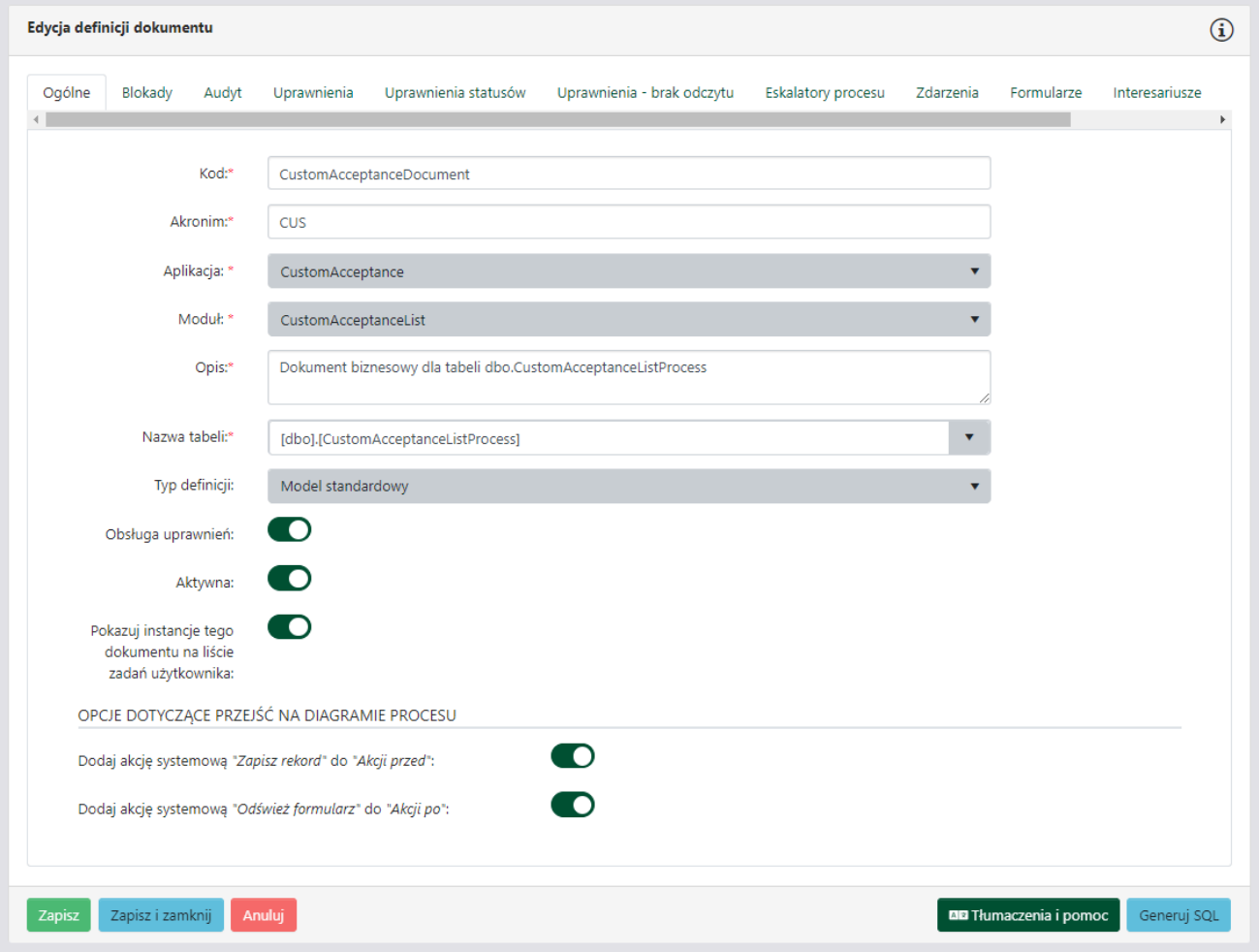
Powiązane tematy:

- Kreator modeli danych, generowanie elementów aplikacji
- Edytor diagramów procesów

5.1.1. Karta Ogólne

Podsumowanie:
Na karcie `Ogólne` określa się podstawowe elementy definicji dokumentu.

☛ *AdminSPA* > `PROCESY BIZNESOWE` > `Dokumenty biznesowe` > `Nowy/Edytuj`



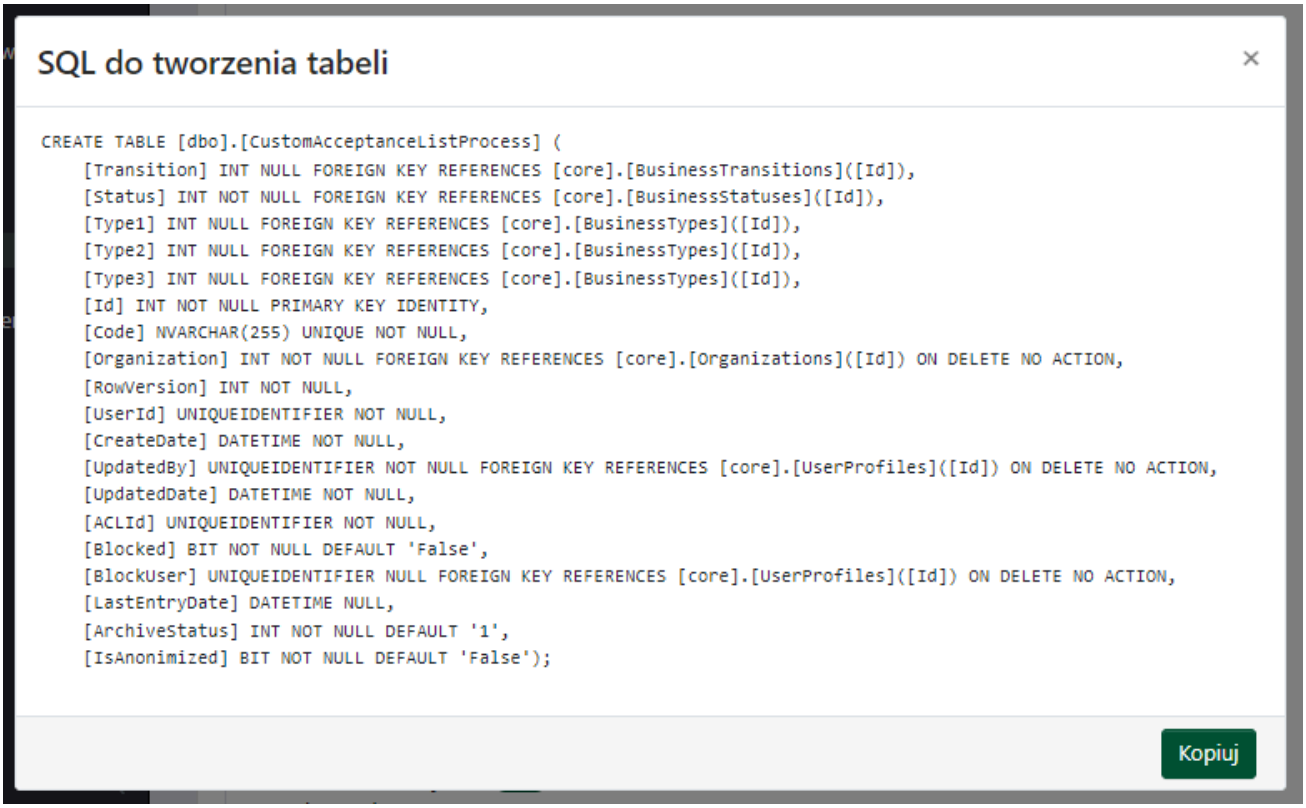
Karta Ogólne

Na karcie `Ogólne` w oknie `Tworzenie/Edycja definicji dokumentu` znajdują się następujące pola i elementy sterujące:

- `Kod`: dowolny ciąg znaków identyfikujący tworzoną definicję dokumentu.
- `Akronim`: skrót identyfikujący kategorię dokumentów biznesowych, którego można używać w nagłówku formularza; akronim może mieć od 1 do 4 znaków (litery, cyfry, znaki specjalne); dla istniejących definicji dokumentów biznesowych akronim jest tworzony automatycznie z trzech pierwszych znaków kodu dokumentu. W taki sam sposób są tworzone domyślne akronimy w procesie tworzenia elementów aplikacji. Automatycznie utworzone akronimy można zmieniać. Akronimy są zapisywane w kolumnie *Acronym* w tabeli *[core].[BusinessDocuments]*.

- `Aplikacja`: aplikacja, do której zostanie przypisana tworzona definicja dokumentu.
- `Moduł`: moduł aplikacji, do którego zostanie przypisana tworzona definicja dokumentu.
- `Opis`: opis tworzonej definicji dokumentu.
- `Nazwa tabeli`: tabela dokumentu biznesowego, dla której ma zostać utworzona definicja dokumentu.
- `Typ definicji`: jedna z wartości dostępnych na liście:
    - `Model uproszczony`: uproszczony model danych, który od modelu standardowego różni się brakiem statusu i typu dokumentu biznesowego
    - `Model standardowy`: standardowy model danych, tabela w tym modelu, oprócz kolumn zdefiniowanych przez użytkownika, zawiera szerego kolumn systemowych, w tym kolumnę statusu i kolumny typów biznesowych
    - `Tabela pomocnicza`: tabela w bazie danych służąca do przechowywania danych pomocniczych; wybór tej wartości powoduje wyświetlenie dodatkowego pola:
        - `Kolumna ID tabeli`: podaj nazwę kolumny tabeli zewnętrznej, w której znajduje się identyfikator (klucz główny).
- `Obsługa uprawnień`: włącz lub wyłącz obsługę uprawnień dla dokumentów tworzonych w ramach bieżącej definicji dokumentu; po włączeniu w oknie są wyświetlane dwie zakładki `Uprawnienia` i `Uprawnienia statusów`.
- `Aktywna`: wyłącz ten przełącznik, aby bieżąca definicja dokumentu stała się nieaktywna.
- `Pokazuj instancje tego dokumentu na liście zadań użytkownika`: (domyślnie włączone) decyduje o tym, czy instancje tej definicji dokumentu są wyświetlane na liście zadań użytkownika (funkcja systemowa *UserTasks_View*).
- `Dodaj akcję systemową "Zapisz rekord" do "Akcji przed"`: (domyślnie włączone) pozwala zdecydować, czy na początku sekwencji akcji przed zmianą statusu (przejścia na diagramie procesów) ma być automatycznie dodawana akcja systemowa `Zapisz rekord`, która spowoduje zapisanie danych w formularzu.
- `Dodaj akcję systemową "Odśwież formularz" do "Akcji po"`: (domyślnie włączone) pozwala zdecydować, czy do sekwencji akcji po zmianie statusu (przejścia na diagramie procesów) ma być automatycznie dodawana akcja systemowa `Odśwież formularz`, która spowoduje zapisanie danych w formularzu.
- `Zapisz`: kliknij, aby zapisać zmiany.
- `Zapisz zmiany i zamknij`: kliknij, aby zapisać zmiany i zamknąć okno.
- `Anuluj`: kliknij, aby zamknąć okno bez zapisywania zmian.
- `Generuj SQL`: wyświetla okno z zapytaniem SQL *CREATE TABLE*, którego można

użyć do utworzenia tabeli dla danej definicji dokumentu biznesowego; zapytanie pozwala utworzyć tabelę zawierającą tylko kolumny systemowe, trzeba je uzupełnić o definicje kolumn użytkownika; ta funkcja jest przydatna, kiedy użytkownik samodzielnie tworzy definicję dokumentu biznesowego, nie korzystając z polecenia `Utwórz elementy aplikacji` w *Kreatorze modeli danych*. Ten przycisk jest dostępny tylko dla definicji typu `Dokument` i `Katalog`



Zapytanie SQL do tworzenia tabeli

5.1.2. Karta Reguły walidacji

Podsumowanie:
Na karcie `Reguły walidacji` można zdefiniować reguły walidacji dla definiowanych przez użytkownika pól modelu danych (także kilka reguł dla tego samego pola). Nie można definiować reguł walidacji dla pól systemowych.

Reguły walidacji są sprawdzane przed zapisem instancji dokumentu (akcja zapisu, zapis z formularza publicznego, zapis z Public API, zapis instancji dokumentu dla podprocesu, mail monitor, zapis danych dodatkowych użytkownika/organizacji). Reguły walidacji nie są sprawdzane w przypadku akcji importu dokumentów.

☛ *AdminSPA* > `PROCESY BIZNESOWE` > `Dokumenty biznesowe` > `Nowy/Edytuj`

Reguły walidacji tworzy się w trybie edycji inline po kliknięciu przycisku `Dodaj regułę`. Należy wybrać pole, którego dotyczy reguła, operator oraz wartość do porównania

Dla każdej reguły walidacji można dodatkowo zdefiniować warunek sprawdzania danej reguły. Reguła zostanie sprawdzona tylko wtedy, gdy warunek sprawdzania będzie spełniony.

Operatory dostępne w regułach walidacji zależą od typu danych pola. Oprócz standardowych operatorów dla danych liczbowych, tekstowych oraz typu data/godzina dostępny jest np. typ walidacji *Predicate*, który sprawdza czy wartość reguły jest równa 1. Umożliwia to napisanie dowolnego SQL który wykona walidację i zwróci wartość równą 1 w przypadku sukcesu a różną od 1 w przypadku błędu walidacji. Inny przykład typu walidacji to *RegexMatch*, który sprawdza czy wartość dla danego pola modelu spełnia wyrażenie regularne przekazane jako wartość reguły.

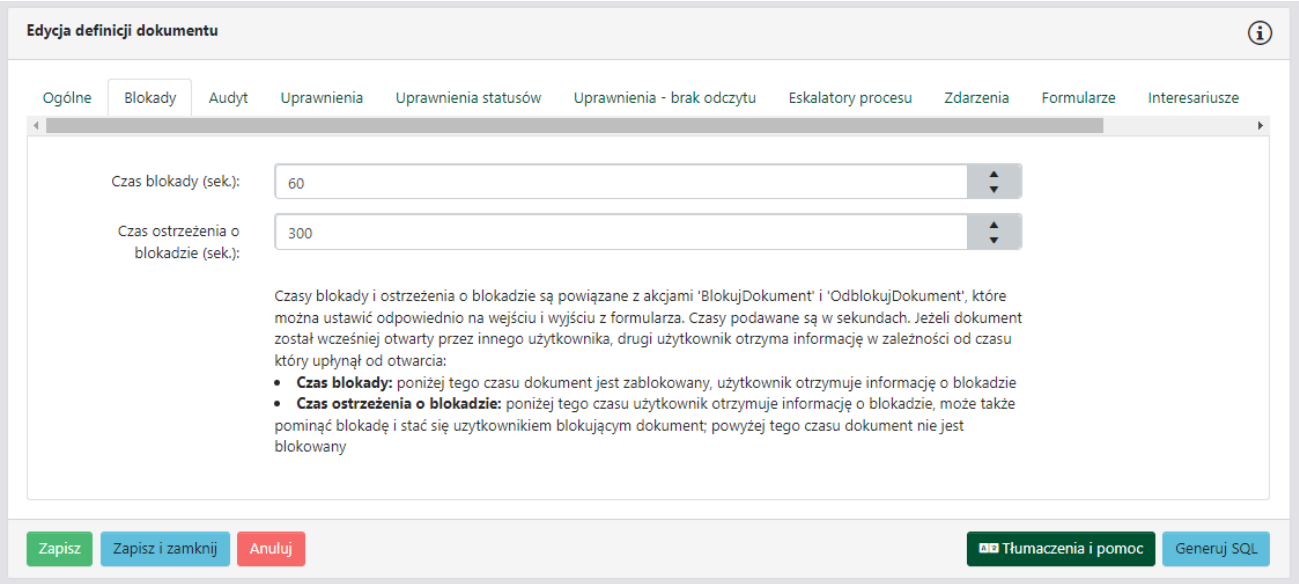
Komunikaty walidacji obsługują tłumaczenia (na język użytkownika lub, w przypadku formularzy publicznych, na język formularza). Nazwy pól których dotyczy dany błąd walidacji pochodzą z tłumaczeń dla pola `Etykieta` z kreatora modeli danych.

Reguły walidacji zaimplementowano w oparciu o bibliotekę .NET FluentValidation.

5.1.3. Karta Blokady

Podsumowanie:
W środowisku produkcyjnym może wystąpić sytuacja, w której co najmniej dwóch użytkowników będzie próbowało jednocześnie edytować ten sam dokument. Dostępna w nAxiom funkcja *blokowania dokumentu* pozwala uniknąć takiego problemu. Służą do tego akcje `Blokuj dokument` i `Odblokuj dokument`. Na karcie `Blokady` można ustawić domyślne czasy dla akcji `Blokuj dokument`.

☛ *AdminSPA* > `PROCESY BIZNESOWE` > `Dokumenty biznesowe` > `Nowy/Edytuj` > `Blokady`



Karta Blokady

- `Czas blokady (sek.)`: czas liczony od otwarcia dokumentu do edycji; przed upływem tego czasu, żaden użytkownik nie może edytować dokumentu; próba otwarcia dokumentu powoduje wyświetlenie komunikatu z informacją o blokadzie.
- `Czas ostrzeżenia o blokadzie (sek.)`: czas liczony od od otwarcia dokumentu do edycji; jeśli próba otwarcia zablokowanego dokumentu będzie miała miejsce po upływie czasu blokady, a przed upływem tego czasu, zostanie wyświetlony komunikat z informacja o blokadzie z możliwością zignorowania blokady; jeśli użytkownik zignoruje blokadę, stanie się użytkownikiem blokującym; po upływie tego czasu dokument zostanie odblokowany.

Powiązane tematy:

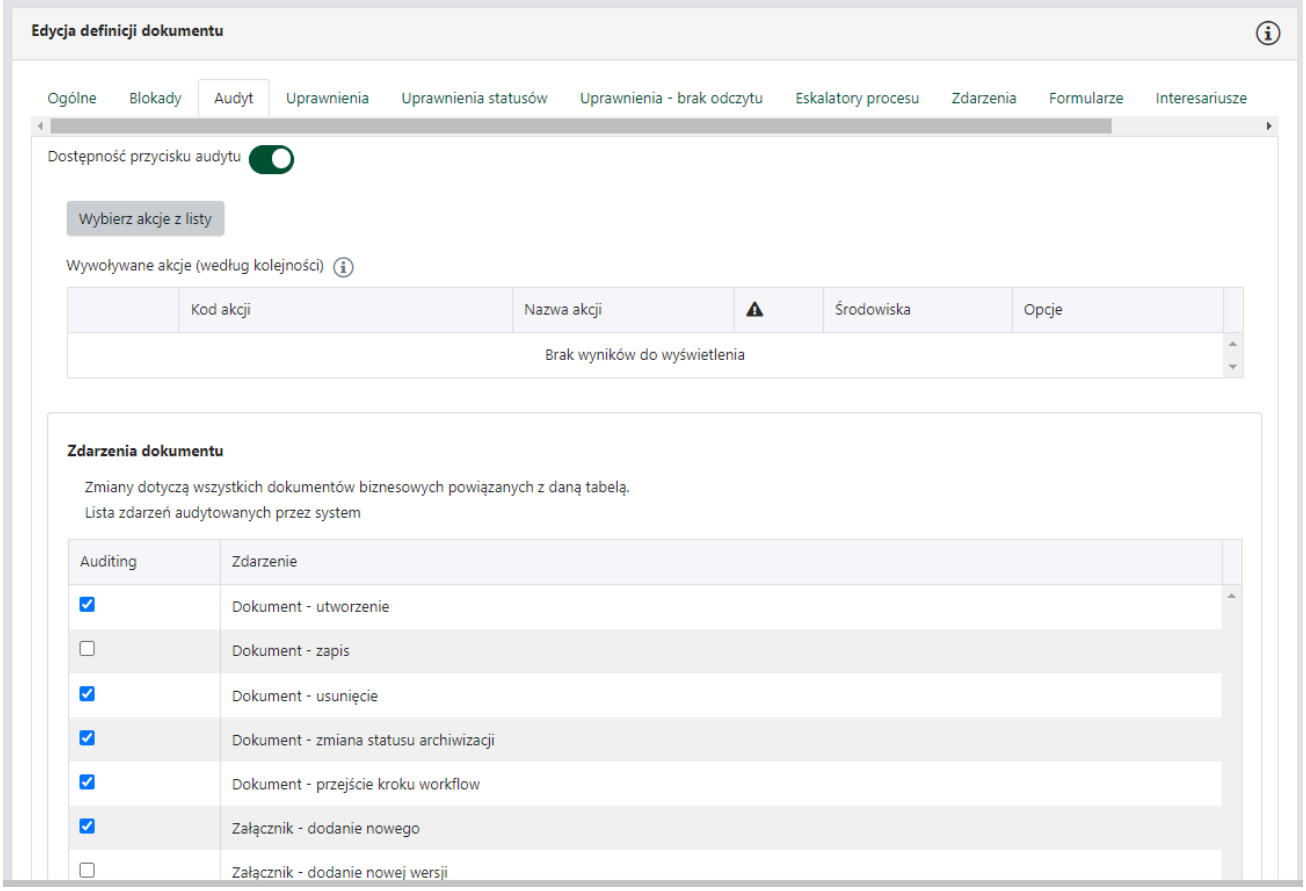
- Właściwości formularza

5.1.4. Karta Audyt

Podsumowanie:
Funkcja audytu służy do rejestrowania zdarzeń występujących w procesie biznesowym oraz zmian danych dokonywanych w tym procesie. W przypadku definicji dokumentu biznesowego tworzonej poleceniem `Generuj elementy aplikacji` w *Kreatorze modelu danych*, domyślnie włączony jest audyt zdarzeń tworzenia dokumentu, usunięcia dokumentu oraz zmiany statusu archiwizacji. Patrz także artykuł Audyt danych biznesowych w nAxiom w witrynie dokumentacji.

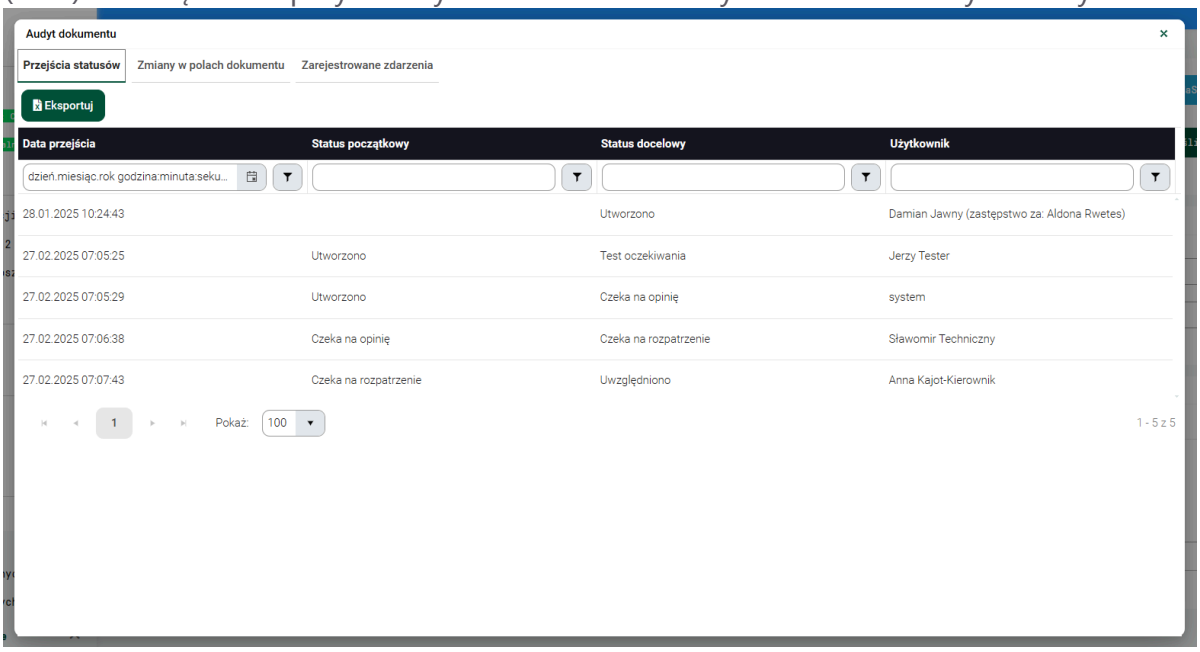
Na karcie `Audyt` określa się ustawienia dotyczące zakresu audytu (kategorii zdarzeń i kolumn tabeli).

☛ *AdminSPA* > `PROCESY BIZNESOWE` > `Dokumenty biznesowe` > `Nowy/Edytuj` > `Audyt`



Karta Audyt — Zdarzenia dokumentu

- `Dostępność przycisku audytu`: określa czy w aplikacji *FrontSPA* w prawym górnym rogu formularza ma być wyświetlany przycisk wyświetlający dane audytowe ( ). Po włączeniu przycisk wyświetla standardowy formularz z danymi audytu.



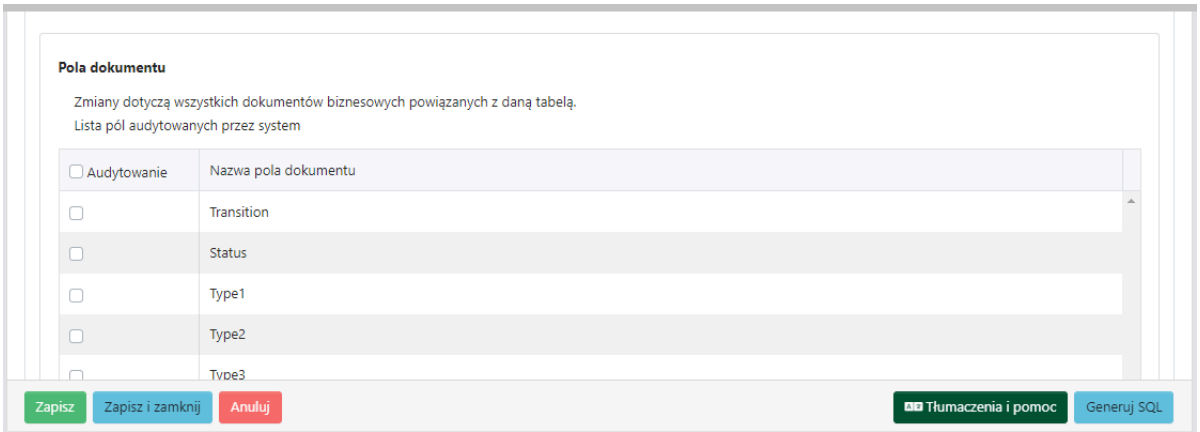
> **UWAGA**
> Dostępność przycisku audytu zależy także od posiadania przez konkretnego użytkownika odpowiedniego uprawnienia PBA.

- `Wybierz akcję z listy`: wybierz z listy akcję otwierania formularza, w którym zostaną wyświetlone dane audytu.

W części `Zdarzenia dokumentu` należy zaznaczyć pola wyboru obok zdarzeń, które mają być rejestrowane. Dostępne są następujące zdarzenia:

- Dokument - utworzenie,
- Dokument - zapis,
- Dokument - usunięcie,
- Dokument - zmiana statusu archiwizacji,
- Dokument - przejście kroku workflow,
- Załącznik - dodanie nowego,
- Załącznik - dodanie nowej wersji,
- Załącznik - nadpisanie,
- Załącznik - usunięcie,
- Załącznik - usunięcie wersji,
- Właściciel - przypisanie,
- Właściciel - zwrócenie do puli,
- Interesariusz - dodanie nowego,
- Interesariusz - usunięcie,
- Eskalatory - wysyłka,
- RODO - akcja anonimizacji danych,

- Dodanie komentarza



W części `Pola dokumentu` należy zaznaczyć pola wyboru obok nazw kolumn tabeli, których zmiany mają być rejestrowane. Domyślnie audyt obejmuje wszystkie kolumny, oprócz kolumn systemowych: *ArchiweStatus*, *Blocked*, *BlockUser*, *LastENtryDate* oraz *IsAnonimized*.

Dane rejestrowane w ramach audytu obejmują między innymi dane użytkownika, który zainicjował zdarzenie/wprowadził zmianę, data wystąpienia, wartość przed zmianą i po zmianie, nazwę kolumny tabeli i wiele innych. Jeśli użytkownik pracuje w zastępstwie za innego użytkownika, jest to również rejestrowane w ramach audytu.

Konfiguracja audytu jest zapisywana w bazie danych w tabelach *core.AuditEvents* (zdarzenia) i *core.AuditColumns* (kolumny). Dane audytu są zapisywane w tabelach *core.AuditEventLogs* i *core.AuditLogs*. Jeżeli wystąpienie zdarzenia wiąże się ze zmianą danych w audytowanych kolumnach, zapisy w obu tabelach są powiązane wartością z kolumny *BatchId*.

Konfiguracja audytu jest przypisana bezpośrednio do tabeli, która jest podstawą dla definicji dokumentu biznesowego. W przypadku, kiedy dla tej samej tabeli zostanie wygenerowany inny dokument biznesowy, konfiguracja audytu dla tej definicji będzie taka sama.

PRZYKŁAD ZAPISU AUDYTU

Dodanie komentarza powoduje zapisanie nowego rekordu w tabeli *[core].[AuditEventLogs]*. W kolumnie *Value* zapisywane są dane dotyczące zdarzenia w formacie JSON.

```
{
    "OldStatusId":null,
    "NewStatusId":null,
    "StakeholderId":null,
    "UserProfileIdList":null,
    "UserProfileId":null,
    "ArchiveStatusId":null,
    "Fields":null,
    "Attachment":null,
    "Comment":
    {
        "CommentId":20,
        "Message":"Treść komentarza"
    }
}
```

Dla zdarzenia dodania/usunięcia w grupie interesariuszy w kolumnie *Value* zapisywany jest obiekt JSON, który zawiera trzy dodatkowe flagi:

```
{"Stakeholder": {
  "StakeholderId":9,
  "OrgUnitRoleIds":[],
  "AllOrgUnitRolesDeleted":true,
  "AllGlobalRolesDeleted":false,
  "AllUsersDeleted":false
}
}
```

Dla zdarzenia usunięcia indywidualnych członków z grupy interesariuszy oraz dla zdarzenia dodania do grupy interesariuszy wszystkie flagi mają wartość false.

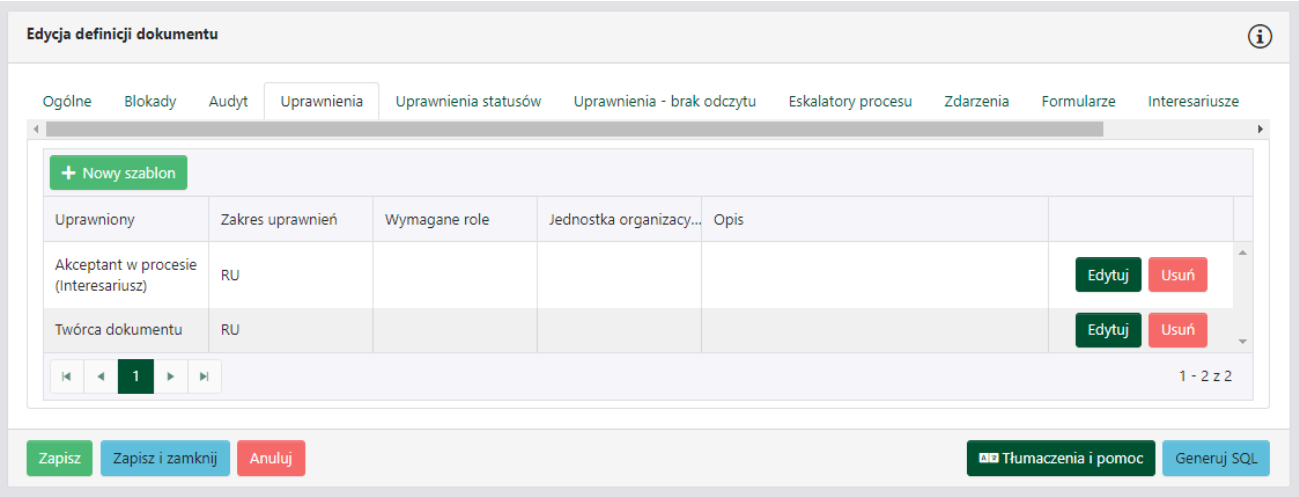
Powiązane tematy:

- Ustawienia systemu, Audyt

5.1.5. Karty Uprawnienia, Uprawnienia statusów

Podsumowanie:
Na kartach `Uprawnienia` i `Uprawnienia statusów` definiuje się szablony uprawnień do wszystkich dokumentów tworzonych w ramach danej definicji dokumentu. Szablony obu kategorii definiuje sie w ten sam sposób. Różnica polega na tym, że na karcie `Uprawnienia statusów` nowy szablon tworzy się w sekcji konkretnego statusu zdefiniowanego dla tego dokumentu.

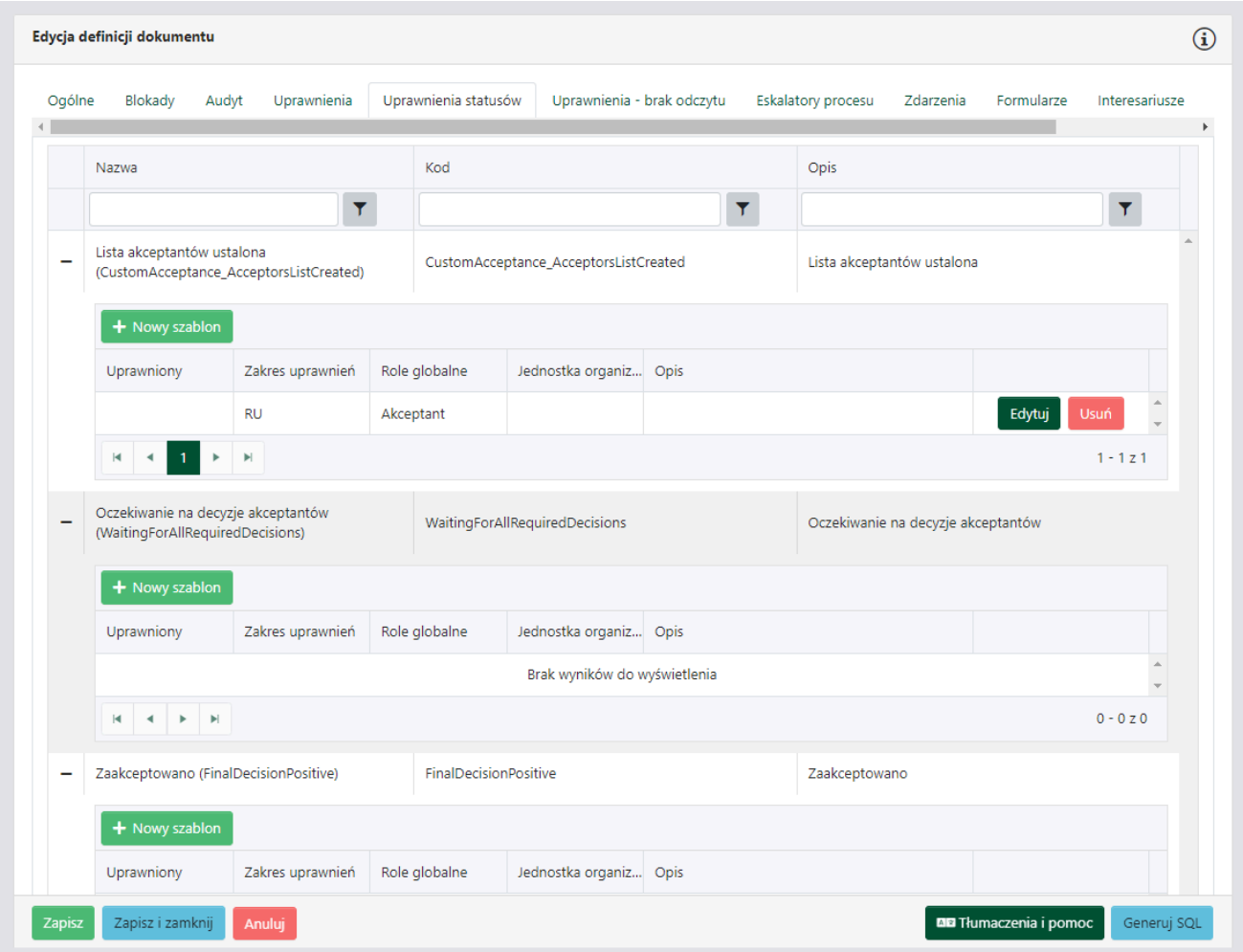
☛ *AdminSPA* > `PROCESY BIZNESOWE` > `Dokumenty biznesowe` > `Nowy/Edytuj` > `Uprawnienia`/`Uprawnienia statusów`



Karta Uprawnienia

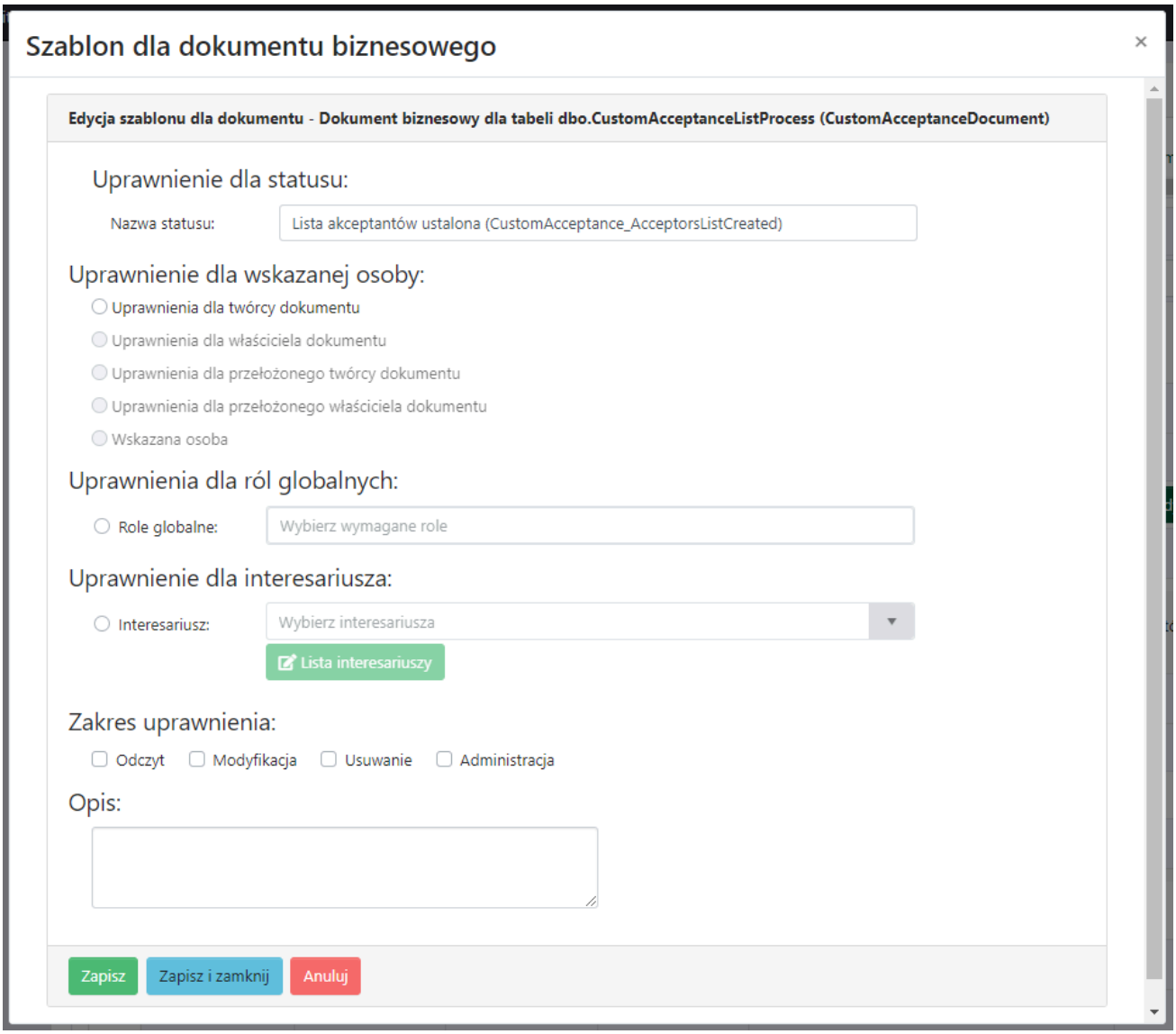
> **INFO**
> Karty `Uprawnienia` i `Uprawnienia statusów` są widoczne dopiero po włączeniu obsługi uprawnień; patrz *Powiązane tematy*.

Na karcie `Uprawnienia statusów` zdefiniowane szablony są pogrupowane według statusów danej definicji dokumentu.



Karta Uprawnienia statusów

Kliknięcie przycisku `Nowy szablon` wyświetla okno dialogowe szablonu uprawnień.



Szablon dla dokumentu biznesowego

- `Nazwa statusu`: pole tylko do odczytu z nazwą statusu, dla którego będzie obowiązywał szablon; wyświetlane tylko na karcie `Uprawnienia statusów`.
- `Uprawnienia dla twórcy dokumentu`: szablon uprawnień będzie obowiązywał dla użytkownika, który utworzył dokument.
- `Wymagane role`: szablon uprawnień będzie obowiązywał dla użytkowników przypisanych do ról wybranych z listy; jeśli zostanie wybranych kilka ról, uprawnionymi będą użytkownicy posiadający co najmniej jedną z wybranych ról.
- `Interesariusz`: szablon uprawnień będzie obowiązywał dla użytkowników, którzy w trakcie procesu biznesowego zostaną przypisani do grupy interesariuszy o nazwie podanej w polu obok.
- `Lista interesariuszy`: wyświetla okno dialogowe, w którym można zarządzać

kategoriami interesariuszy, tak jak na karcie `Interesariusze`.

- `Zakres uprawnienia`:
  - `Odczyt`: nadaje (zaznaczone) lub odbiera (niezaznaczone) uprawnienia do odczytu dokumentu.
  - `Modyfikacja`: nadaje (zaznaczone) lub odbiera (niezaznaczone) uprawnienia do modyfikacji dokumentu.
  - `Usuwanie`: nadaje (zaznaczone) lub odbiera (niezaznaczone) uprawnienia do usunięcia dokumentu.
  - `Administracja`: nadaje (zaznaczone) lub odbiera (niezaznaczone) uprawnienia do zarządzania szablonami uprawnień dla dokumentu.

UWAGA
Co najmniej jedno pole wyboru w sekcji `Zakres uprawnienia` musi zostać zaznaczone.

- `Opis`: pozwala dodatkowo opisać tworzony szablon uprawnień.
- `Zapisz`: zapisuje dokonane zmiany szablonu.
- `Zapisz i zamknij`: zapisuje dokonane zmiany szablonu i zamyka okno dialogowe.
- `Anuluj`: zamyka bieżące okno.

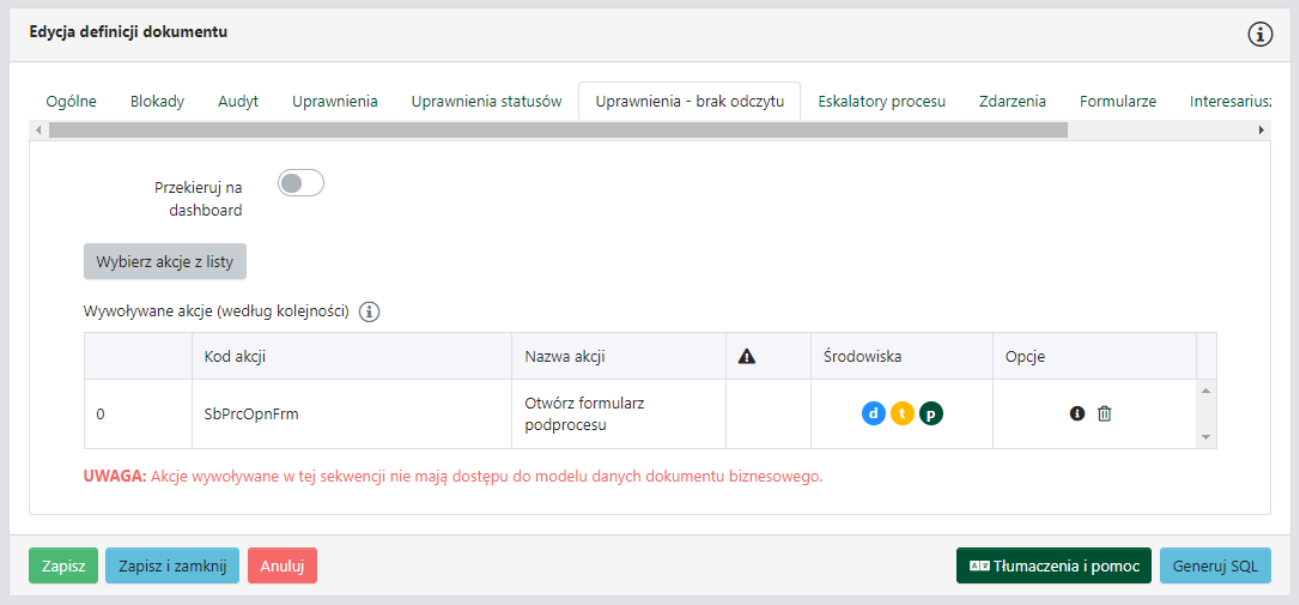
Powiązane tematy:

- Uprawnienia ACL do dokumentów
- Nowy szablon uprawnień do dokumentu
- Włączanie obsługi uprawnień
- Administrowanie uprawnieniami ACL
- Mechanika uprawnień ACL (dokumenty)
- Uprawnienia do rekordów

5.1.6. Karta Uprawnienia — brak odczytu

Podsumowanie:
Karta `Uprawnienia – brak odczytu` pozwala określić zachowanie nAxiom w przypadku, kiedy użytkownik próbuje wyświetlać dokumenty bez uprawnień do ich odczytu.

☛ *AdminSPA* > `PROCESY BIZNESOWE` > `Dokumenty biznesowe` > `Nowy/Edycja` > `Uprawnienia – brak odczytu`



Karta Uprawnienia — brak odczytu

- `Przekieruj na dashboard`: domyślnie włączony, powoduje on przekierowanie użytkownika na stronę startową aplikacji.
- `Wybierz akcje z listy`: (przycisk wyświetlany po wyłączeniu przełącznika `Przekieruj na dashboard`) wyświetla listę akcji, które zostaną wykonane.

UWAGA
Akcje nie powinny odwoływać się do modelu danych dokumentu (zmienne typu *{@nazwa}*), ponieważ nie mają dostępu do tego modelu.

5.1.7. Karta Eskalatory procesu

Podsumowanie:
Eskalatory to powiadomienia (e-mail i systemowe) o zbliżającym się lub przekroczonym terminie wykonania określonych czynności w przebiegu procesu biznesowego. Czasy wysyłania eskalatorów i ich adresatów definiuje się dla indywidualnych kroków na diagramie procesu biznesowego (*workflow*) — jest to możliwe tylko dla kroków, dla których zdefiniowano termin wykonania. Powiadomienia systemowe są sygnalizowane ikoną dzwonka w nagłówku witryny ( ) i wyświetlane w panelu powiadomień po kliknięciu tej ikony.

W definicji dokumentu biznesowego na karcie `Eskalatory procesu` określa się akcję e-mail i treść powiadomienia systemowego osobno dla eskalatorów wykonywanych przed terminem i po terminie.

☛ *AdminSPA* > `PROCESY BIZNESOWE` > `Dokumenty biznesowe` > `Nowy/Edytuj` > `Eskalatory procesu`

- `Akcja email dla eskalatorów – przed terminem`: wybierz zdefiniowaną akcję e-mail wysyłki, która będzie wykonywana dla wszystkich eskalatorów wysyłanych przed upływem terminu kroku procesu.
- `Treść notyfikacji systemowej – przed terminem`: wpisz treść powiadomienia systemowego, która będzie używana dla wszystkich eskalatorów wysyłanych przed upływem terminu kroku procesu.
- `Akcja email dla eskalatorów – po terminie`: wybierz zdefiniowaną akcję e-mail wysyłki, która będzie wykonywana dla wszystkich eskalatorów wysyłanych po upływie terminu kroku procesu.
- `Treść notyfikacji systemowej – po terminie`: wpisz treść powiadomienia systemowego, która będzie używana dla wszystkich eskalatorów wysyłanych po upływie terminu kroku procesu.

W treści eskalatorów można używać następujących parametrów:

- *{@TaskOwner}*: imię i nazwisko użytkownika przypisanego do danego kroku procesu jako *właściciel zadania*
- *{@TaskDeadline}*: termin wykonania danego kroku procesu
- *{@TaskDeadlineTimeLeft}*: czas pozostały do upływu terminu dla kroku procesu
- *{@TaskDeadlineDelay}*: czas, o jaki przekroczono termin dla kroku procesu

Można także używać parametrów z modelu danych w formacie *{@NazwaPola}*.

WAŻNE
Po zdefiniowaniu eskalatorów wymagany jest restart witryny i serwisu Taskservice (puli aplikacji).

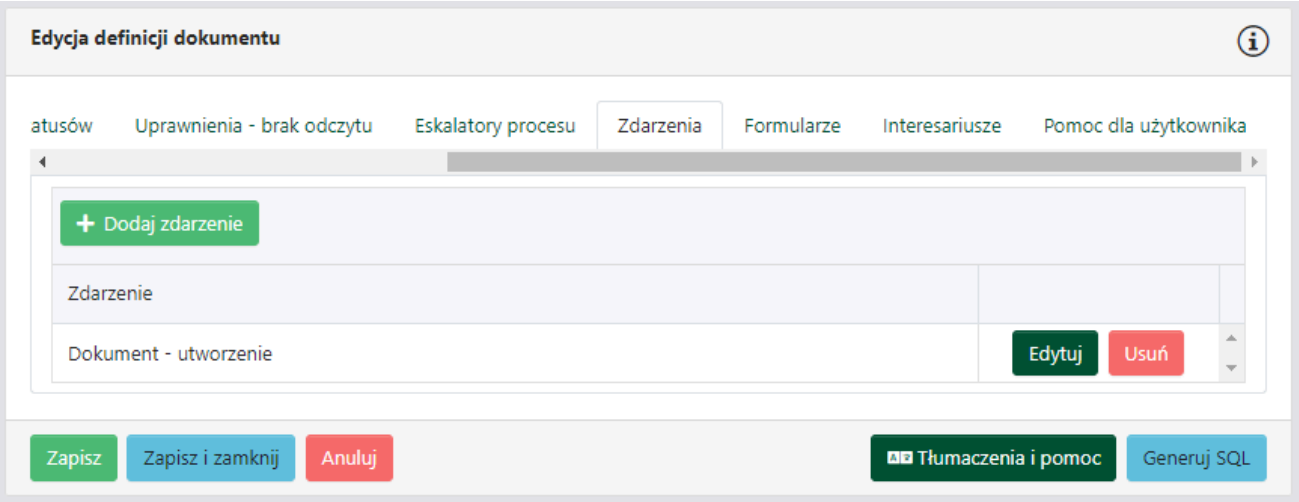
Powiązane tematy:

- Diagram procesów, zadanie ręczne

5.1.8. Karta Zdarzenia

Podsumowanie:
W przebiegu procesu biznesowego w nAxiom wyróżnia się szereg zdarzeń, których wystąpienie może stanowić wyzwalacz do wykonania sekwencji akcji. Na przykład dodanie załącznika może wyzwolić akcję rozpoznawania danych OCR i akcję aktualizacji dokumentu na podstawie rozpoznanych danych. Po wystąpieniu zdarzenia przypisane akcje są wykonywane asynchronicznie, bez oczekiwania na rezultat akcji. W akcjach można odwoływać się do wartości z bieżącej instancji dokumentu, używając składni *{@PoleFormularza}*.
Na karcie `Zdarzenia` definiuje się akcje dla zdarzeń w procesach.

☛ *AdminSPA* > `PROCESY BIZNESOWE` > `Dokumenty biznesowe` > `Nowy/Edytuj` > `Zdarzenia`



Karta Zdarzenia

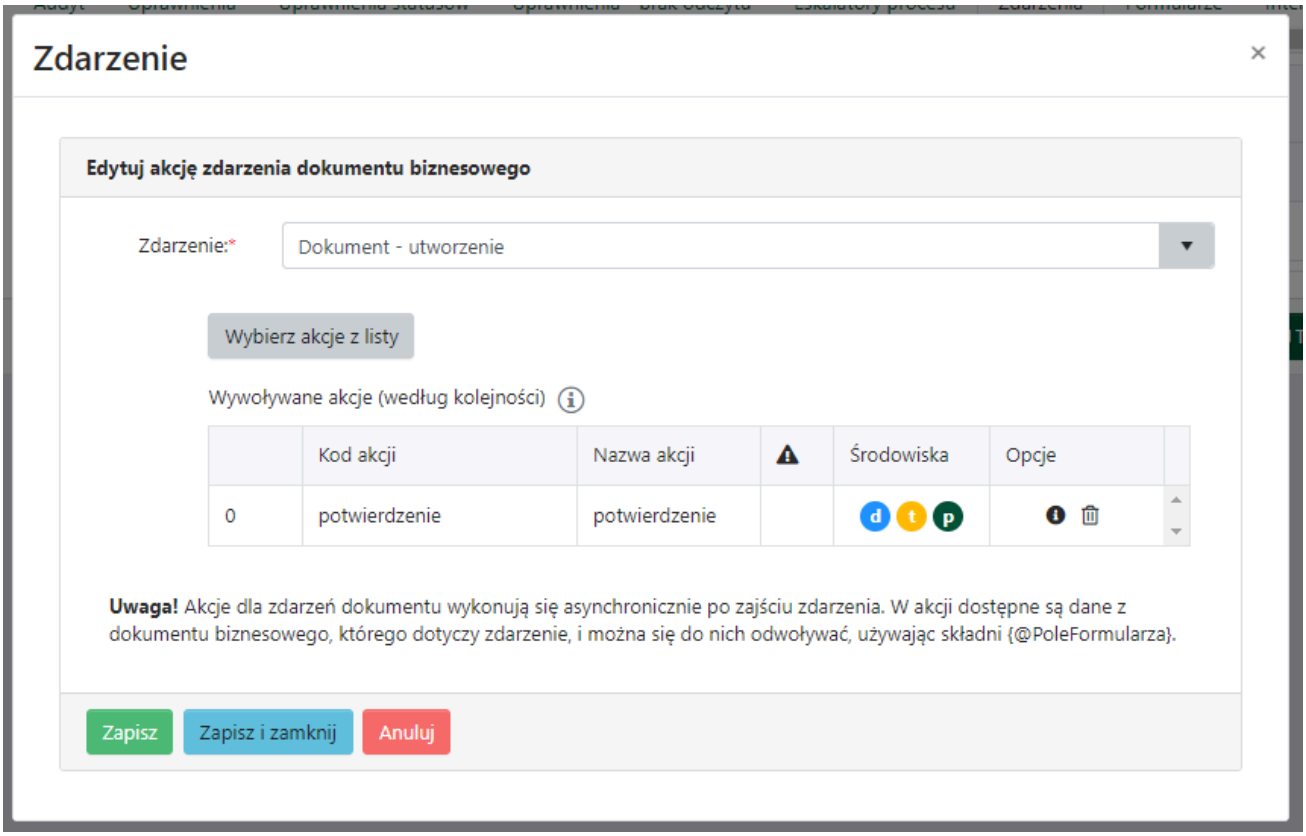
Karta zawiera listę zdarzeń, dla których zdefiniowano akcje wykonywane po wystąpieniu zdarzenia. Dla każdego typu zdarzenia można zdefiniować tylko jedną sekwencję akcji. Na karcie są dostępne przyciski:

- `Dodaj zdarzenie`: otwiera okno `Edytuj akcję zdarzenia dokumentu`

`biznesowego`, w którym można wybrać zdarzenie i przypisać mu sekwencję akcji.

- `Edytuj`: otwiera okno `Edytuj akcję zdarzenia dokumentu biznesowego`, w którym można edytować sekwencję akcji przypisaną do wybranego zdarzenia.
- `Usuń`: usuwa zdarzenie z listy

OKNO EDYTUJ AKCJĘ ZDARZENIA DOKUMENTU BIZNESOWEGO



Akcja zdarzenia

Aby przypisać sekwencję akcji do zdarzenia, wybierz rodzaj zdarzenia z listy `Zdarzenie`, a następnie dwukrotnie kliknij akcje, które chcesz wykonywać wraz z wystąpieniem danego zdarzenia. Dostępne są akcje, które zostały zdefiniowane a aplikacji. W tym oknie nie ma możliwości edycji wybranych akcji.

Na liście `Zdarzenie` są dostępne następujące zdarzenia:

- Dokument - utworzenie,
- Dokument - zapis,
- Dokument - usunięcie,
- Dokument - zmiana statusu archiwizacji,
- Dokument - przejście kroku workflow,
- Załącznik - dodanie nowego,

- Załącznik - dodanie nowej wersji,
- Załącznik - nadpisanie,
- Załącznik - usunięcie,
- Załącznik - usunięcie wersji,
- Właściciel - przypisanie,
- Właściciel - zwrócenie do puli,
- Interesariusz - dodanie nowego,
- Interesariusz - usunięcie,
- Eskalatory - wysyłka,
- RODO - akcja anonimizacji danych,
- Dodanie komentarza

UWAGA
W akcjach wyzwalanych zdarzeniami dotyczącymi załączników są obsługiwane dodatkowe parametry:

- *{@_Event.AttachmentId}*: identyfikator załącznika
- *{@_Event.AttachmentFileExtention}*: rozszerzenie nazwy pliku załącznika (bez kropki)
- *{@_Event.AttachmentFileSize}*: rozmiar pliku załącznika

Jeśli użytkownik doda, zaktualizuje lub usunie więcej niż jeden załącznik w ramach jednej operacji, zdarzenie dotyczące załącznika wystąpi osobno dla każdego załącznika i za każdym razem wyzwoli przypisane akcje. Dla każdego załącznika wartości kontekstu zdarzenia zostaną zaktualizowane.

UWAGA
W akcjach wyzwalanych zdarzeniami dotyczącymi załączników są obsługiwane dodatkowe parametry:

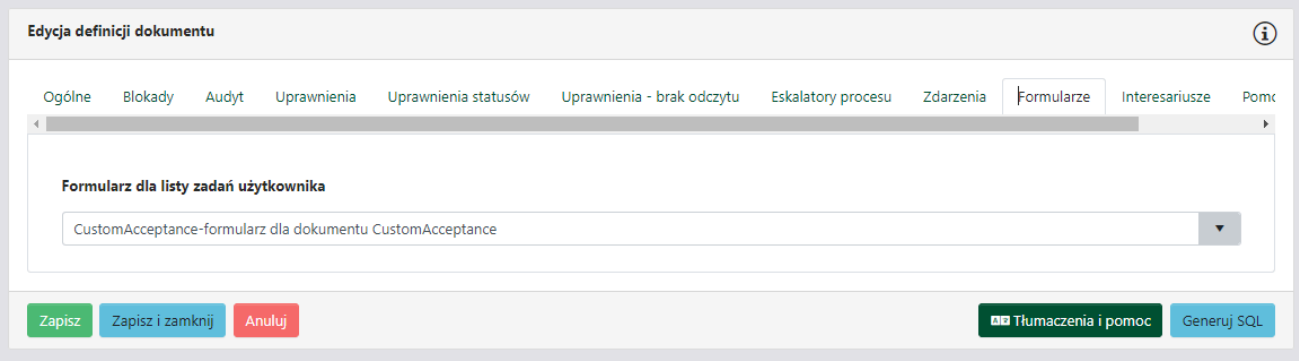
- *{@_Event.AttachmentId}*: identyfikator załącznika
- *{@_Event.AttachmentCategoryId}*: kategoria załączników
- *{@_Event.AttachmentFileExtention}*: rozszerzenie nazwy pliku załącznika (bez kropki)
- *{@_Event.AttachmentFileSize}*: rozmiar pliku załącznika

Jeśli użytkownik doda, zaktualizuje lub usunie więcej niż jeden załącznik w ramach jednej operacji, zdarzenie dotyczące załącznika wystąpi osobno dla każdego załącznika i za każdym razem wyzwoli przypisane akcje. Dla każdego załącznika wartości kontekstu zdarzenia zostaną zaktualizowane.

5.1.9. Karta Formularze

Podsumowanie:
Na karcie `Formularze` można wybrać formularz, który będzie używany do wyświetlania instancji tego dokumentu po kliknięciu przycisku edycji na liście opartej na funkcji tabelarycznej ***UserTasks_View***. Domyślnie jest to formularz tworzony poleceniem `Generuj elementy aplikacji` na podstawie modelu danych.

☛ *AdminSPA* > `Dokumenty biznesowe` > `Nowy/Edytuj` > `Formularze`



Karta Formularze

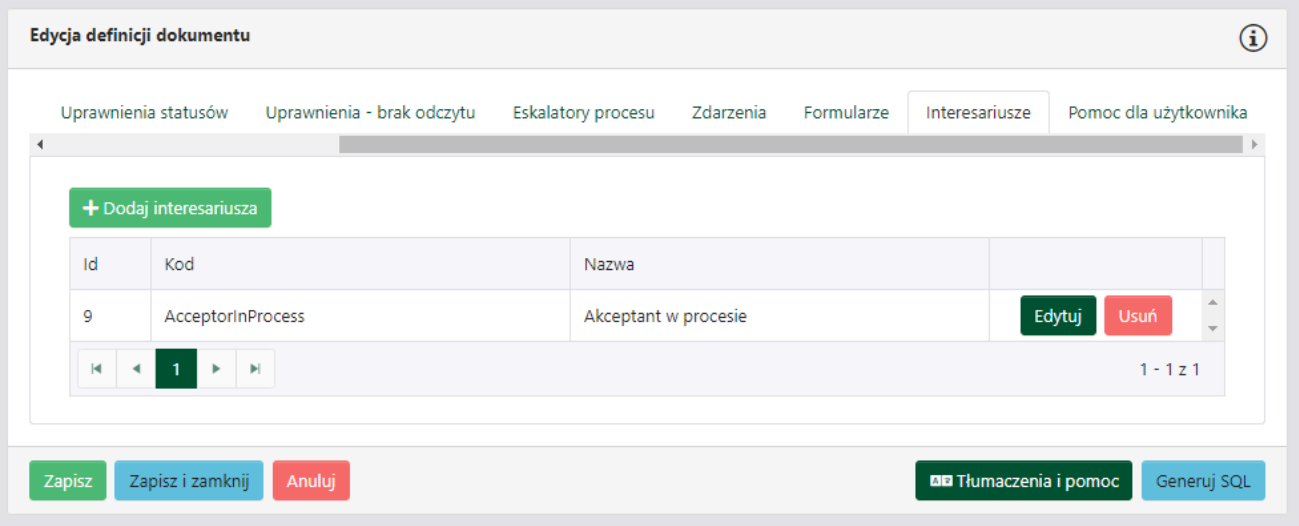
`Formularz dla listy zadań użytkownika`: wybierz z listy formularz, w którym będą wyświetlane instancje danego dokumentu z poziomu listy zadań użytkownika (systemowa funkcja tabelaryczna *UserTasks_View*).

5.1.10. Karta Interesariusze

Podsumowanie:
Interesariusze to kategoria użytkowników pozostających w określonej relacji do określonego rodzaju dokumentu. Na przykład dla wniosków urlopowych można zdefiniować kategorię interesariuszy *wnioskodawca*, dla umów — *autor opinii prawnej*, dla faktur — *akceptant* itd. Z faktu istnienia relacji użytkownik-dokument mogą wynikać określone uprawnienia, które można przyznać przez zdefiniowanie szablonu uprawnień. Kategorie interesariuszy definiuje się podczas projektowania aplikacji, natomiast dodawanie indywidualnych użytkowników do kategorii interesariuszy odbywa się dynamicznie w procesie biznesowym. Użytkownik dodany do kategorii interesariuszy nabywa uprawnienia zdefiniowane w szablonie uprawnień dla tej kategorii, użytkownik usunięty z kategorii interesariuszy traci te uprawnienia.

Na karcie *Interesariusze* można dodawać, edytować i usuwać kategorie interesariuszy dla określonej definicji dokumentu.

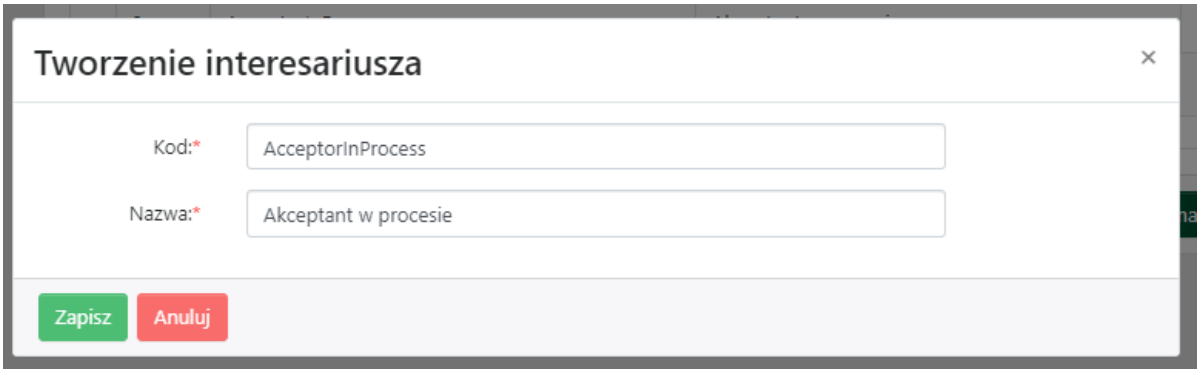
☛ *AdminSPA* > `PROCESY BIZNESOWE` > `Dokumenty biznesowe` > `Nowy/Edytuj` > `Interesariusze`



Karta Interesariusze

`Dodaj interesariusza`: wyświetl okno dialogowe, w którym można zdefiniować kategorię interesariuszy, a następnie wypełnij pola:

- `Kod`: kod kategorii interesariuszy (używany przez mechanizm *smart numbers*).
- `Nazwa`: nazwa kategorii interesariuszy.



Okno Tworzenie interesariusza

`Edytuj`: wyświetla wyświetl okno dialogowe, w którym można zmienić kod lub nazwę kategorii interesariuszy.

> **UWAGA**
> Zmiana kodu kategorii interesariuszy spowoduje konieczność zaktualizowania wszystkich odwołań do tej kategorii przy użyciu mechanizmu *smart numbers*.

`Usuń`: usuwa kategorię interesariuszy z definicji dokumentu.

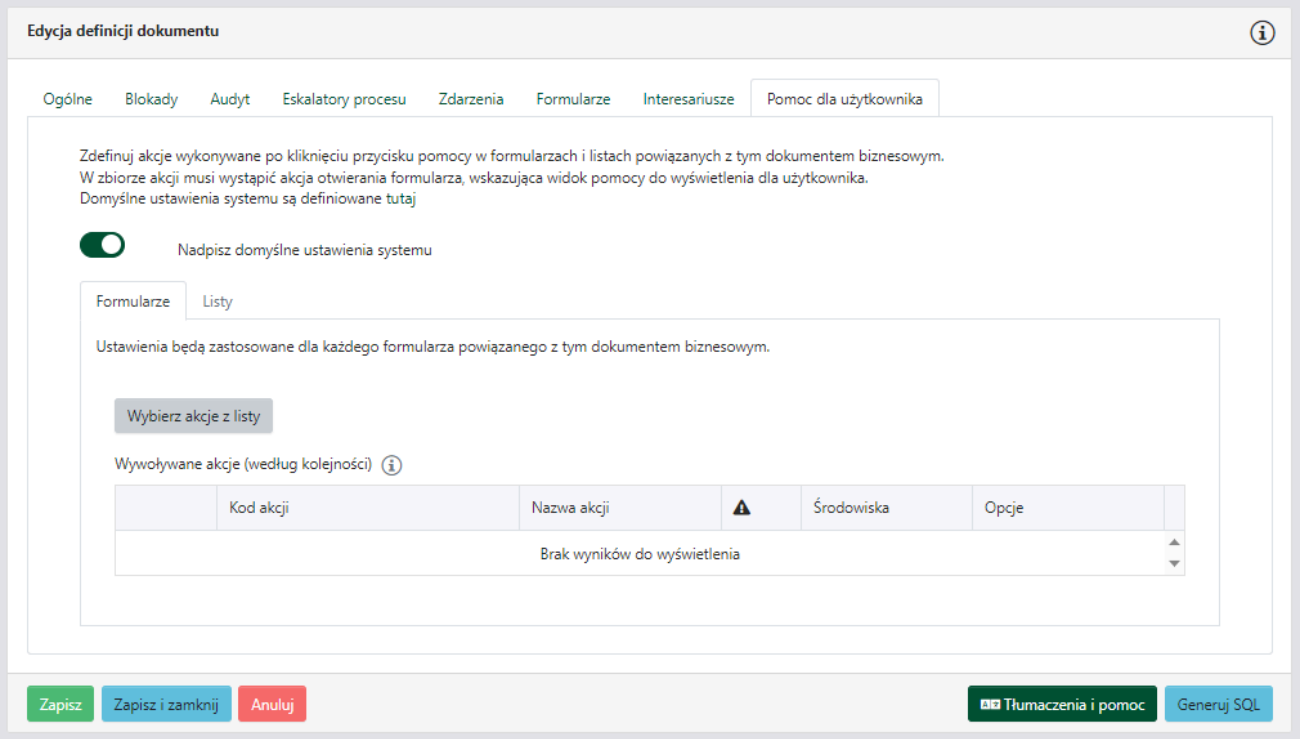
Powiązane tematy:

- Smart numbers

5.1.11. Karta Pomoc dla użytkownika

Podsumowanie:
Na karcie `Pomoc dla użytkownika` można zdefiniować indywidualną obsługę ikony pomocy dla danej definicji dokumentu biznesowego. Polega to na skonfigurowaniu sekwencji akcji, wśród których musi występować akcja otwierania formularza. Ta sekwencja będzie wykonywana po kliknięciu ikony pomocy wyświetlanej na wszystkich formularzach i listach powiązanych z bieżącą definicją dokumentu biznesowego.

☛ *AdminSPA* > `PROCESY BIZNESOWE` > `Dokumenty biznesowe` > `Nowy/Edytuj` > `Pomoc dla użytkownika`.



- `Nadpisz domyślne ustawienia systemu`: po włączeniu wyświetla komponent do wyboru akcji dla list i formularzy.
- `Formularze/Listy`: Kliknij odpowiednią zakładkę, aby skonfigurować sekwencję akcji wykonywanych po kliknięciu ikony pomocy na formularzach lub listach powiązanych z daną definicją dokumentu biznesowego. Jeśli wśród tych akcji nie będzie akcji otwierania formularza, nie będzie można zapisać zmian.
- `Dostępne akcje`: kliknij dwukrotnie wybraną akcję, aby przenieść ją do kolumny

`Wybrane akcje`; w tej kolumnie wyświetlane są tylko akcje zdefiniowane globalnie, bez akcji systemowych.

- `Wybrane akcje`: kliknij dwukrotnie wybraną akcję, aby usunąć ją z tej kolumny; możesz także przeciągać akcje myszą, aby zmienić ich kolejność

INFO
Po włączeniu przełącznika `Nadpisz ustawienia systemu` wymagane jest skonfigurowanie akcji zarówno dla formularzy, jak i dla list.

Powiązane tematy:

- Otwieranie formularza

5.2. TYPY DOKUMENTÓW

Podsumowanie:
Typ dokumentu to kombinacja dodatkowych atrybutów (maksymalnie trzech) umożliwiająca podział dokumentów tworzonych w ramach każdej definicji dokumentu na osobne kategorie. Dla każdej definicji dokumentu konsultant może określić wartości typów dokumentu w maksymalnie trzech grupach — systemowych kolumnach tabeli powiązanej z definicją dokumentu: *Type1*, *Type2* i *Type3*. Na podstawie typu dokumentu można konstruować osobne przejścia na diagramie procesów oraz określać różne schematy numeracji. Ponadto można używać typu dokumentu do filtrowania danych.

☛ *AdminSPA* > `PROCESY BIZNESOWE` > `Typy dokumentów`

Każda tabela dokumentu biznesowego tworzona w nAxiom ma predefiniowane kolumny systemowe `Type1`, `Type2` i `Type3` (typu *int*). Pozwalają one definiować *typ dokumentu* — kombinację wartości z tych trzech kolumn — i w ten sposob wydzielać podklasy dokumentów z klasy, którą opisuje definicja dokumentu biznesowego. Użytkownicy końcowi mogą ustawiać typ dokumentu na podstawie wartości zdefiniowanych przez konsultantów, korzystając z list wyboru typu dokumentu na formularzach.

W procesie generowania elementów aplikacji tworzona jest jedna wartość w kolumnie *Type1*, która następnie jest używana jako wartość domyślna. Dodatkowe wartości, o ile mają być używane, musi zdefiniować konsultant. Służy do tego polecenie `Typy dokumentów` w sekcji `PROCESY BIZNESOWE`. Po kliknięciu przycisku `Nowy typ` w wyświetlonym oknie dialogowym nalezy określić następujące właściwości:

- `Kod`: wpisz dowolny ciąg znaków identyfikujący tworzony typ dokumentu; ponieważ

kod jest używany w składni smart numbers, nie powinien zawierać znaków diakrytycznych.

- `Opis`: wpisz opis tworzonego typu dokumentu; opis jest wyświetlany na listach wyboru typu dokumentu, zarówno wbudowanych w nAxiom, jak i definiowanych jako kontrolki na formularzach.
- `Definicja dokumentu`: wybierz definicję dokumentu biznesowego, dla której ma zostać utworzony typ dokumentu.
- `Kolumna`: wybierz jedną z wartości dostępnych na liście:
  - `Type1`: utworzony typ dokumentu będzie wartością w kolumnie `Type1`.
  - `Type2`: utworzony typ dokumentu będzie wartością w kolumnie `Type2`.
  - `Type3`: utworzony typ dokumentu będzie wartością w kolumnie `Type3`.
- `Zapisz`: kliknij, aby zapisać zmiany i zamknąć okno.
- `Anuluj`: kliknij, aby zamknąć okno bez zapisywania zmian.

Usunięcie typu dokumentu jest możliwe tylko w przypadku, kiedy nie ma dokumentów, którym przypisano ten typ.

INFO

W celu użycia identyfikatora typu dokumentu w zapytaniu należy użyć składni smartnumbers, np.

> {&Reklamacje.BusinessTypes.rklRjs.personel_kod}

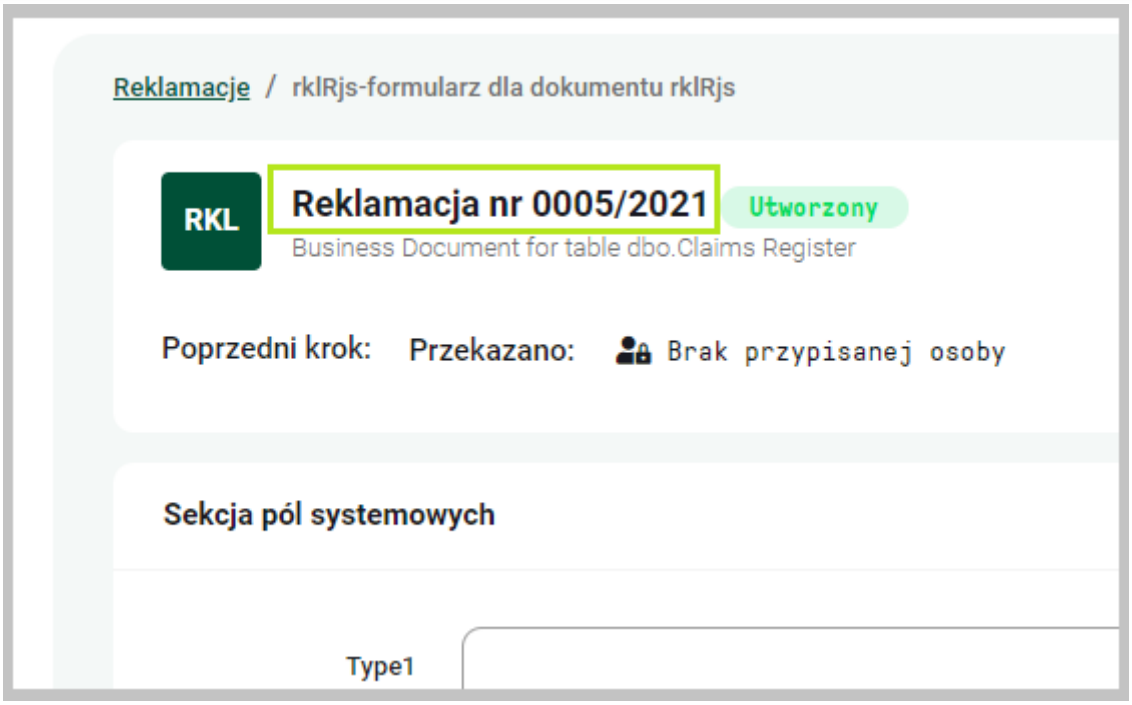
Powiązane tematy:

- Autonumeracja dokumentów
- Kreator modeli danych, generowanie elementów aplikacji
- Formularze, kontrolka Typy biznesowe

5.3. AUTONUMERACJA DOKUMENTÓW

Podsumowanie:
W obrębie każdej definicji dokumentu konsultant może zdefiniować indywidualny schemat numeracji dokumentów, na przykład *Nota nr 100/09/23/jtester*, gdzie numer, który składa się z numeru kolejnego dokumentu, miesiąca, roku i loginu użytkownika, jest dodatkowo poprzedzony tekstem *Nota nr*.
Ta funkcja jest dostępna tylko dla definicji dokumentów generowanych poleceniem `Generuj elementy aplikacji` na podstawie modelu standardowego lub uproszczonego.

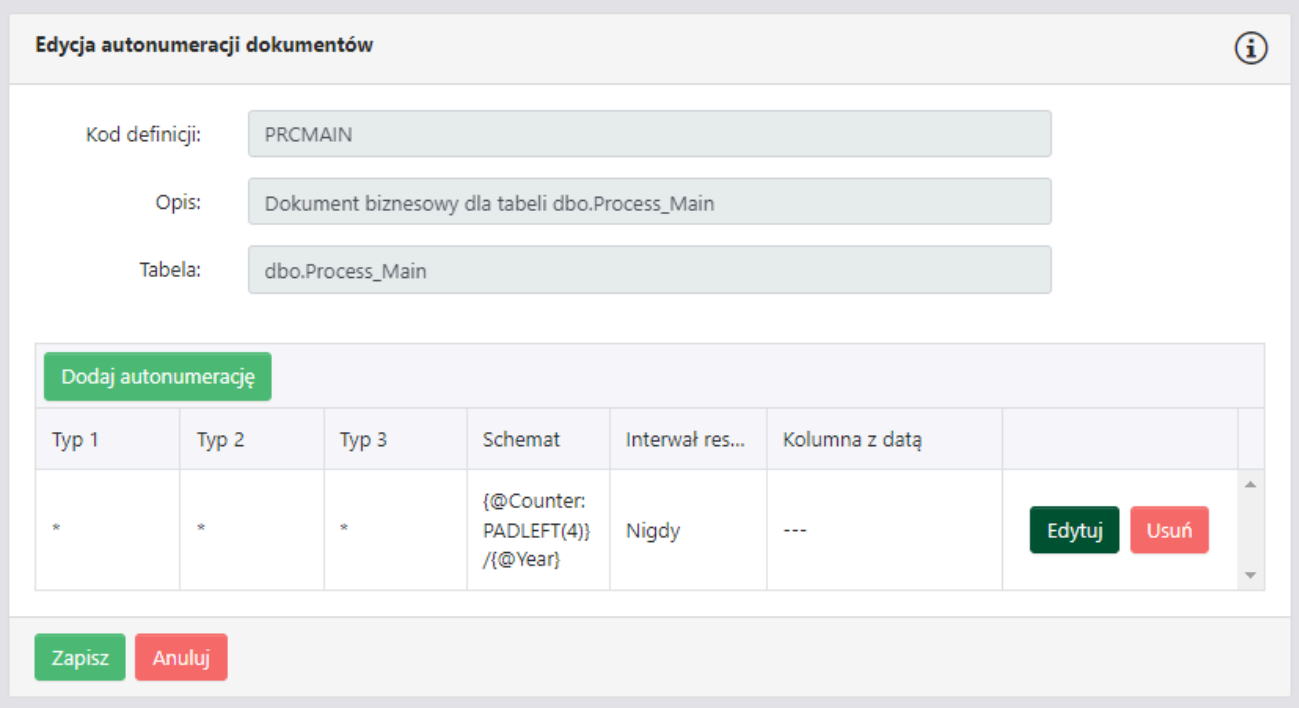
☛ *AdminSPA* > `PROCESY BIZNESOWE` > `Autonumeracja dokumentów`



Numer dokumentu w nagłówku formularza

W procesie generowania elementów aplikacji nAxiom automatycznie tworzy prosty schemat numeracji dokumentów tworzonych w ramach danej definicji dokumentu biznesowego. Ma on formę numeru kolejnego „łamanego” przez bieżący rok. Numer kolejny jest dopełniany z lewej strony zerami, tak aby ta część miała zawsze cztery cyfry, co ułatwia przeglądanie takich numerów w kolumnach. Na przykład pierwsze trzy dokumenty będą oznaczone numerami *0001/2023*, *0002/2023* i *0003/2023*. Te numery są zapisywane w kolumnie systemowej `Code` w tabeli dokumentu biznesowego. Do zarządzania schematami numeracji służy polecenie `Autonumeracja dokumentu`. Wyświetla ono listę definicji dokumentów z nazwą tabeli oraz liczbę liczników (schematów numeracji) zdefiniowanych dla każdej

definicji, a także nazwą aplikacji i modułu. Aby edytować definicję numeracji lub dodać nową definicję kliknij polecenie `Edytuj` obok żądanej definicji dokumentu.

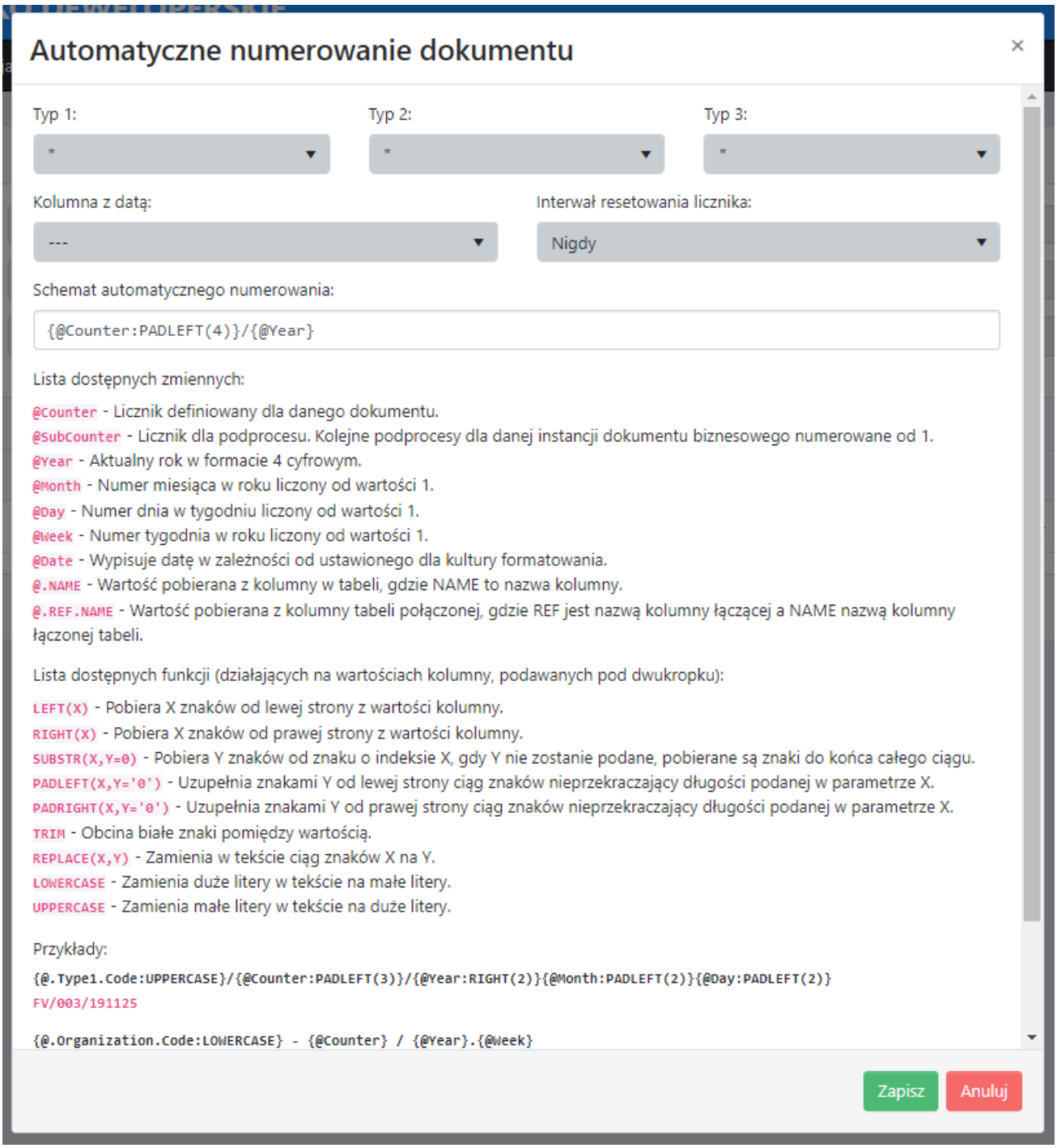


Edycja autonumeracji dokumentów

W wyświetlonym oknie można dodać nowy schemat numeracji, edytować zdefiniowany schemat lub usunąć schemat.

> **UWAGA**
> W przypadku usunięcia wszystkich schematów numeracji dla danej definicji dokumentu biznesowego, nie będzie możliwe zapisanie nowego dokumentu.

Po kliknięciu przycisku `Dodaj autonumerację` lub `Edytuj` obok zdefiniowanego schematu numeracji zostanie wyświetlone okno dialogowe z opisanymi niżej ustawieniami schematu numeracji.



Automatyczne numerowanie dokumentu

- `Type1`, `Type2`, `Type3`: wybierz z list kombinację wartości, aby utworzyć schemat numerowania dla wybranego typu dokumentu; wartość domyślna `*` oznacza dowolną wartość lub jej brak; schemat numerowania utworzony dla domyślnej wartości we wszystkich trzech polach będzie obowiązywał dla wszystkich dokumentów w ramach danej definicji dokumentu biznesowego, bez względu na ich typ, i będzie miał pierwszeństwo przed schematami zdefiniowanymi dla indywidualnych typów.
- `Kolumna z datą`: wybierz z listy nazwę kolumny typu `date/datetime`, na której wartościach będą działać funkcje daty i godziny używane w schemacie numeracji; domyślnie kolumna z datą nie jest używana w schemacie numeracji, a funkcje daty i

godziny operują na dacie bieżącej.

- `Interwał resetowania licznika`: wybierz okres, po upływie którego numeracja dokumentu będzie resetowana (pierwszy dokument po zresetowaniu będzie miał numer kolejny 1); dostępne wartości `Nigdy` (domyślna), `Co dzień`, `Co tydzień`, `Co miesiąc`, `Co rok`.

- `Schemat automatycznego numerowania`: wpisz wyrażenie, na podstawie którego będą numerowane kolejne dokumenty; patrz opis zmiennych i składni w sekcji

- `Zapisz`: kliknij, aby zapisać zmiany i zamknąć okno.
- `Anuluj`: kliknij, aby zamknąć okno bez zapisywania zmian.

UWAGA
nAxiom nie wymusza unikalności numerów generowanych według schematu. Ten obowiązek spoczywa na konsultancie.

5.4. DIAGRAMY PROCESÓW

Podsumowanie:
Diagram procesu to logiczny schemat procesu biznesowego, w którym są przetwarzane dokumenty. nAxiom tworzy zalążek diagramu procesu automatycznie w procesie generowania elementów aplikacji. Następnie taki diagram należy rozbudować w specjalnym edytorze, który opisano osobno.

☛ *AdminSPA* > `PROCESY BIZNESOWE` > `Diagramy procesów`

Modelowanie procesów w nAxiom jest oparte na statusach dokumentów. Proces to sekwencja przejść przez kolejne statusy, od pierwszego, którym jest zawsze utworzenie dokumentu, do ostatniego, który kończy proces. Oznacza to, że obsługa procesu realizowana przy użyciu aplikacji tworzonych w nAxiom odbywa się w przejściach pomiędzy statusami.

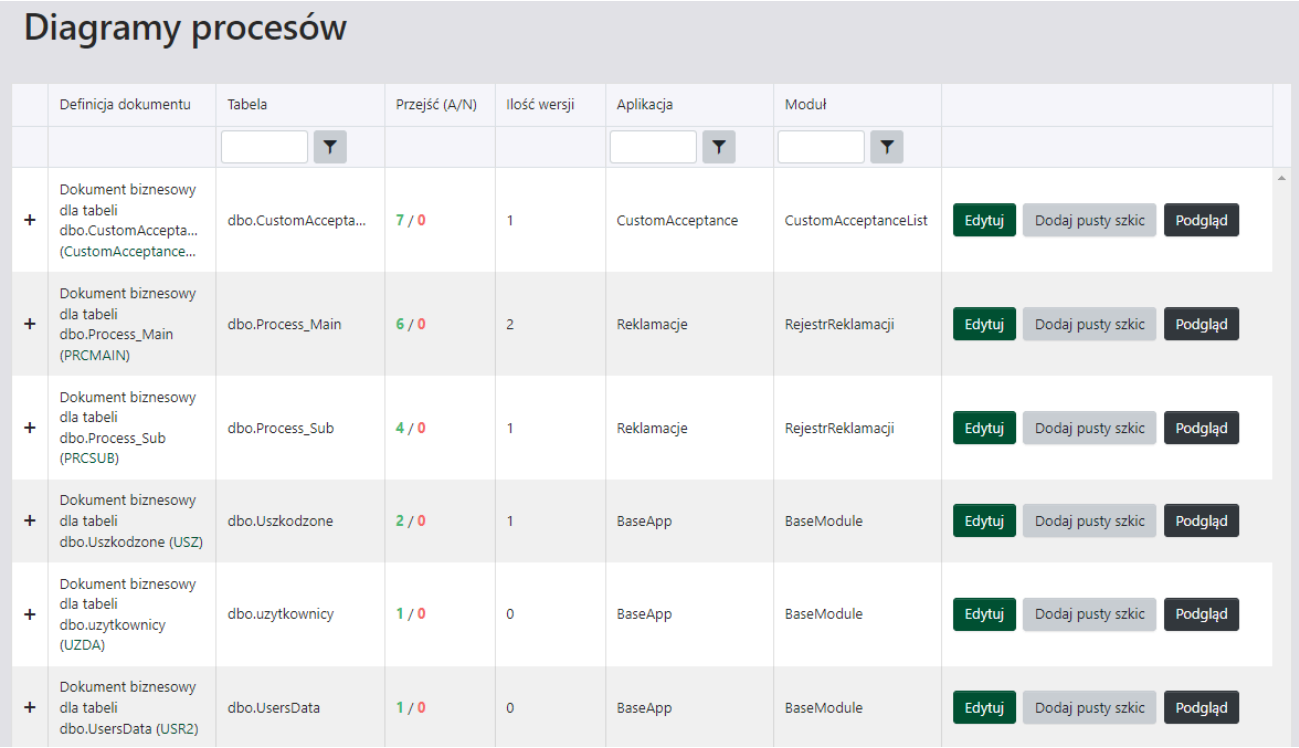
W nAxiom diagram procesu, nazywany także *workflow*, określa kolejność przechodzenia dokumentu od statusu do statusu oraz użytkowników (poprzez role), którzy mogą zmienić status dokumentu. Początkowy diagram procesu jest generowany automatycznie i można go później odpowiednio zmodyfikować. Wraz z diagramem tworzony jest cały model procesu biznesowego, na który składają się:

- definicja dokumentu biznesowego,
- deklaracja typu dokumentu biznesowego,
- schemat automatycznej numeracji,

- status dokumentu,
- diagram procesu.

Wygenerowanie modelu wymaga zdefiniowania tabeli dokumentu biznesowego, a następnie użycia polecenia `Generuj elementy aplikacji` z menu kontekstowego tej tabeli.

Aby modyfikować diagramy procesów wybierz polecenie `Diagramy procesów` z menu `Procesy biznesowe` w menu *AdminSPA*. Wyświetla ono listę diagramów procesów dla definicji dokumentów.



Diagramy procesów

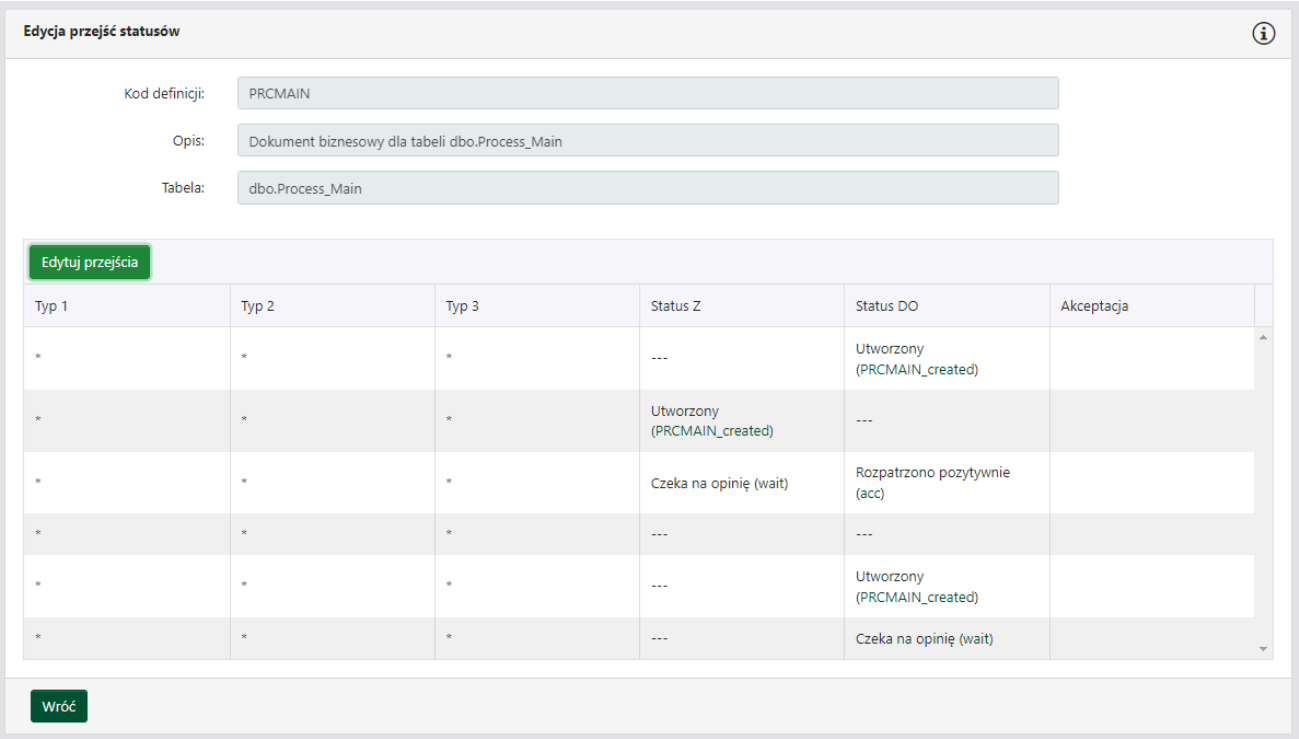
> **INFO**
> Z lewej strony każdego wiersza znajduje się symbol `+`. Jego kliknięcie powoduje rozwinięcie wiersza i wyświetlenie listy wersji diagramu.

Dla każdego diagramu w kolumnie akcji są dostępne trzy przyciski:

- `Edytuj`: kliknij, aby otworzyć daną definicję przejść statusów w wizualnym edytorze wyświetlanym w osobnej karcie.
- `Dodaj pusty szkic`: kliknij aby otworzyć edytor diagramów procesów i utworzyć w nim nowy diagram, który zostanie zapisany jako wersja głównego diagramu zdefiniowanego dla definicji dokumentu.

INFO
Aby zdefiniowana wersja diagramu była używana w aplikacji, należy ją zapisać przy użyciu przycisku `Zapisz i aktywuj` w edytorze diagramów.

- `Podgląd`: kliknij, aby wyświetlić okno `Edycja przejść statusów`, w którym przejścia statusów są wyświetlane w formie tabeli.



Edycja przejść statusów

Powiązane tematy:

- Kreator modeli danych, generowanie elementów aplikacji

5.4.1. Edytor diagramów procesów

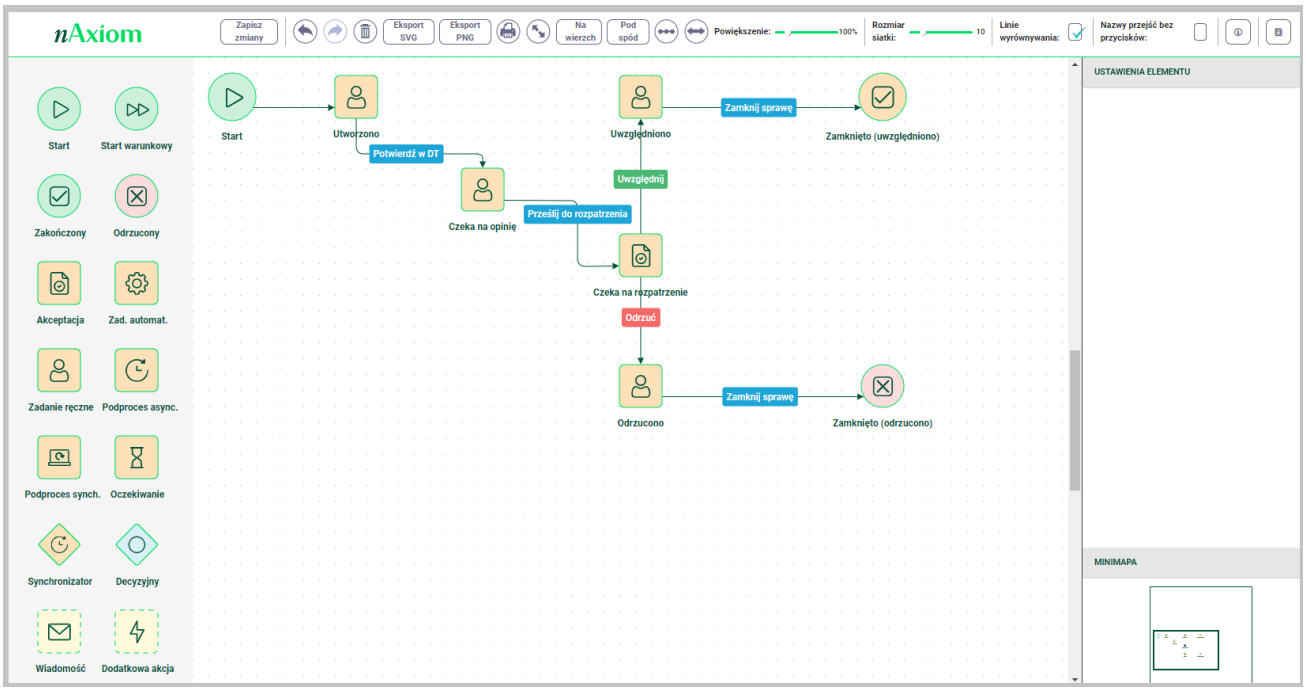
Podsumowanie:
Edytor diagramów procesów to odrębna aplikacja (*WorkflowSPA* lub *workflow*) do wizualnego projektowania diagramów procesów. Tworzenie lub modyfikowanie diagramu procesu w edytorze polega na umieszczeniu w obszarze roboczym odpowiednich bloków, połączeniu ich przejściami oraz ustawieniu właściwości bloków i przejść.

☛ *AdminSPA*: `PROCESY BIZNESOWE` > `Diagramy procesów` > `Edytuj`

W TYM TEMACIE

- Okno edytora diagramów procesów
- Praca w edytorze diagramów procesów

OKNO EDYTORA DIAGRAMÓW PROCESÓW



Edytor diagramów procesów

Okno edytora składa się z kilku głównych obszarów:

1. **Obszar roboczy** (w środku): obszar, w którym definiuje się diagram procesu
2. **Przybornik** (z lewej strony): zawiera bloki, których można używać do tworzenia diagramu procesu; aby użyć bloku, należy go przeciągnąć na obszar roboczy
3. **Ustawienia elementu** (z prawej strony u góry): panel właściwości elementu zaznaczonego w obszarze roboczym, to jest bloku lub przejścia
4. **Minimapa** (z prawej strony u dołu): ułatwia orientację w przypadku bardziej rozbudowanych diagramów.
5. **Ustawienia elementu** (z prawej strony): panel właściwości obiektu wybranego w obszarze roboczym, to jest bloku lub przejścia
6. **Pasek narzędzi** (u góry) — zawiera kontrolki do zarządzania diagramem i jego wyglądem. Są to (od lewej):

- `Zapisz zmiany`: zapisuje zmiany diagramu procesu.
- `Zapisz i aktywuj`: zapisuje diagram procesu i ustawia jako wersję aktywną dla

definicji dokumentu biznesowego; przycisk dostępny tylko po zdefiniowaniu alternatywnej wersji diagramu procesu (w menu back-office: `PROCESY BIZNESOWE` > `Diagramy procesów` > `Dodaj pusty szkic`).

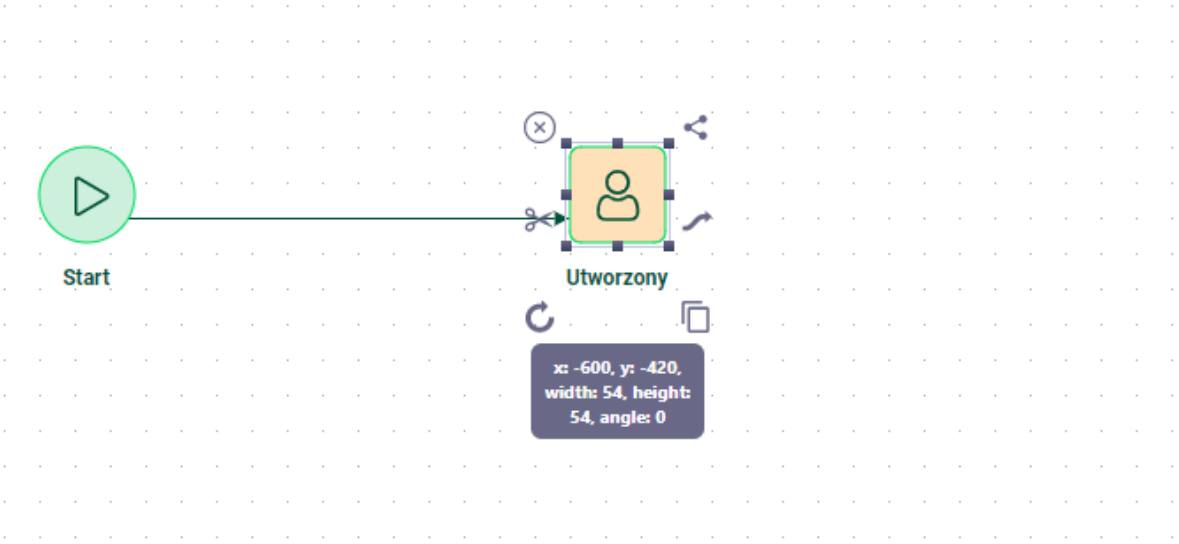
- `Cofnij`: cofa ostatnią zmianę.
- `Przywróć`: przywraca cofniętą zmianę.
- `Usuń`: czyści obszar roboczy ze wszystkich elementów.
- `Eksport SVG`: eksportuje zawartość obszaru roboczego do pliku SVG.
- `Eksport PNG`: eksportuje zawartość obszaru roboczego do pliku PNG.
- `Drukuj`: otwiera okno ustawień drukowania obszaru roboczego.
- `Pełny ekran`: włącza/wyłącza tryb pełnoekranowy.
- `Na wierzch`: przenosi dany element na pierwszy plan w przypadku kilku elementów nachodzących na siebie w obszarze roboczym w celu ułatwienia manipulowania nimi.
- `Pod spód`: przenosi dany element na ostatni plan w przypadku kilku elementów nachodzących na siebie w obszarze roboczym w celu ułatwienia manipulowania nimi.
- `Uporządkuj`: automatycznie układa elementy w obszarze roboczym.
- `Wypełnij`: automatycznie dostosowuje powiększenie obszaru roboczego.
- `Powiększenie`: umożliwia ustawienie dowolnego powiększenia obszaru roboczego.
- `Rozmiar siatki`: określa gęstość pomocniczej siatki punktów przyciągania na obszarze roboczym; bloki na obszarze roboczym są przyciągane do najbliższego punktu.
- `Linie wyrównywania`: włącza/wyłącza wyświetlanie pomocniczych linii służących do wyrównywania położenia elementów względem siebie.
- `Nazwy przejść bez przycisków`: włącza/wyłącza wyświetlanie na diagramie nazw przejść, które nie mają przycisków (np. przejście z bloku `Start` do bloku `Utworzony`)
- `Pomoc` (?): wyświetla niniejszy tekst
- `Tłumaczenia i pomoc`: wyświetla okno dialogowe `Tłumaczenia dla procesu` w którym można modyfikować teksty opisu i pomocy dla poszczególnych obiektów diagramu w różnych wersjach językowych.

PRACA W EDYTORZE DIAGRAMÓW PROCESÓW

Bloki reprezentują statusy dokumentu, dodatkowo rodzaj bloku określa właściwości przejścia, które można z takiego bloku poprowadzić. Przejścia odpowiadają operacjom wykonywanym na dokumencie. Po takiej operacji użytkownik aplikacji może na przykład

kliknąć odpowiedni przycisk, co spowoduje zmianę statusu dokumentu. W nAxiom takie „procesowe” przyciski są definiowane w edytorze workflow jako przyciski na przejściach prowadzonych między innymi z bloków typu `Zadanie ręczne`. Po zapisaniu przejść statusów i uruchomieniu aplikacji po stronie front-office te przyciski są dynamicznie wyświetlane na formularzu dokumentu odpowiednio do bieżącego statusu dokumentu. W celu umieszczenia bloku w obszarze roboczym należy go przeciągnąć myszą z przybornika. Kliknięcie bloku w obszarze roboczym powoduje wyświetlenie wokół niego kilku elementów sterujących. Są to, licząc od prawego górnego narożnika, zgodnie z kierunkiem wskazówek:

- **Klonowanie z połączeniem**: kliknij i przeciągnij, aby utworzyć kopię zaznaczonego bloku połączoną z tym blokiem.
- **Przejście**: kliknij falowaną strzałkę i przeciągnij do bloku docelowego, aby połączyć oba bloki; zwróć uwagę, aby upuścić strzałkę dopiero po wyświetleniu pomarańczowej ramki wokół obiektu docelowego, tylko wtedy oba bloki zostaną połączone przejściem.
- **Klonowanie**: kliknij, aby utworzyć kopię zaznaczonego bloku.
- **Obrót**: kliknij i przeciągnij, aby obrócić blok.
- **Usuwanie przejść**: kliknij, aby usunąć wszystkie przejścia łączące ten blok z innymi.
- **Usuwanie**: kliknij, aby usunąć blok z obszaru roboczego.

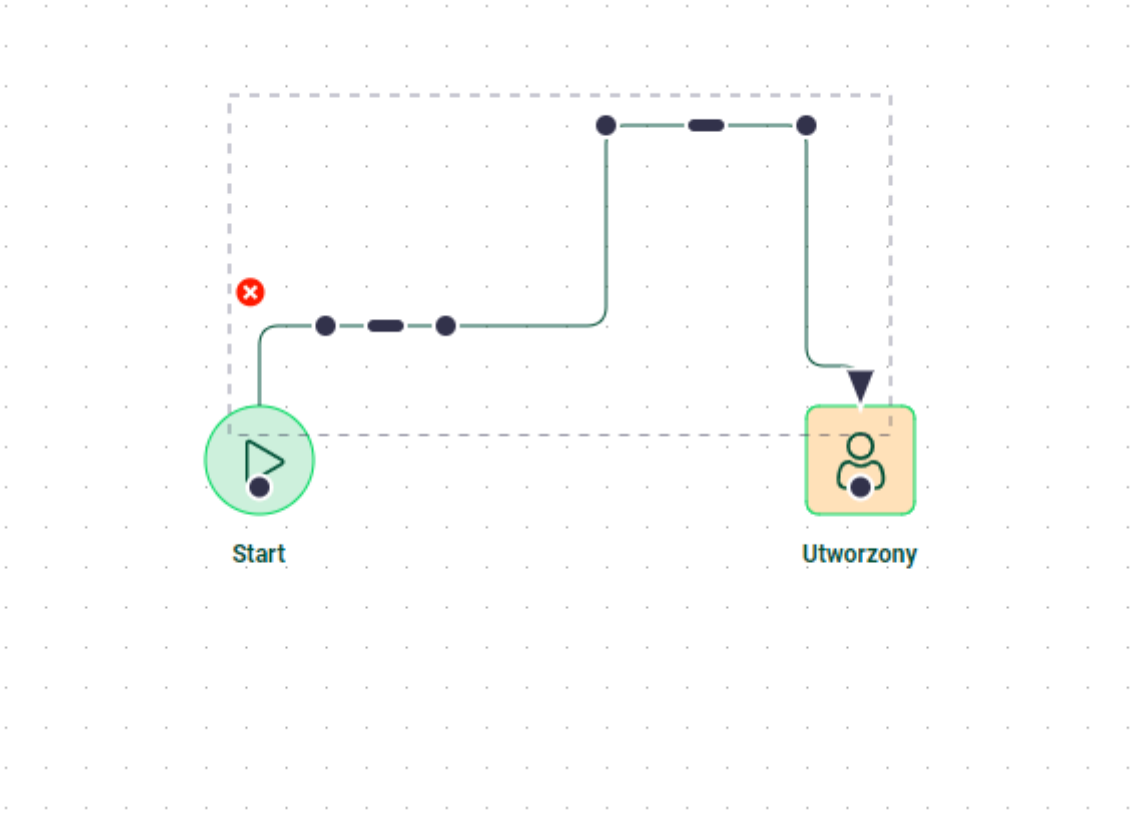


Elementy sterujące bloku

Nie wszystkie bloki mają komplet opisanych powyżej elementów sterujących. Na przykład z bloku zakończenie nie można poprowadzić przejścia do innego bloku.

W ogólności z bloków można wyprowadzać kilka przejść odpowiednio do tworzonego modelu procesu. Kliknięcie przejścia powoduje jego zaznaczenie, to jest wyświetlenie wokół niego kreskowanego obramowania. Kliknięcie czerwonego kółka z białym krzyżykiem spowoduje usunięcie przejścia. Kliknięcie zaznaczonego przejścia powoduje wyświetlenie na

nim punktów węzłowych, które można przeciągać myszą, aby nadać przejściu odpowiedni kształt i poprawić czytelność diagramu.



Elementy sterujące przejścia

Kolejnym elementem definiowania modelu procesu jest ustawianie właściwości bloków i przejść. Dla bloków podstawową właściwością będzie status, a dla przejść nazwa przycisku, którego kliknięcie spowoduje „przejście" dokumentu do kolejnego statusu, a także rola użytkownika, dla którego taki przycisk będzie widoczny. Bloki i przejścia mają także inne właściwości, które opisano w kolejnych sekcjach. W sekcji z opisem właściwości bloku opisano także właściwości przejść, które można poprowadzić z tego bloku.

5.4.2. Wspólne właściwości bloków i przejść

Podsumowanie:
Diagram procesów tworzy się, wybierając różne rodzaje bloków, łącząc je przejściami i odpowiednio ustawiając ich właściwości. W tym temacie opisano wspólne właściwości bloków i przejść.

☛ *AdminSPA*: `PROCESY BIZNESOWE` > `Diagramy procesów` > `Edytuj`

W TYM TEMACIE

- Wspólne właściwości przejść
- Wspólne właściwości bloków
- Konfigurowanie akcji na przejściach

WSPÓLNE WŁAŚCIWOŚCI PRZEJŚĆ

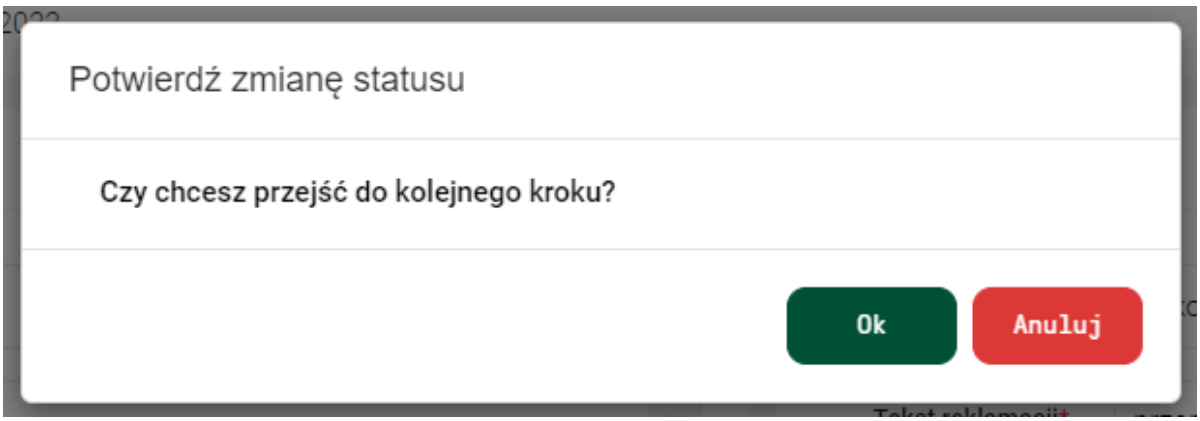
Z każdego bloku (z wyjątkiem bloku Zakończenie) można prowadzić przejścia do kolejnych bloków na diagramie procesu. Niektóre z opisanych niżej właściwości występują tylko w przypadku określonych rodzajów bloków. Przejścia na diagramie procesu w nAxiom mają następujące właściwości:

Ogólne:

- `Identyfikator przejścia`: (tylko do odczytu) globalny unikalny identyfikator przejścia, może być używany w wyrażeniach, na przykład warunkach wyświetlania przejścia.
- `Nazwa przejścia`: opisowa nazwa przejścia ułatwiająca jego identyfikację m. in. w wyrażeniach dokonujących wyboru przejścia; standardowo ma format: *blok1(status1)->(przycisk)->blok2(status2)*.
- `Lista typów`: kliknij, aby wyświetlić trzy listy rozwijane `Typ 1/2/3`; z list można wybrać wartości określające typ dokumentu, dla którego przejście będzie wyświetlane; wartość domyślna `*` — przejście jest wyświetlane dla wszystkich dokumentów.

Zmiana statusu:

- `Zmiana statusu wymaga potwierdzenia`: włącz przełącznik, aby po kliknięciu przycisku zmiany statusu wyświetlane było pytanie o dodatkowe potwierdzenie; włączenie powoduje wyświetlenie pola `Tekst potwierdzenia`, w którym można wpisać tekst pytania o potwierdzenie zmiany statusu. Kliknięcie przycisku `OK` spowoduje przejście do kolejnego statusu, a klikniecie `Anuluj` spowoduje pozostanie w bieżącym statusie.



Potwierdzenie zmiany statusu

- `Warunek wyświetlania przycisku`: określ warunek widoczności przycisku przejścia, dostępne wartości:
  - `Dla wybranych ról biznesowych`: określ role, które musi mieć użytkownik aplikacji, aby widzieć ten przycisk; z listy można wybrać więcej niż jedną wartość.
  - `Tryb zaawansowany`: zaznacz tę opcję i kliknij przycisk `SQL`, aby określić warunek wyświetlania jako zapytanie SQL. Zapytanie musi zwrócić wartość typu *bit*. Np. `SELECT CAST(1 AS BIT) AS isVisible`.
- `Warunek wykonania przejścia`: dostępne wartości:
  - `Bez warunku`: zmiana statusu reprezentowana przez przejście odbywa się bezwarunkowo.
  - `Ustaw warunek`: zaznacz opcję, kliknij przycisk `SQL` i wpisz wyrażenie SQL, którego wynik będzie decydować o tym, czy w aplikacji jest możliwa zmiana statusu dokumentu reprezentowana przez to przejście; wartość `0` spowoduje, że zmiana statusu nie będzie możliwa — podczas próby zostanie wyświetlony komunikat o błędzie. Poniższe zapytanie pozwala zmienić status w tym przejściu tylko wtedy, gdy data bieżąca jest późniejsza od wartości w polu o nazwie `DataWyjscia` na formularzu:

    ```
    SELECT CASE
    WHEN CAST(GETDATE() AS DATE) > {@DataWyjscia} THEN 1
    ELSE 0
    END
    ```

    W warunku można użyć funkcji *RAISERROR('[MSG]Treść komunikatu',...)* w celu wyświetlenia komunikatu. Jeśli poziom istotności w funkcji RAISERROR będzie większy od 11, przejście nie zostanie wykonane. W przeciwnym razie wykonanie przejście będzie zależne od wyniku zapytania.

  - `Uwzględnij także warunek wyświetlania przycisku`: (domyślnie włączony) pozwala zdecydować, czy wykonanie przejścia (łącznie z

przypisanymi akcjami przed i po) zależy tylko od warunku wykonania, czy łącznie od warunku wykonania przejścia i warunku wyświetlania przycisku.

> **WAŻNE**
> Ustawienie to jest ignorowane w przypadku, kiedy wykonanie danego przejścia jest obsługiwane przez *taskservice*. Ma to miejsce we wszystkich przypadkach, kiedy wykonanie przejścia nie jest rezultatem kliknięcia przez użytkownika przycisku nowego statusu.(np. akcja zmiana statusu lub importu dokumentów), a także wtedy, gdy proces jest wznawiany po bloczku oczekiwania. W takich sytuacjach używany jest domyślny profil użytkownika *system*, który nie ma przypisanych żadnych ról.

Przycisk na przejściu:

- `Nazwa przycisku`: wpisz, który zostanie użyty jako etykieta przycisku.
- `Kolejność wyświetlania`: wpisz liczbę całkowitą, aby określić kolejność wyświetlania danego przycisku na formularzu; wartość domyślna `1`; właściwość nieużywana.
- `Kolor czcionki i ikony`: wybierz kolor czcionki i ikony z próbnika.
- `Kolor wypełnienia`: wybierz kolor wypełnienia z próbnika.
- `Grubość ramki`: ustaw grubość obramowania (piksele).
- `Położenie ikony`: określa położenie ikony względem tekstu; dostępne wartości:
  - `Po lewej od tekstu`
  - `Po prawej od tekstu`
  - `Nad tekstem`
  - `Pod tekstem`
- `Kolor tekstu i ikony dla krycia tła przycisku 0% lub braku tła`: określa kolor tekstu i ikony wyświetlany po najechaniu kursorem myszy na przycisk z przezroczystym tłem; domyślnie jest to `Kolor akcji` ze stylu ustawionego w kreatorze stylów. Do ustawiania kolorów służy selektor, który zawiera następujące elementy:

1. Paleta odcieni, 2. Wyłączenie koloru, 3. Suwak koloru, 4. Suwak krycia, 5. Szesnastkowy kod koloru, 6. Wartości R, G i B koloru, 7. Wartość krycia, 8. Paleta kolorów standardowych.

Selektor koloru

Paleta kolorów standardowych w pierwszych trzech kolumnach zawiera: kolory kreatora stylów (główny, zależne A, B i C, akcji i zależny A), standardowe odcienie szarości oraz kolory z biblioteki *Bootstrap* (btn-success, btn-danger, btn-warning, btn-info, btn-light, btn-dark).

- `Rozmiar tekstu`: określa w procentach rozmiar tekstu względem ustawienia w kreatorze stylów (`INTERFEJS UŻYTKOWNIKA` > `Kreator stylów systemu`); wartość maksymalna to 100%.
- `Rozmiar ikony`: określa w procentach rozmiar ikony względem ustawienia w kreatorze stylów; wartość maksymalna to 100%.
- `Odstęp ikony od tekstu`: określa w pikselach odległość między ikoną i tekstem przycisku.
- `Margines wewnętrzny poziomy`: określa w pikselach *padding* w poziomie (od lewej i od prawej krawędzi przycisku).
- `Margines wewnętrzny pionowy`: określa w pikselach *padding* w pionie (od górnej i od dolnej krawędzi przycisku).

- `Kolor ramki`: wybierz kolor obramowania z próbnika.
- `Ikona przycisku`: wybierz ikonę przycisku; kliknij przycisk z ikoną lupy, aby wyświetlić okno dialogowe z ikonami do wybrania; wpisz słowo kluczowe (w języku

angielskim), aby znaleźć ikonę powiązaną z tym hasłem; wyświetlanie ikony na przycisku zależy od wartości wybranej z listy `Typ wyświetlania`.

- `Typ zachowania`: te ustawienia nie są uwzględniane dla przycisków diagramu procesów.
- `Typ wyświetlania`: określa wygląd etykiety przycisku; dostępne wartości: `Ikona i tekst` (wartość domyślna), `Ikona` i `Tekst`.

Akcje i powiadomienia:

- `Akcje uruchamiane przed`: kliknij przycisk `Ustaw`, aby wyświetlić okno dialogowe i wybrać akcje, które zostaną wykonane w momencie, kiedy użytkownik kliknie przycisk przejścia do kolejnego statusu, ale zanim jeszcze status zostanie zmieniony; nie dotyczy przejścia wychodzącego z bloku `Start`; patrz także Akcje w nAxiom .
- `Akcje uruchamiane po`: kliknij przycisk `Ustaw`, aby wyświetlić okno dialogowe i wybrać akcje, które zostaną wykonane w momencie, kiedy użytkownik kliknie przycisk przejścia do kolejnego statusu i status zostanie zmieniony; w przypadku bloku *Zadanie automatyczne* akcje przypisane w tej właściwości wykonają się automatycznie, ponieważ przejścia wychodzące z tego bloku nie mają przycisku; patrz także Akcje w nAxiom .
- `Notyfikacje akcji przejścia`: określa sposób wyświetlania powiadomień związanych z akcjami przypisanymi jako akcje uruchamiane przed i akcje uruchamiane po, domyślnie wyświetlane są wszystkie powiadomienia; dostępne wartości:
    - `Wyświetlaj pozytywne`: pozwala wyłączyć wyświetlanie powiadomień o powodzeniu wykonania akcji.
    - `Wyświetlaj negatywne`: pozwala wyłączyć wyświetlanie powiadomień o niepowodzeniu wykonania akcji.

WSPÓLNE WŁAŚCIWOŚCI BLOKÓW

Bloki diagramu procesu zarówno reprezentują statusy dokumentu, jak i autonomiczne podprocedury, w których przetwarzanie przebiega bez udziału użytkownika. Poniżej opisano typowe właściwości bloków, przy czym niektóre z nich są dostępne tylko dla określonych rodzajów bloków.

Ogólne:

- `Tekst`: wpisz tekst, który będzie wyświetlany na diagramie jako podpis bloku.
- `Status`: wybierz z listy status dokumentu, który będzie reprezentowany przez ten blok; przyciski `Edytuj` i `Dodaj` umożliwiają zmodyfikowanie statusu lub

zdefiniowanie nowego statusu; patrz także Definicje statusów .

Terminy i eskalatory:

- `Termin dla kroku procesu`: po włączeniu przełącznika można zdefiniować termin, który będzie używany przez eskalatory, czyli definiowane przez użytkownika powiadomienia o zbliżającym się/przekroczonym terminie wykonania danego kroku. Parametry określa się za pomocą zapytania SQL, które ma zwrócić interwał w formacie ISO. W przebiegu procesu te wartości są przeliczane na wartości *datetime* i zapisywane w systemowej tabeli bazy danych *KpiExecutionTimes* wraz z wartościami granicznymi, to jest momentami zmiany statusu na wejściu i wyjściu tego kroku. Przycisk `Info` wyświetla okienko z informacjami dotyczącymi formatu interwału ISO.

Zasady notacji ISO-8601

- `Algorytm obliczania terminu dla kroku procesu`: sposób obliczania daty terminu na podstawie określonego interwału z uwzględnieniem kalendarza dni wolnych od pracy (tabela *[core].[NonWorkingDays]*); dostępne opcje:
  - `Dokładnie w dacie terminu kalendarzowego`: data terminu jest obliczana przez dodanie interwału do daty wykonania poprzedniego przejścia bez uwzględniania dni wolnych od pracy.
  - `Licz tylko dni robocze`: data terminu jest obliczana z pominięciem dni wolnych przypadających na interwał terminu.

- `Oblicz ostatni dzień roboczy przed terminem kalendarzowym`: jeśli data terminu przypada na dzień wolny, przesuwa ją na poprzedni dzień roboczy.
- `Oblicz pierwszy dzień roboczy po terminie kalendarzowym`: jeśli data terminu przypada na dzień wolny, przesuwa ją na następny dzień roboczy.

`Aktywuj eskalatory procesu`: po włączeniu pozwala skonfigurować powiadomienia o zbliżającym się/przekroczonym terminie zdefiniowanym dla kroku procesu. Kliknij przycisk `konfiguracja eskalatorów` i skonfiguruj eskalator:

`Eskalator 1`: wpisz zapytanie SQL, które zwróci termin aktywowania powiadomienia względem terminu dla kroku procesu, użyj znaku -/+, aby wskazać, czy eskalator ma zostać aktywowany przed czy po upływie terminu. Np. `SELECT '-P1D'` aktywuje eskalator na dzień przed upływem terminu. Brak znaku jest traktowany jak +.

`Algorytm obliczania terminu wysyłki eskalatora`: sposób obliczania daty terminu wysyłki na podstawie określonego interwału z uwzględnieniem kalendarza dni wolnych od pracy (tabela *[core].[NonWorkingDays]*); dostępne opcje są takie same jak w przypadku algorytmu obliczania terminu dla kroku procesu.

`Powiadamiaj przez`: wybierz rodzaj powiadomienia dla eskalatora, dostępne wartości: `wiadomość email` i `powiadomienia systemowe`.

`Adresaci`: wybierz osoby, które otrzymają powiadomienie, dostępne wartości: `właściciel zadania`, `przełożony właściciela zadania`, przycisk `Ustaw` wyświetla okno dialogowe, w którym można skonfigurować adresatów wiadomości e-mail.

`Adresaci Do`, `Adresaci DW`, `Adresaci UDW`: karty na których można zdefiniować adresatów wiadomości. Każda karta ma pięć kart podrzędnych, na których można definiować adresatów odrębnymi metodami.

`Interesariusze`: jako adresaci zostaną dodani użytkownicy przypisani do wybranych grup interesariuszy dla instancji dokumentu zwróconych przez zapytanie:

- `Dokument biznesowy`: definicja dokumentu, dla której zdefiniowano grupy interesariuszy.
- `Instancje`: zapytanie SQL zwracające identyfikatory instancji dokumentu (wymagane jest pełne zapytanie ze wskazaniem tabeli w klauzuli *FROM*); jako adresaci zostaną wybrani użytkownicy przypisani do wybranych grup interesariuszy dla tych instancji.
- Lista grup interesariuszy dla wybranej definicji dokumentu; z listy należy wybrać grupy interesariuszy, z których będą wybierani adresaci.

`Posiadacze ról globalnych`: jako adresaci zostaną dodani użytkownicy, którym przypisano role wybrane z listy ról dostępnej na karcie.

`Użytkownicy wskazani przez zapytanie SQL`: jako adresaci zostaną dodani użytkownicy, których identyfikatory zostaną zwrócone przez zapytanie SQL.

`Adresy e-mail wskazane przez zapytanie SQL`: jako adresaci zostaną dodani właściciele adresów e-mail zwróconych zapytaniem SQL; zapytanie może zwrócić wyniki w formie kolumny lub ciągu wartości rozdzielonych znakiem średnika.

`Adresy e-mail wskazane bezpośrednio`: jako adresaci zostaną dodani właściciele adresów e-mail wpisanych w kontrolce wielowyboru (obsługiwana jest notacja smart numbers).

Na każdej z tych kart znajduje się przełącznik `Włączone`, którym można dezaktywować konfigurację adresatów bez konieczności usunięcia jej. Zakładki z aktywną konfiguracją posiadają pogrubiony tekst i systemowy ciemno-zielony kolor.

`Dodaj eskalator`: pozwala zdefiniować kolejny eskalator dla danego kroku procesu.

KONFIGUROWANIE AKCJI NA PRZEJŚCIACH

Na przejściach można konfigurować zbiory akcji wywoływanych przed zmianą statusu dokumentu i po niej. Domyślnie nAxiom dodaje akcję zapisu rekordu do akcji przed oraz akcję odświeżenia formularza do akcji po. Odpowiadają za to ustawienia przełączników na karcie `Ogólne` w oknie `Edycja definicji dokumentu`.

Akcje związane z obsługą graficznego interfejsu użytkownika (odświeżanie formularza lub listy, JavaScript) nie są wykonywane, jeśli zostaną umieszczone na przejściu między blokami nie posiadającymi statusu.

Jeśli między dwoma blokami z przypisanym statusem znajduje się np. kilka bramek decyzyjnych, zaleca się skonfigurowanie akcji zapisu rekordu na początku zbioru akcji przed dla przejścia początkowego oraz akcji odświeżania formularza lub listy na końcu zbioru akcji po dla przejścia zamykającego sekwencję.

W przypadku używania akcji zapisu rekordu wewnątrz takiej sekwencji, należy na początku zestawu akcji w przejściu końcowym użyć akcji odświeżenia modelu, aby uniknąć komunikatu o błędzie informującego o edycji rekordu przez innego użytkownika.

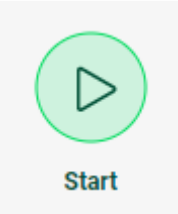
> **WAŻNE**
> Po zdefiniowaniu eskalatorów wymagany jest restart witryny i serwisu Taskservice (puli aplikacji).

Powiązane tematy:

- Odświeżanie formularza podczas zapisu rekordu

5.4.3. Blok Start

Podsumowanie:
Blok `Start` reprezentuje rozpoczęcie procesu, zwykle będzie to utworzenie nowego dokumentu, ale jeszcze przed zapisaniem go. Z bloku start można poprowadzić tylko jedno przejście. To przejście jest wykonywane w momencie zapisania nowego dokumentu.

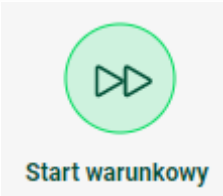


Powiązane tematy:

- Wspólne właściwości bloków i przejść

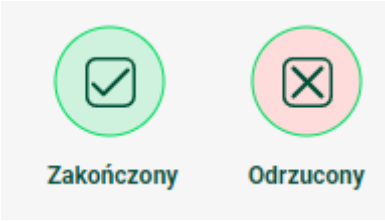
5.4.4. Start warunkowy

Podsumowanie:
Od wersji 1.11.2.0 blok Start warunkowy zostaje uznany za nieobsługiwany. Planowane jest usunięcie tego bloku z edytora diagramów procesów.



5.4.5. Zakończenie

Podsumowanie:
Bloki zakończenia dostępne w nAxiom różnią się tylko wyglądem i pozwalają zilustrować pomyślny lub niepomyślny status kończący procesu. Ponieważ te bloki wskazują na zakończenie procesu, nie można z nich poprowadzić przejść do innych bloków.



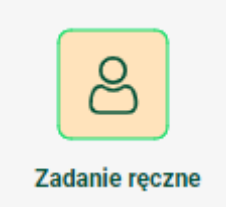
Powiązane tematy:

- Wspólne właściwości bloków i przejść

5.4.6. Zadanie ręczne

Podsumowanie:
Blok *Zadanie ręczne* jest używany w edytorze workflow do modelowania etapu procesu, na którym zmiana statusu dokumentu będzie wynikiem kliknięcia przez użytkownika przycisku zdefiniowanego na przejściu wychodzącym z tego bloku.

Z tego bloku można poprowadzić więcej niż jedno przejście.

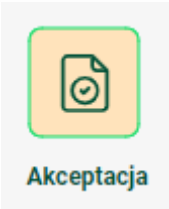


Powiązane tematy:

- Wspólne właściwości bloków i przejść

5.4.7. Akceptacja

Podsumowanie:
Blok `Akceptacja` reprezentuje krok w procesie, w którym co najmniej jeden użytkownik z wyznaczoną rolą musi podjąć decyzję, mając do wyboru dwie możliwości: zaakceptować dokument lub go odrzucić.



Oprócz właściwości opisanych osobno, dla bloku akceptacji i wychodzących z niego przejść konieczne jest określenie specyficznych właściwości.

BLOK:

- `Akceptacja przez przełożonego`: włącz, aby automatycznie wyznaczyć do akceptacji przełożonego bieżącego użytkownika, to jest użytkownika wybranego w polu `Osoba nadrzędna` podczas tworzenia profilu użytkownika; patrz także Tworzenie i edycja profilu użytkownika ; włączenie tej opcji spowoduje ukrycie pól pozostałych właściwości.
- `Akceptacja przez role`: wybierz jedną lub kilka ról — tylko użytkownicy, którym zostały przypisane te role będą mieć w aplikacji dostęp do dokumentów przedłożonych do akceptacji; jeśli wybierzesz kilka ról, akceptować będą mogli użytkownicy, którzy mają przypisane wszystkie te role.
- `Ilość wymaganych akceptacji`: (domyślnie *1*) określ, ile osób musi zaakceptować dokument, aby jego status zmienił się; akceptacja oznacza kliknięcie przycisku na przejściu, dla którego ustawiono właściwość `Rezultat przejścia` = `Pozytywny`.
- `Filtrowanie użytkowników w rolach`: wpisz wyrażenie SQL, które pozwoli zawęzić grupę użytkowników z przypisanymi rolami wybranymi w polu `Akceptacja przez role`; wyrażenie musi zwracać identyfikatory użytkowników.

> **INFO**
> Kiedy dokument znajdzie się w statusie oczekiwania na akceptację, wszyscy użytkownicy, którzy są uprawnieni do zaakceptowania/odrzucenia dokumentu otrzymają powiadomienie systemowe (ikona dzwonka w nagłówku witryny w aplikacji *FrontSPA*).

PRZEJŚCIE:

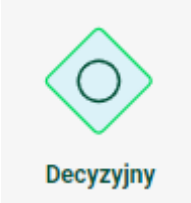
- `Rezultat akceptacji`: wybierz rodzaj akceptacji dla przejścia; dostępne wartości: `Pozytywny`, `Negatywny`, `Brak`. Tym wartościom odpowiadają następujące wartości zapisywane w kolumnie *[AcceptanceResult]* tabeli *[core].[BusinessTransitionAcceptances]*:
  - `Pozytywny`: *1*
  - `Negatywny`: *2*
  - `Brak`: *0*

Powiązane tematy:

- Wspólne właściwości bloków i przejść

5.4.8. Blok decyzyjny

Podsumowanie:
Blok decyzyjny pozwala uzależnić przebieg procesu od warunku zdefiniowanego w zapytaniu SQL określonym w właściwościach bloku. Zapytanie musi zwrócić nazwę jednego z przejść wychodzących z bloku. Alternatywnie, do wyboru przejścia można użyć akcji Web Service. Blok decyzyjny nie ma przypisanego żadnego statusu. Kiedy dokument uzyska status bloku umieszczonego bezpośrednio przed blokiem decyzyjnym zostanie wykonane zapytanie SQL, które zwróci nazwę przejścia, odpowiednio do wyniku warunku. W ten sposób można parametryzować przebieg procesu. Na przykład w przypadku faktur można skonstruować warunek, który spowoduje, że faktury z kwotą powyżej pewnego limitu będą kierowane do akceptacji przez osoby z przypisaną rolą kierowniczą, podczas gdy dla faktur z kwotą niższą ten krok będzie pomijany.



Oprócz właściwości opisanych osobno, dla bloku decyzyjnego konieczne jest określenie specyficznych właściwości.

- `Akcja Web Service`: włącz ten przełącznik, aby nazwa przejścia była wybierana przez zapytanie SQL zdefiniowane w akcji Web Service wybranej z listy rozwijanej; w tym przypadku wybór przejścia można uzależnić od danych zwróconych z systemu zewnętrznego; patrz także Akcja zapytań do web service .
- `Kierunek przejścia`: w tym polu wpisz kod zapytania SQL, które zwróci nazwę przejścia w zależności od wyniku wyrażenia warunkowego użytego w zapytaniu; kliknij przycisk `Konstruktor zapytania`, aby skonstruować zapytanie przy użyciu konstruktora (wynikowy kod SQL zostanie wyświetlony w polu tekstowym, gdzie można go modyfikować); patrz Okno Konstruktor SQL - definiowanie warunków przejść bloku decyzyjnego .

PRZYKŁADY:

W przypadku, gdy wartość pola *Kwota* w dokumencie biznesowym przekracza *300*, dokument zostanie skierowany do akceptacji, w przeciwnym razie zostanie zatwierdzony automatycznie.

```
SELECT
  CASE
    WHEN ({@Kwota} > 300)
    THEN 'do kontroli'
    ELSE 'zatwierdź automatycznie'
  END
```

W przypadku, gdy po przemnożeniu wartości z pola *Kwota* przez kurs średni waluty (*mid*) zwrócony przez akcję Web Service wynik będzie większy od *300*, dokument zostanie skierowany do akceptacji, w przeciwnym razie zostanie zatwierdzony automatycznie.

```
SELECT
 CASE
   WHEN (
     SELECT [mid]*{@Kwota} from openjson({@_Response}, '$.rates')
        WITH ([no] varchar(20), [effectiveDate] varchar(20), [mid] decimal(18,4))
        ) > 300
   THEN 'do kontroli'
   ELSE 'zatwierdź automatycznie'
 END
```

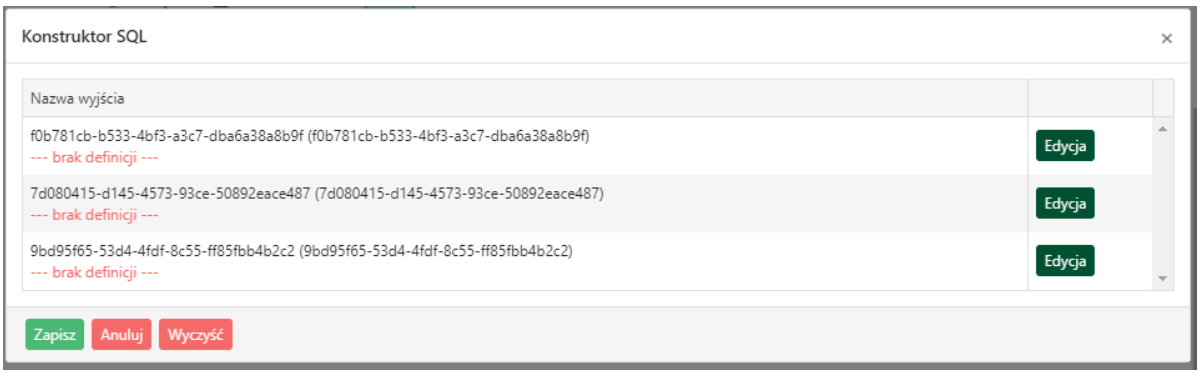
Literały *do kontroli* i *zatwierdź automatycznie* to wartości właściwości `Nazwa przejścia` określone dla przejść wychodzących z bloku decyzyjnego.

Powiązane tematy:

- Wspólne właściwości bloków i przejść

5.4.8.1. OKNO KONSTRUKTOR SQL - DEFINIOWANIE WARUNKÓW PRZEJŚĆ BLOKU DECYZYJNEGO

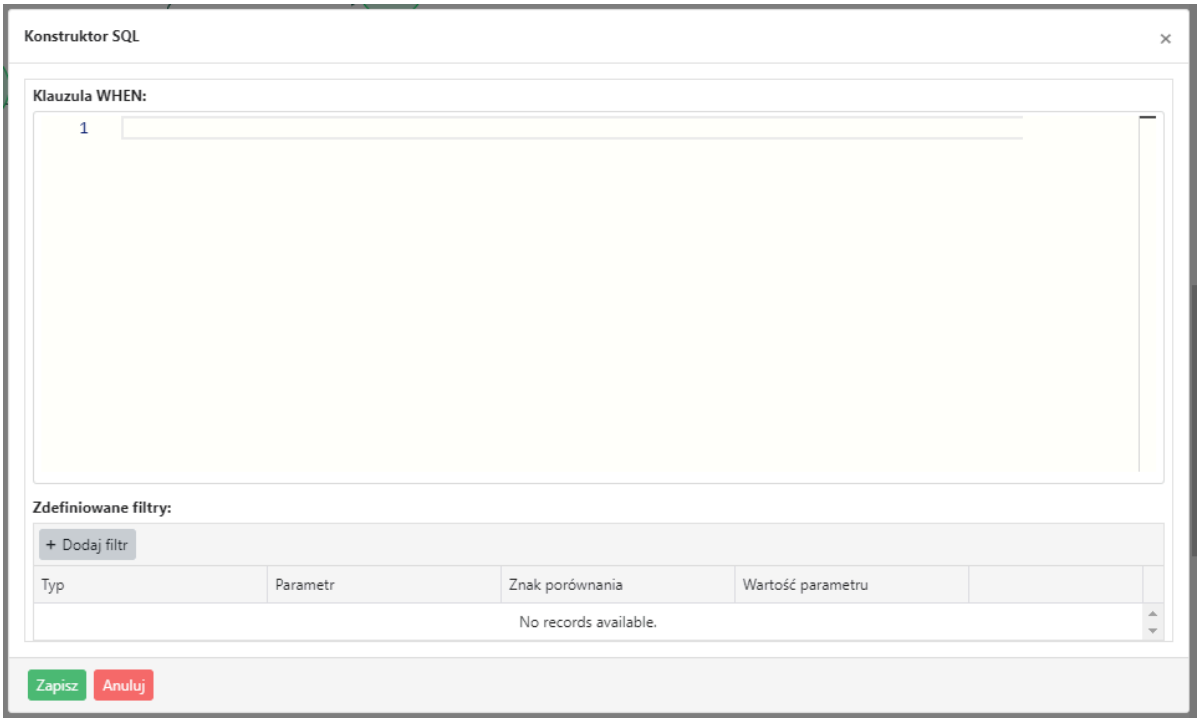
1. Kliknij przycisk `Konstruktor zapytania` we właściwościach bloku decyzyjnego. Zostanie wyświetlone okno `Konstruktor SQL`.



Okno `Konstruktor SQL`

> **WSKAZÓWKA**
> na ilustracji przedstawiono okno dla bloku, z którego poprowadzono trzy przejścia; dla każdego z nich należy skonstruować warunek wyboru.

2. Kliknij przycisk `Edycja` w wierszu przejścia.
   Zostanie wyświetlone okno konstruowania warunków (filtrów).

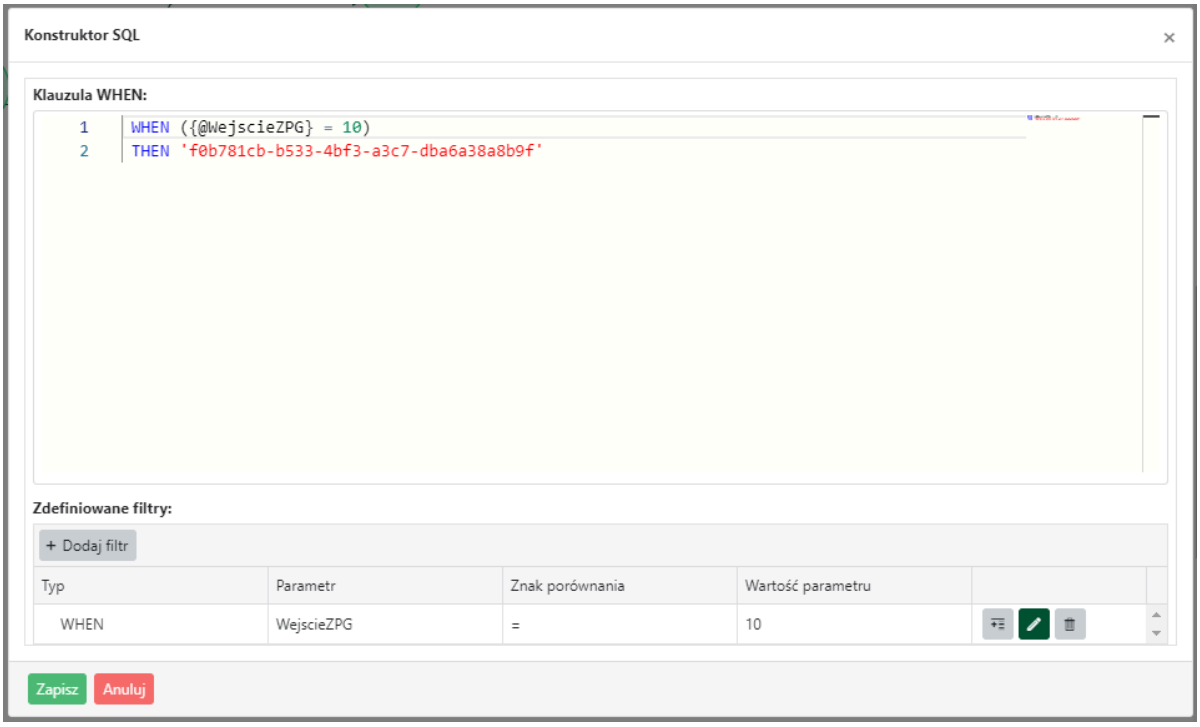


Konstruowanie warunku SQL

3. Kliknij przycisk `Dodaj filtr`. Zostanie rozwinięty wiersz definiowania warunku.
4. Wybierz kolejno:
   - `Parametr`: wybierz nazwę kolumny tabeli dokumentu biznesowego, z której wartość ma zostać użyta w warunku; dostępne są kolumny zdefiniowane przez użytkownika i kolumny systemowe.
   - `Znak porównania`: wybierz z listy operator warunku.
   - `Wartość parametru`: wpisz wartość, która zostanie porównana w warunku

z wartością z kolumny wybranej na liście `Parametr`.

5. Kliknij ikonę dyskietki, aby zapisać zdefiniowany warunek.
   Warunek zostanie wyświetlony powyżej w obszarze `Klauzula WHEN`.



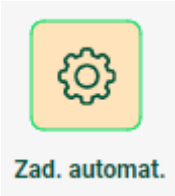
Warunek WHEN

6. Ponownie kliknij przycisk `Dodaj przycisk`, aby dodać nowy warunek; typ razem z listy `Typ` można wybrać operator `AND` lub `OR`, aby połączyć oba warunki w koniunkcję lub alternatywę logiczną; ten sam efekt można uzyskać, klikając ikonę rozwijania (`+`)w wierszu warunku.

> **WSKAZÓWKA**
> Aby zmodyfikować lub usunąć warunek, kliknij odpowiednio ikonę z ołówkiem lub ikonę kosza na śmieci w wierszu tego warunku.

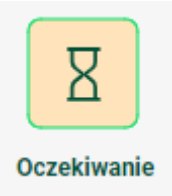
5.4.9. Zadanie automatyczne

> **Podsumowanie:**
> Blok nie jest obsługiwany.



5.4.10. Oczekiwanie

Podsumowanie:
Blok `Oczekiwanie` pozwala wstrzymać przebieg procesu na określony czas i wznowić go po upływie ustawionego czasu. Blok oczekiwania ma status, który pozwala informować użytkownika o bieżącym miejscu przetwarzania. Przejście do statusu tego bloku zamyka transakcję otwartą w chwili kliknięcia przez użytkownika przycisku zmiany statusu. Po upływie czasu oczekiwania aplikacja *taskservice* otwiera kolejną transakcję. Tę właściwość, w połączeniu z możliwością ustawienia czasu oczekiwania 0, można wykorzystać, aby pominąć blokadę okna przeglądarki z animacją operacji w toku stosowaną w przypadku długotrwałych operacji. Ten blok ma domyślnie przypisaną akcję *Odśwież formularz*, która jest wykonywana po przejściu z tego bloku do następnego.



Oprócz właściwości opisanych osobno, dla bloku oczekiwania konieczne jest określenie specyficznej właściwości.

- `Czas oczekiwania (w sekundach)`: ustaw liczbę sekund, po upływie których proces przejdzie do następnego bloku. Ta właściwość wymaga ustawienia parametru `Interwał powtarzania uruchamiania oczekujących procesów (sek)` w sekcji `Procesy biznesowe` w ustawieniach systemu. Po zmianie tego ustawienia konieczne jest restartowanie aplikacji *Taskservice*.

Dzięki swoim właściwościom blok oczekiwania jest przydatny do obsługi bloku podprocesu asynchronicznego. W przypadku kiedy zainicjowanie takiego podprocesu trwa zauważalnie długo (np. podprocesy są wywoływane w pętli), użytkownik musiałby czekać aż zniknie animacja operacji w toku, która blokuje okno przeglądarki. Zastosowanie na wejściu do bloku podprocesu bloku oczekiwania 0 s pozwala ominąć tę blokadę, ponieważ nie jest ona stosowana do transakcji otwieranych przez aplikację *taskservice*, a tak się dzieje, kiedy nAxiom wznawia przetwarzanie po upływie czasu oczekiwania. W takim przypadku można dodatkowo ustawić czas oczekiwania równy 0, aby nie wprowadzań niepotrzebnych opóźnień.

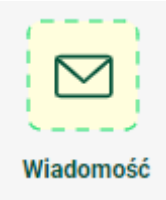
Ponadto użycie bloku oczekiwania 0 s jest konieczne w przypadku podprocesów, które wykonują się automatycznie, np. składają się tylko z bloku *Start* i *Utworzony*. Bez tego bloku wywołanie takiego podprocesu, niezależnie od trybu, zakończy się niepowodzeniem.

Powiązane tematy:

- Wspólne właściwości bloków i przejść
- Ustawienia systemu, sekcja Procesy biznesowe

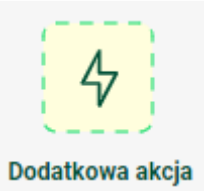
5.4.11. Wiadomość

Podsumowanie:
Blok nie jest obsługiwany.



5.4.12. Dodatkowa akcja

Podsumowanie:
Blok nie jest obsługiwany.



5.4.13. Podprocesy

Podsumowanie:
W przebiegu procesów biznesowych często spotyka się sytuacje, kiedy jeden proces inicjuje odrębny proces. Proces inicjujący nazywa się *procesem głównym*, a proces inicjowany *podprocesem*. W nAxiom podprocesy mogą być wykonywane synchronicznie — proces główny czeka na zakończenie podprocesu, lub asynchroniczne — podproces jest inicjowany z procesu głównego, ale jego dalsze wykonywanie nie wstrzymuje biegu procesu głównego.

W TYM TEMACIE

- Wstęp
- Przykład konfiguracji podprocesu
- Dobre praktyki projektowe

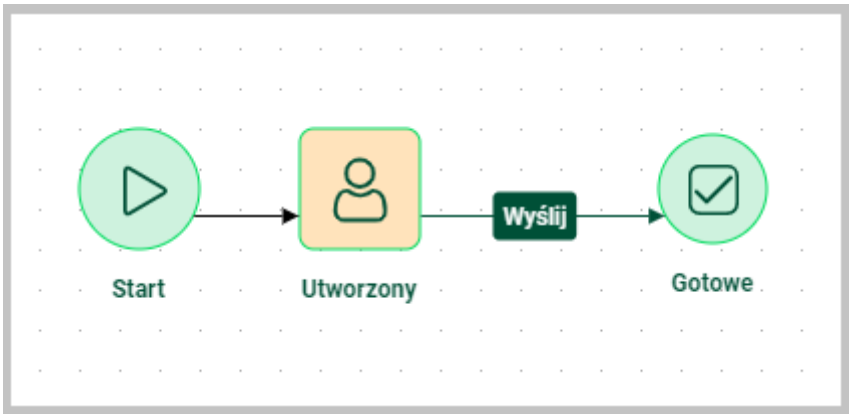
WSTĘP

Analizując procesy biznesowe toczące się w różnych organizacjach, łatwo zauważyć, że często „przecinają się” one z innymi procesami. Na przykład proces realizacji umowy wymaga co pewien czas uruchomienia procesu wystawienia faktury, z kolei proces wystawienia faktury powinien zainicjować proces monitorowania płatności. Podstawą wystawienia faktury w ramach umowy jest przekazanie przedmiotu umowy, co zwykle wymaga kilku procesów akceptacji.

W nAxiom do modelowania podprocesów służą bloki `Podproces synchroniczny` i `Podproces asynchroniczny` dostępne w edytorze diagramów procesów. Ich właściwości ustawia się na panelu z lewej strony, podobnie jak w przypadku innych bloków. Przede wszystkim konieczne jest przypisanie dokumentu biznesowego, którego workflow będzie reprezentowany przez blok podprocesu. Zatem użycie bloku podprocesu wymaga wcześniejszego przygotowania osobnego dokumentu biznesowego z całym workflow.

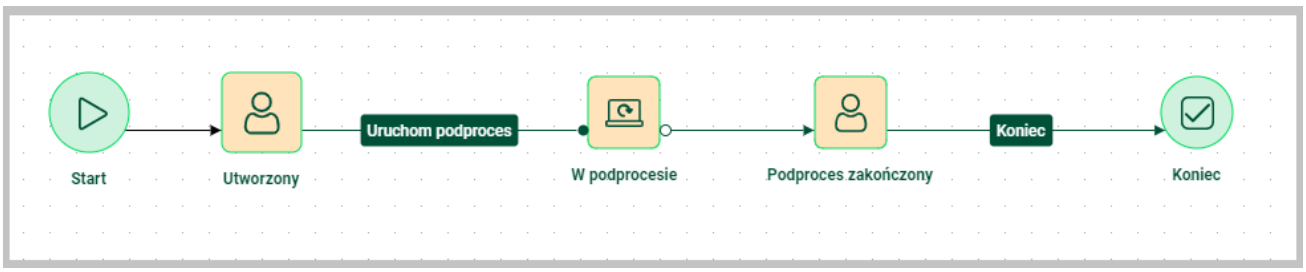
PRZYKŁAD KONFIGURACJI PODPROCESU

Na ilustracji poniżej przedstawiono uproszczony model procesu, który będzie inicjowany jako podproces z procesu głównego. W procesie jest tworzony nowy dokument biznesowy, a przetwarzanie sprowadza się do uzupełnienia pola tekstowego i kliknięcia przycisku `Wyślij`.



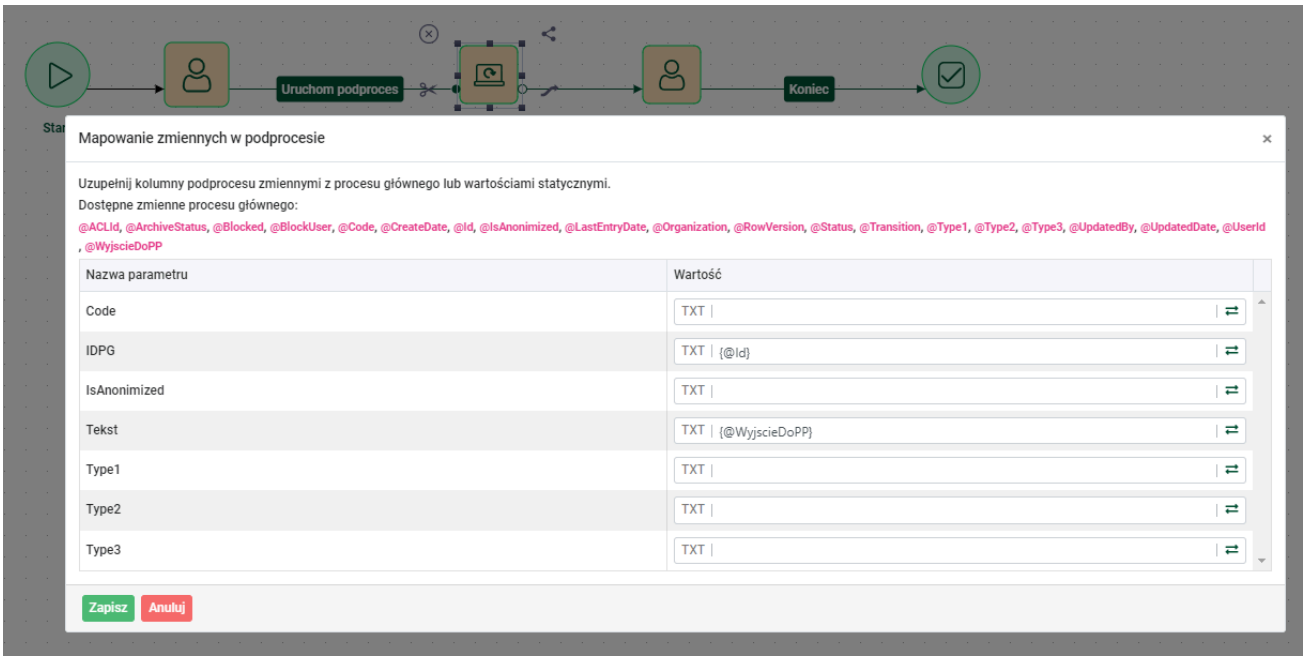
Model procesu, który będzie inicjowany jako podproces

Przykładowy proces główny jest również maksymalnie uproszczony: z poziomu dokumentu biznesowego jest inicjowany proces innego dokumentu biznesowego przypisanego do bloku podprocesu synchronicznego. Z procesu głównego do podprocesu jest przekazywana wartość, a dla przejścia z podprocesu do procesu głównego określa się status dokumentu w podprocesie.



Model procesu głównego, który będzie inicjował podproces

Na panelu właściwości bloku podprocesu określa się *mapowanie*, to jest przyporządkowanie pól dokumentu w procesie głównym do pól dokumentu w podprocesie. Dzięki temu możliwe jest przekazywanie wartości, które mogą być potrzebne w podprocesie.



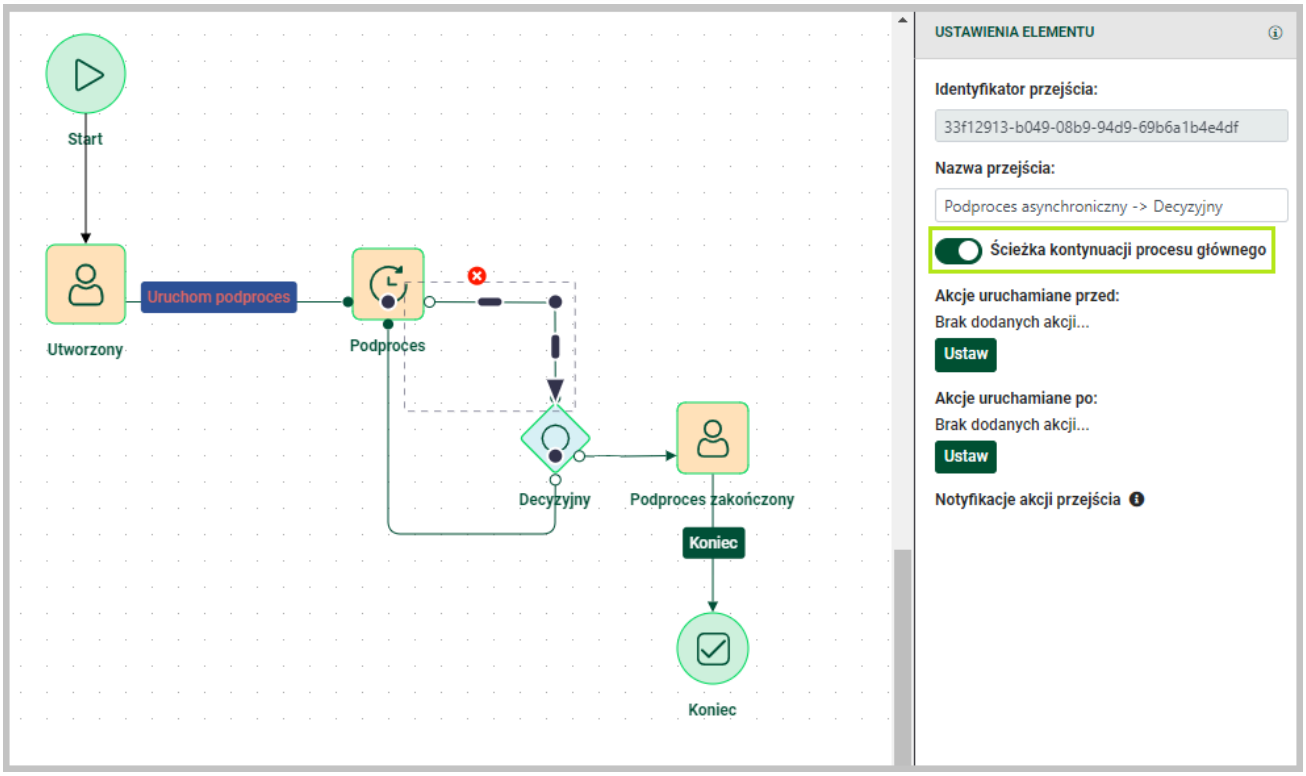
Mapowanie pola podprocesu na pole procesu głównego

Dla przejść z podprocesu do procesu głównego ustawia się status zakończenia podprocesu — w podprocesie można określić kilka końcowych statusów dokumentu i dla każdego z nich przejście z podprocesu będzie „trafiać” w inne miejsce, to jest do innego statusu, w procesie głównym.

W przypadku podprocesu asynchronicznego, dla jednego z przejść wychodzących można włączyć przełącznik `Kontynuacja procesu głównego`. To przejście przekazuje sterowanie do procesu głównego, umożliwiając jego kontynuowanie, bez względu na stan przetwarzania w podprocesie.

UWAGA

Przejście procesu głównego z bloku podprocesu asynchronicznego można poprowadzić do bloku synchronizatora lub do bloku decyzyjnego. W tym drugim przypadku możliwe jest utworzenie pętli w celu wygenerowania kilku dokumentów. Bloku decyzyjnego można także użyć jako bloku pośredniczącego w celu poprowadzenia procesu głównego do innych bloków, na przykład zadania ręcznego.



Kontynuacja procesu głównego

Jeśli w procesie głównym konieczna jest synchronizacja ze statusami dokumentów w podprocesie, należy użyć bloku `Synchronizator`. Rolą tego bloku jest synchronizacja przetwarzania w procesie głównym i w podprocesie. Należy do niego poprowadzić przejścia z procesu głównego i z podprocesu i skonfigurować *warunek synchronizacji*, to jest zapytanie SQL, które zwrócić wartość *1*, gdy warunek zostanie spełniony. Wówczas nastąpi przejście procesu głównego do kolejnego statusu. W ogólności w warunku należy sprawdzić, czy odpowiednia liczba dokumentów w podprocesie ma oczekiwany status oraz, czy dokument procesu głównego ma status odpowiadający statusowi bloku synchronizatora. Warunek synchronizacji jest sprawdzany, gdy dokument w podprocesie osiągnie oczekiwany status oraz gdy proces główny osiągnie status synchronizatora.

Właściwości bloków `Proces synchroniczny`, `Proces asynchroniczny` i `Synchronizator` opisano w kolejnych tematach.

DOBRE PRAKTYKI PROJEKTOWE

W przypadku najprostszego podprocesu (Start -> Utworzony) konieczne jest wstawienie po bloku start bloku oczekiwania z czasem oczekiwania 0 s. W przeciwnym razie wywołanie tego podprocesu w procesie głównym nie powiedzie się (zarówno synchronicznie, jak i asynchronicznie).

Dodatkowo zaleca się używanie bloku oczekiwania 0 s przed wejściem do bloku podprocesu asynchronicznego. Zastosowanie go spowoduje, zakończenie transakcji rozpoczętej przez użytkownika, na czas której jest włączana blokada ekranu (wirujący symbol operacji w toku). Dodatkowo, jeśli podprocesy są wywoływane asynchronicznie w pętli, każdy przebieg tej pętli będzie wykonywany w osobnej transakcji. W takim przypadku błąd w kolejnej iteracji nie wpłynie na dokumenty podprocesu utworzone wcześniej.

Powiązane tematy:

- Blok oczekiwania w obsłudze podprocesów

5.4.14. Podproces synchroniczny

Podsumowanie:
Blok `Podproces synchroniczny` służy do wywołania zdefiniowanego osobno procesu dokumentu biznesowego. Po wywołaniu podprocesu dokument w procesie głównym przyjmuje status tego bloku i czeka, aż dokument w podprocesie zmieni status na zadeklarowany we właściwościach jednego z przejść wychodzących z tego bloku.

Oprócz właściwości opisanych osobno, dla bloku podprocesu asynchronicznego i wychodzących z niego przejść konieczne jest określenie specyficznych właściwości.

BLOK:

- `Dokument podprocesu`: lista rozwijana umożliwiająca wybór definicji dokumentu, której diagram procesu zostanie użyty jako podproces; zainicjowanie podprocesu z procesu głównego powoduje utworzenie nowego dokumentu zgodnego z wybraną definicją.
- `Przekieruj do formularza podprocesu po zainicjowaniu podprocesu`: po włączeniu pozwala wybrać z listy jedną ze zdefiniowanych globalnych akcji otwierania formularza lub zdefiniować nową akcję tego typu. Akcja ma otworzyć formularz dokumentu podprocesu, kiedy użytkownik kliknie na formularzu

dokumentu procesu głównego przycisk powodujący przejście do statusu bloku podprocesu.

> **UWAGI**
>
> - W oknie definiowania akcji należy wybrać opcję `wskazany dokument` i w polu tekstowym wpisać zmienną ***{@Id}***.
> - W przypadku definiowania nowej akcji, po jej zapisaniu i zamknięciu okna dialogowego należy ją wybrać z listy. W przeciwnym razie akcja nie zostanie wybrana.
> - Jeśli pomimo włączenia przełącznika, akcja otwierania formularza nie zostanie wybrana, po ponownym wyświetleniu właściwości tego bloku przełącznik będzie wyłączony.

- `Własna akcja do zapisu podprocesu`: po włączeniu pozwala wybrać akcję `SQL zapisu danych`, która zostanie użyta do zapisania nowego dokumentu w podprocesie; ta akcja musi zostać zdefiniowana wcześniej.

- `Konfiguracja mapowania zmiennych przy zapisie`: ustawienie dostępne, o ile nie wskazano własnej akcji zapisu podprocesu; po kliknięciu przycisku `Edytuj` otwiera okno dialogowe, w którym mapuje się pola dokumentów podprocesu i procesu głównego:
  - `Nazwa parametru`: nazwa pola dokumentu tworzonego w podprocesie.
  - `Wartość`: zmienna odpowiadająca nazwie pola dokumentu w procesie głównym; składnia *{@NazwaPola}*, u góry okna podano listę zmiennych reprezentujących dostępne pola dokumentu w procesie głównym.

> **UWAGA**
> W tym polu można wpisać stałą wartość, zmienną typu *{@Nazwa}* lub wyrażenie SQL; w ostatnim przypadku należy kliknąć przycisk ze strzałkami z prawej strony pola.

PRZEJŚCIE:

- `Status zakończenia podprocesu`: wybierz status dokumentu w podprocesie, dla którego proces główny przejdzie do bloku docelowego tego przejścia.
  - `Dodaj`: otwiera okno dialogowe `Tworzenie statusu`, w którym można utworzyć nowy status; po utworzeniu statusu należy wybrać go z listy.
  - `Edytuj`: otwiera okno dialogowe `Edycja statusu`, w którym można zmodyfikować status wybrany na liście.

WSKAZÓWKA
Z bloku podprocesu synchronicznego można wyprowadzić kilka przejść wychodzących i dla każdego z nich wybrać inny status dokumentu w podprocesie — dotyczy to tylko statusów końcowych przetwarzania dokumentu w podprocesie.

Powiązane tematy:

- Wspólne właściwości bloków i przejść

5.4.15. Podproces asynchroniczny

Podsumowanie:
Blok `Podproces asynchroniczny` służy do wywołania zdefiniowanego osobno procesu dokumentu biznesowego. Ten blok nie ma definiowanego statusu ze względu na asynchroniczny tryb wywołania podprocesu. Proces główny przechodzi do statusu bloku wskazanego przez przejście z włączonym przełącznikiem `Kontynuacja procesu głównego`.

UWAGA
Według stanu dla wersji nAxiom 1.12 to przejście można poprowadzić do bloku synchronizatora lub do bloku decyzyjnego. W tym drugim przypadku możliwe jest wygenerowanie jednego lub kilku dokumentów w podprocesie i przekazanie sterowania do procesu głównego.

Oprócz właściwości opisanych osobno, dla bloku podprocesu asynchronicznego i wychodzących z niego przejść konieczne jest określenie specyficznych właściwości.

BLOK:

`Dokument podprocesu`: lista rozwijana umożliwiająca wybór definicji dokumentu, której diagram procesu zostanie użyty jako podproces; zainicjowanie podprocesu z procesu głównego powoduje utworzenie nowego dokumentu zgodnego z wybraną definicją.

- `Własna akcja do zapisu podprocesu`: po włączeniu pozwala wybrać akcję `SQL zapisu danych`, która zostanie użyta do zapisania nowego dokumentu w podprocesie; ta akcja musi zostać zdefiniowana wcześniej.

- `Konfiguracja mapowania zmiennych przy zapisie`: ustawienie dostępne, o ile nie wskazano własnej akcji zapisu podprocesu; po kliknięciu przycisku `Edytuj` otwiera okno dialogowe, w którym mapuje się pola dokumentów podprocesu i procesu głównego:
    - `Nazwa parametru`: nazwa pola dokumentu tworzonego w podprocesie.
    - `Wartość`: zmienna odpowiadająca nazwie pola dokumentu w procesie głównym; składnia *{@NazwaPola}*, u góry okna podano listę zmiennych reprezentujących dostępne pola dokumentu w procesie głównym.

> **UWAGA**
> W tym polu można wpisać stałą wartość, zmienną typu *{@Nazwa}* lub wyrażenie SQL; w ostatnim przypadku należy kliknąć przycisk ze strzałkami z prawej strony pola.

PRZEJŚCIE:

- `Ścieżka kontynuacji procesu głównego`: przełącznik, który może zostać włączony tylko dla jednego przejścia wychodzącego z bloku procesu asynchronicznego; powoduje przejście procesu głównego do statusu bloku docelowego dla tego przejścia; jeśli przełącznik jest włączony, lista wyboru `Status zakończenia podprocesu` jest niedostępna.

> **UWAGA**
> W związku z ograniczeniami występującymi w wersji 1.12 po wywołaniu podprocesu asynchronicznego proces główny może przejść do statusu synchronizatora lub do statusu bloku umieszczonego po bloku decyzyjnym (na wyjściu z pętli blok podprocesu asynchronicznego-blok decyzyjny).

-
    - `Status zakończenia podprocesu`: wybierz status dokumentu w podprocesie, dla którego proces główny przejdzie do bloku docelowego tego przejścia.
    - `Dodaj`: otwiera okno dialogowe `Tworzenie statusu`, w którym można utworzyć nowy status; po utworzeniu statusu należy wybrać go z listy.
    - `Edytuj`: otwiera okno dialogowe `Edycja statusu`, w którym można zmodyfikować status wybrany na liście.

> **WSKAZÓWKA**
> Z bloku podprocesu synchronicznego można wyprowadzić kilka przejść wychodzących i dla każdego z nich wybrać inny status dokumentu w podprocesie — dotyczy to tylko statusów końcowych przetwarzania dokumentu w podprocesie.

Powiązane tematy:

- Wspólne właściwości bloków i przejść

5.4.16. Synchronizator

Podsumowanie:
Blok `Synchronizator` służy do synchronizacji przetwarzania w procesie głównym i w podprocesie asynchronicznym. Należy do niego poprowadzić przejścia z procesu głównego i z podprocesu oraz skonfigurować *warunek synchronizacji*. Ten warunek to zapytanie SQL, które musi zwrócić wartość *1*, gdy warunek zostanie spełniony. Wówczas nastąpi przejście procesu głównego do kolejnego statusu. W ogólności w warunku można sprawdzić, czy odpowiednia liczba dokumentów w podprocesie ma oczekiwany status.
Warunek jest sprawdzany za każdym razem, kiedy dokument w podprocesie osiągnie status ustawiony na przejściu podprocesu z bloku podprocesu asynchronicznego.

Oprócz właściwości opisanych osobno, dla bloku synchronizatora konieczne jest określenie specyficznych właściwości.

- `Warunek synchronizacji`: wyrażenie SQL, które ma zwrócić wartość `0` lub `1`:
  - *0* — status dokumentu w procesie głównym nie zmienia się, synchronizator czeka na zakończenie pozostałych podprocesów asynchronicznych lub na proces główny;
  - *1* — status dokumentu w procesie głównym zostaje zmieniony na status bloku, do którego prowadzi przejście wybrane przez warunek we właściwości `Kierunek przejścia` w bloku synchronizatora; podprocesy asynchroniczne zainicjowane dla tego dokumentu głównego zostały zakończone lub ich dalsze przetwarzanie nie będzie miało wpływu na proces główny.
- `Kierunek przejścia`: w tym polu wpisz kod zapytania SQL, które zwróci nazwę przejścia w zależności od wyniku wyrażenia warunkowego użytego w zapytaniu; kliknij przycisk `Konstruktor zapytania`, aby skonstruować zapytanie przy użyciu konstruktora (wynikowy kod SQL zostanie wyświetlony w polu tekstowym, gdzie można go modyfikować); patrz Okno Konstruktor SQL - definiowanie warunków przejść bloku decyzyjnego .

Powiązane tematy:

- Wspólne właściwości bloków i przejść

5.5. DEFINICJE STATUSÓW

Podsumowanie:
Status określa etap w cyklu przetwarzania dokumentu. W typowym przebiegu procesu biznesowego wykonanie określonych czynności na danym etapie kończy się kliknięciem przycisku zmiany statusu i przejściem dokumentu do następnego etapu. Status dokumentu może także zostać zmieniony automatycznie w wyniku działania akcji zmiany statusu. Status jest atrybutem dokumentu w standardowym modelu danych, dokumenty w ramach definicji opartych na modelu uproszczonym nie mają statusu.

☛ *AdminSPA* > `PROCESY BIZNESOWE` > `Definicje statusów`

Polecenie `Definicje statusów` wyświetla listę definicji statusów dla wszystkich definicji dokumentów w danym środowisku nAxiom. Na liście statusy są domyślnie pogrupowane według definicji dokumentu biznesowego. W każdym wierszu znajdują się przyciski do edycji definicji i jej usunięcia.

UWAGA
Usunięcie statusu dokumentu jest możliwe tylko wtedy, gdy w bazie danych nie ma dokumentów, którym przypisano ten status. Jeśli takie dokumenty istnieją, należy je usunąć przed usunięciem statusu.

Aby dodać nowy status, kliknij przycisk `Nowy status` nad listą z lewej strony i wypełnij pola w wyświetlonym oknie dialogowym.

- `Kod`: wpisz dowolny ciąg znaków identyfikujący tworzony status dokumentu.
- `Opis`: wpisz opis tworzonego typu dokumentu; opis jest wyświetlany na diagramie workflow i na listach rozwijanych jako nazwa statusu.
- `Definicja dokumentu`: wybierz definicję dokumentu biznesowego, dla której ma zostać utworzony status dokumentu; na liście są wyświetlane tylko definicje oparte na modelu standardowym.
- `Aktywny`: wyłącz ten przełącznik, aby bieżący status dokumentu stał się nieaktywny.
- `Zapisz`: kliknij, aby zapisać zmiany i zamknąć okno.
- `Anuluj`: kliknij, aby zamknąć okno bez zapisywania zmian.

WSKAZÓWKA
Statusy można również tworzyć i edytować w edytorze diagramu procesu.

Powiązane tematy:

- Zadanie ręczne
- Zmiana statusu dokumentów
- Modele danych w nAxiom

5.6. MODELE DECYZYJNE DMN

Podsumowanie:
Modele decyzyjne służą do modelowania procesów decyzyjnych. Model takiego procesu buduje się w edytorze diagramów. Diagramy procesów decyzyjnych składają się z bloków zmiennych wejściowych oraz bloków decyzji. Dostępne są dwa typy decyzji: wyrażenia literałowe i tabele decyzyjne. Modele decyzyjne są uruchamiane za pomocą akcji Uruchom model decyzyjny .

☛ *AdminSPA* > `PROCESY BIZNESOWE` > `Modele decyzyjne`

- Na ekranie `Lista modeli decyzyjnych` kliknij przycisk `Nowy model decyzyjny`, wybierz aplikację i moduł oraz wypełnij pola `Kod`, `Nazwa` i `Opis` i kliknij przycisk `Zapisz`. Zostanie utworzona wersja modelu w statusie *Roboczy*.
- Następnie na liście modeli decyzyjnych kliknij symbol + z lewej strony w wierszu utworzonego modelu, aby rozwinąć listę wersji tego modelu. Kliknij przycisk `Designer`, aby przejść do projektowania modelu decyzyjnego w edytorze diagramów (aplikacja *BpmnSPA* wyświetlana na osobnej karcie przeglądarki).
- Aby móc uruchamiać gotowy model decyzyjny, należy w menu przycisku akcji dla wersji wybrać polecenie `Opublikuj`. Spowoduje to zmianę statusu wersji z *Roboczy* na `Opublikowany`. W tym statusie nie można zmienić projektu modelu, ale można zmieniać jego metadane (kod, nazwa, opis).
- Ponadto, w menu akcji są dostępne następujące polecenia:
  - `Zapisz jako nową wersję roboczą`: dla wersji w statusie `Opublikowany`; tworzy kopię w statusie `Roboczy`, wymagane jest podanie nowego kodu w metadanych wersji.
  - `Archiwizuj`: zmienia status wersji na `Zarchiwizowany`.
  - `Usuń`: usuwa wersję w statusie `Zarchiwizowany` i `Roboczy`.

- `Przywróć roboczy`: zmienia status wersji ze `Zarchiwizowany` na `Roboczy`.
- `Edytuj`: otwiera okno edycji metadanych wersji (kod, nazwa, opis).

Do projektowania modeli decyzyjnych w nAxiom zaimplementowano edytor ***dmn-js*** (https://github.com/bpmn-io/dmn-js) udostępniany jako oprogramowanie open source. Szczegółowa dokumentacja jest dostępna pod adresem https://docs.camunda.io/docs. Edytor zapisuje konfigurację diagramu w pliku XML, który jest następnie przekazywany do silnika Common DMN Engine (https://github.com/adamecr/Common.DMN.Engine) w celu wyliczenia. Ten silnik odpowiada także za obliczanie wyrażeń definiowanych w modelu. Używa do tego interpretera DynamicExpresso (https://github.com/dynamicexpresso). Interpreter obsługuje proste wyrażenia w języku C# oraz podzbiór języka FEEL. Do testowania wyrażeń można używać interpretera w wersji online dostępnego pod adresem https://dynamic-expresso.azurewebsites.net/.

Specyfikację notacji DMN można znaleźć pod adresem https://www.omg.org/dmn/.

W kolejnym temacie opisano podstawowe zasady budowy modeli decyzyjnych.

5.6.1. Edytor modeli decyzyjnych DMN

Podsumowanie:
Edytor jest otwierany w widoku diagramu DRD (Decision Requirement Diagram), w który przebieg procesu decyzyjnego modeluje się, używając standardowych elementów notacji DMN. Elementy te są widoczne w przyborniku z lewej strony. Są to (od góry):

- blok decyzji,
- blok zmiennej wejściowej,
- blok źródła wiedzy: służy do referencji autorytatywnego źródła, na podstawie którego zbudowano daną decyzję; ten blok nie ma żadnej semantyki wykonania i jest ignorowany podczas wyliczania decyzji,
- blok modelu wiedzy (nieobsługiwany).

W TYM TEMACIE

- Widok diagramu modelu decyzyjnego
- Blok zmiennej wejściowej
- Blok decyzji: wyrażenie literałowe
- Blok decyzji: tabela decyzyjna
- Zmienne wejściowe
- Zmienne wyjściowe
- Reguły
- Polityka trafień

WIDOK DIAGRAMU MODELU DECYZYJNEGO

Po kliknięciu przycisku `Designer` dla nowo utworzonej wersji modelu decyzyjnego wyświetlany jest edytor diagramów modeli decyzyjnych. W obszarze roboczym edytora jest wyświetlany blok decyzji o nazwie odpowiadającej nazwie modelu decyzyjnego. W ogólności projektowanie diagramu polega na umieszczeniu na diagramie odpowiednich bloków, nadaniu im nazw (opcjonalnie; kliknij dwukrotnie blok i wpisz nazwę), połączeniu ze sobą i zdefiniowaniu decyzji. Kliknięcie bloku na diagramie powoduje wyświetlenie menu kontekstowego z następującymi ikonami (zależnie od typu bloku):

- klucz płaski: wyświetla menu wyboru typu decyzji, dostępne wartości:
  - `Decision table`: tabela decyzyjna, może przetwarzać wiele reguł biznesowych dla wielu zmiennych wejściowych i zwracać wiele zmiennych wyjściowych.
  - `Literal expression`: wyrażenie literałowe, pojedyncze wyrażenie, które zwraca jedną zmienną wyjściową typu prostego.
- decyzja, dane wejściowe: kliknięcie powoduje utworzenie odpowiedniego bloku i połączenie z bieżącym blokiem (w obu przypadkach połączenie jest skierowane do bieżącego bloku).
- strzałki: służą do łączenia danych wejściowych i decyzji na diagramie i pokazują przepływ zmiennych, aby zmienna wejściowa była dostępna w decyzji, blok danej wejściowej musi być połączony z blokiem decyzji.
- otwierający nawias prostokątny: komentarz do bloku, może poprawić czytelność diagramu.
- kosz: usuwa dany blok z diagramu.

Z prawej strony okna edytora wyświetlane są właściwości aktywnego bloku lub modelu decyzyjnego.

WAŻNE
Aby opublikować wersję modelu decyzyjnego, należy w jej metadanych wskazać końcową decyzję, której wynik będzie wynikiem wykonania całego modelu. Tę decyzję wybiera sie z listy rozwijanej nazw decyzji zdefiniowanych w modelu (`Edytuj` > `Wywoływana decyzja`). Z tego powodu, chociaż nazwy decyzji nie są obowiązkowe, finalna decyzja musi mieć nazwę, aby można było wybrać ją z listy.

Aby przejść do definicji decyzji, należy kliknąć symbol decyzji w lewym górnym rogu jej bloku. Poniżej opisano podstawowe zasady definiowania poszczególnych bloków.

BLOK ZMIENNEJ WEJŚCIOWEJ

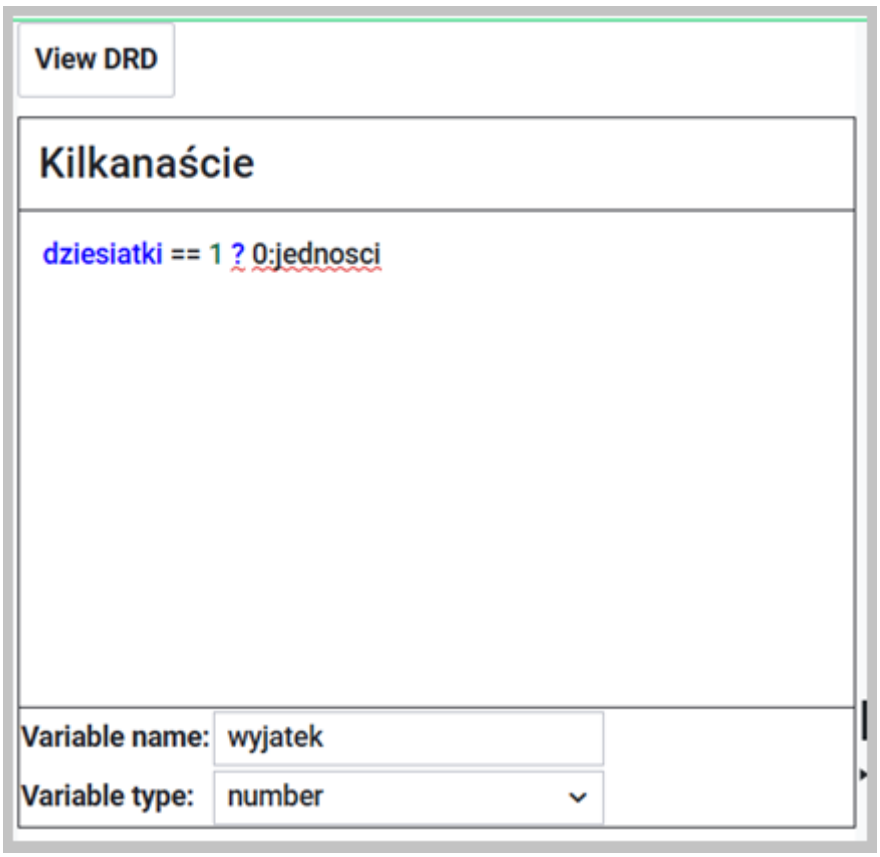
Ten blok służy do zdefiniowania nazwy zmiennej wejściowej, która zostanie przekazana do połączonej decyzji nadrzędnej. W nAxiom dane wejściowe są przekazywane do modelu przez akcję Uruchom model decyzyjny. W definicji akcji są wyświetlane nazwy zmiennych wejściowych z modelu decyzyjnego. Nazwy zmiennych wejściowych przekazywanych przez akcję muszą być zgodne z nazwami zmiennych wejściowych w modelu. Na przykład jeśli w modelu zdefiniowano zmienną o nazwie *Input1* i ma do niej zostać przekazane pole z kontekstu formularza o nazwie *KwotaFaktury*, należy zmapować tę nazwę zapytaniem SQL w definicji akcji:

```
SELECT {@KwotaFatury} AS [Input1]
```

WAŻNE
Zmienne wejściowe są niemodyfikowalne; nie można im przypisać innej wartości niż przekazana do modelu.

BLOK DECYZJI: WYRAŻENIE LITERAŁOWE

Wyrażenie literałowe może zawierać jedno wyrażenie, które zwraca pojedynczą wartość określonego typu prostego. W widoku definicji wyrażenia literałowego należy podać nazwę zmiennej wyjściowej i jej typ oraz wpisać samo wyrażenie. Przedstawione poniżej wyrażenie literałowe przypisuje zmiennej wyjściowej wartość *0*, gdy zmienna *dziesiatki* jest równa *1* lub wartość zmiennej *jednosci* w przeciwnym razie.



Wyrażenie literałowe

BLOK DECYZJI: TABELA DECYZYJNA

Jest to tabela (macierz) zawierająca reguły, czyli warunki dotyczące zmiennych wejściowych. Jeśli reguła jest spełniona, zmiennej wyjściowej jest przypisywana określona wartość (wynik reguły). Zasadę działania ilustruje nagłówek macierzy: *When: Zmienna_wej1 (AND Zmienna_wej2...) then: Zmienna_wyj1 (AND Zmienna_wyj2...)*. W nagłówkach kolumn należy zdefiniować zmienne wejściowe oraz zmienne wyjściowe, a w kolejnych wierszach podać reguły dla zmiennych wejściowych oraz wyrażenia do obliczenia zmiennych wyjściowych. Jedna reguła może generować wartości dla kilku zmiennych wyjściowych.

Widok pustej tabeli decyzyjnej zawiera jedną kolumnę dla zmiennej wejściowej, jedną dla zmiennej wyjściowej i jedną kolumnę adnotacji. Aby dodać kolejne kolumny zmiennych, należy klikać symbol + w kółku na pionowej linii dzielącej kolumny.

ZMIENNE WEJŚCIOWE

Dwukrotne kliknięcie nagłówka z placeholderem *Input* otwiera okienko z następującymi polami:

- `Input`: nazwa wyświetlana zmiennej wejściowej.
- `Expression`: nazwa zmiennej wejściowej (nazwa bloku danych wejściowych) lub

wyrażenie operujące na tej zmiennej.

- `Type`: typ zmiennej wejściowej lub wyniku wyrażenia.
- `Predefined values`: (opcjonalnie) w przypadku gdy w regułach mają być sprawdzane tylko określone wartości zmiennej wejściowej, można je zdefiniować w tym miejscu, aby sprawnie definiować reguły.

ZMIENNE WYJŚCIOWE

Po dwukrotnym kliknięciu nagłówka można zdefiniować nazwę wyświetlaną zmiennej, nazwę i typ zmiennej oraz jej predefiniowane wartości, podobnie jak dla zmiennych wejściowych.

REGUŁY

Pod nagłówkiem tabeli decyzyjnej, w kolejnych wierszach wpisuje się *reguły*. Idea jest następująca: jeśli wartości zmiennych wejściowych w poszczególnych kolumnach spełniają zdefiniowane w danym wierszu warunki, zmiennej wyjściowej zostanie przypisana wartość wskazana w tym wierszu, jak na ilustracji poniżej.

View DRD

koniec | Hit policy: Unique

	When Dzień tygodnia string	And Pogoda string	Then Plan dnia string	Annotations
1	"niedziela"	"słońce"	"idź na spacer"	
2	"niedziela"	"deszcz"	"czytaj książkę"	
3	"sobota"	-	"zrób coś w domu"	
4	not("sobota", "niedziela")	-	"ugotuj obiad na jutro"	
+	-	-		

Tabela decyzyjna

Dozwolone warunki reguł dla zmiennych wejściowych zależą od typu zmiennej, a do ich wpisywania służy okienko dialogowe wyświetlane po kliknięciu ikony ołówka. Na przykład dla typu *string* dostępne są następujące warunki:

- `Match one`: zwraca *true*, jeśli wartość zmiennej jest taka, jak jedna z podanych wartości
- `Match none`: zwraca *true*, jeśli wartość zmiennej jest inna niż podane wartości

Dla zmiennych typu *number* dostępne są warunki porównania lub przynależności do zakresu.

WSKAZÓWKA

Operatorem nierówności jest *not()*, np. *not(13)* odpowiada nierówności *!=13*.

WAŻNE

Jeśli zmienna wejściowa nie ma wartości (*NULL*), silnik DMN przypisze jej wartość domyślną dla danego typu (np. *0* dla wartości liczbowych). Należy mieć to na uwadze podczas projektowania decyzji. W ogólności nie można w regule sprawdzić, czy zmienna wejściowa ma wartość.

WAŻNE

Tabela DMN obsługuje *NULL*, jeśli to zostanie przewidziane na etapie projektowania. Wymaga to wpisania wprost łącznika (-) lub *NULL* w komórce wartości wejściowej. W takim przypadku polityki trafień będą działać zgodnie z definicją.

Jeśli obsługa NULL w regułach nie zostanie wprowadzona, w przypadku braku wartości wejściowej tabela nie zwróci żadnego wyniku. Proces nie zostanie przerwany błędem, a jedynie zaloguje na poziomie *debug* komunikat: *„The decision '{nazwa decyzji}' for decision model version '{kod wersji}' completed without a result. No matching rules found."* w logu serwisu API. Dotychczas w niektórych przypadkach brak wartości wejściowej mógł skutkować błędem i przerwaniem procesu. Jeśli wynik tabeli jest niezgodny z oczekiwaniami, należy sprawdzić w logu, czy nie występuje powyższy wyjątek.

Uwaga: Od wersji 1.15.7 nie zachodzi niejawna konwersja NULL na wartość domyślną dla konkretnego typu danych (pusty tekst dla danych tekstowych lub zero dla danych liczbowych).

Cały proces obliczania macierzy decyzyjnej jest logowany na poziomie *trace* w logu serwisu API.

Wynik reguły, czyli wartość zmiennej wyjściowej może być literałem lub wyrażeniem. Wyrażenie może operować na wartościach zmiennych wejściowych.

POLITYKA TRAFIEŃ

Ważnym parametrem dotyczącym obliczania wyniku tabeli decyzyjnej jest *polityka trafień* (ang. *hit policy*). W zależności od wartości tego ustawienia tabela zwraca wynik dla tylko jednej reguły, dla zestawu reguł lub agregat wyników dla wielu reguł. Dostępne wartości tego parametru to:

- `Unique`: reguły tabeli są rozłączne (nie pokrywają się), w przeciwnym razie jest zwracany błąd; zwracany jest wynik pasującej reguły.

- `First`: reguły mogą się pokrywać i są sprawdzane od pierwszej do ostatniej, zwracany jest wynik pierwszej pasującej reguły.
- `Priority`: (nieobsługiwane) reguły mogą się pokrywać, zwracany jest wynik reguły o najwyższym priorytecie wartości wyjściowej;
- `Any`: wszystkie pasujące reguły muszą zwracać ten sam wynik, w przeciwnym razie zwracany jest błąd; zwracany jest wynik dowolnej pasującej reguły.
- `Collect`: zwracana jest nieuporządkowana lista wartości wyjściowych obliczonych dla pasujących reguł; może być używana z funkcjami agregującymi *SUM*, *MIN*, *MAX* i *COUNT*; uwagi:
    - jeśli ta polityka trafień ma być używana z funkcją agregującą, tabela decyzyjna może mieć tylko jedną zmienną wyjściową (jedną kolumnę wartości wyjściowych),
    - dla funkcji agregujących innych niż *COUNT*, zmienna wyjściowa musi być typu liczbowego lub logicznego,
    - dla zmiennych wyjściowych typu tekstowego funkcja *COUNT* zwraca liczbę reguł jako tekst,
    - agregatory *SUM*, *MIN* i MAX* mogą być używane dla zmiennych wyjściowych typu logicznego:
        - funkcje *SUM* i *MAX* zwracają *true*, jeśli co najmniej jedna pasująca reguła zwraca *true*,
        - funkcja *MIN* zwraca *false*, jeśli co najmniej jedna pasująca reguła zwraca *false*.
- `Rule order`: zwracana jest lista wartości wyjściowych obliczonych dla pasujących reguł, lista jest uporządkowana w kolejności reguł w tabeli decyzyjnej.
- `Output order`: (nieobsługiwane) zwracana jest lista wartości wyjściowych obliczonych dla pasujących reguł uporządkowana malejąco według priorytetu zmiennych wyjściowych.

5.6.2.

6. NARZĘDZIA

W tej sekcji opisano zagadnienia, których dotyczą poleceń widocznych w sekcji `NARZĘDZIA` w aplikacji *AdminSPA*.

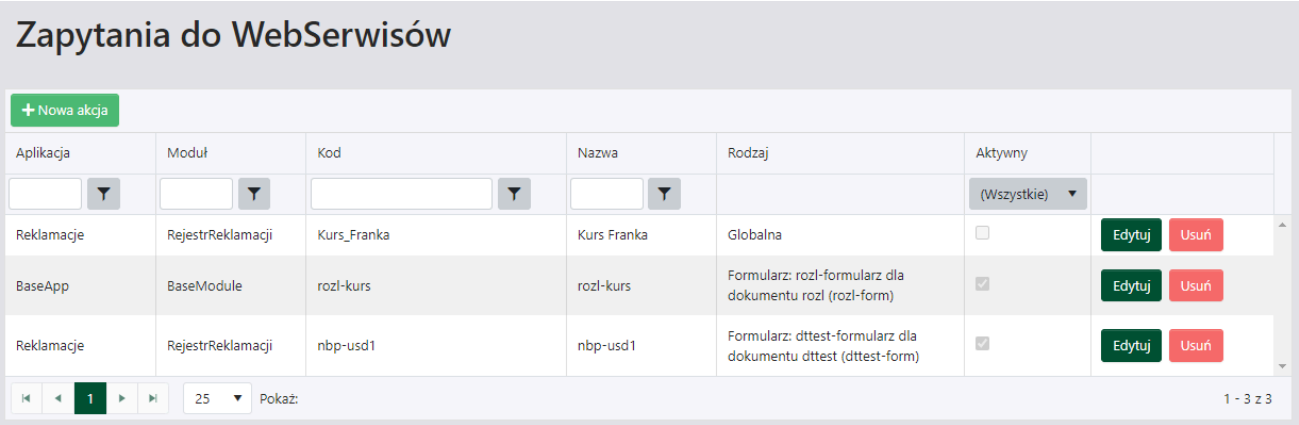
6.1. AKCJE

Podsumowanie:

- Akcje w nAxiom to elementy implementacji modelu procesu biznesowego, które automatyzują wykonywanie różnych zadań, takich jak operacje bazodanowe, import i eksport danych, konwersja i szyfrowanie danych, generowania raportów, obsługa funkcji OCR i wiele innych. Akcje są wyzwalane przez różne zdarzenia, na przykład kliknięcie przycisku, zmianę statusu, a także przez inne akcje.
- Akcje można definiować *globalnie* w aplikacji i używać w różnych miejscach, w razie potrzeby modyfikując akcję i zapisując jako kopię lokalną. Osobna kategoria to akcje systemowe, które występują wyłącznie jako akcje lokalne w miejscu użycia. Te akcje zwykle nie wymagają żadnej konfiguracji, na przykład akcja odświeżenia strony.
- Sekwencje niektórych typów akcji mogą być wykonywane w transakcji — wtedy niepowodzenie jednej z akcji spowoduje wycofanie zmian wprowadzonych przez akcje poprzedzające.

☛ *AdminSPA* > `NARZĘDZIA` > `Akcje`

Kliknięcie menu `Akcje` w sekcji `NARZĘDZIA` rozwija menu podrzędne z typami akcji, które można zdefiniować. Po wybraniu typu akcji jest wyświetlana lista zdefiniowanych akcji tego typu. Obok każdej akcji znajdują się przyciski usuwania i edycji, w lewym górnym rogu znajduje się przycisk `Nowa akcja`, który wyświetla okno dialogowe definiowania nowej akcji. Akcje definiowane w tym miejscu to tak zwane *akcje globalne*. Można ich używać w różnych miejscach w budowanej aplikacji.

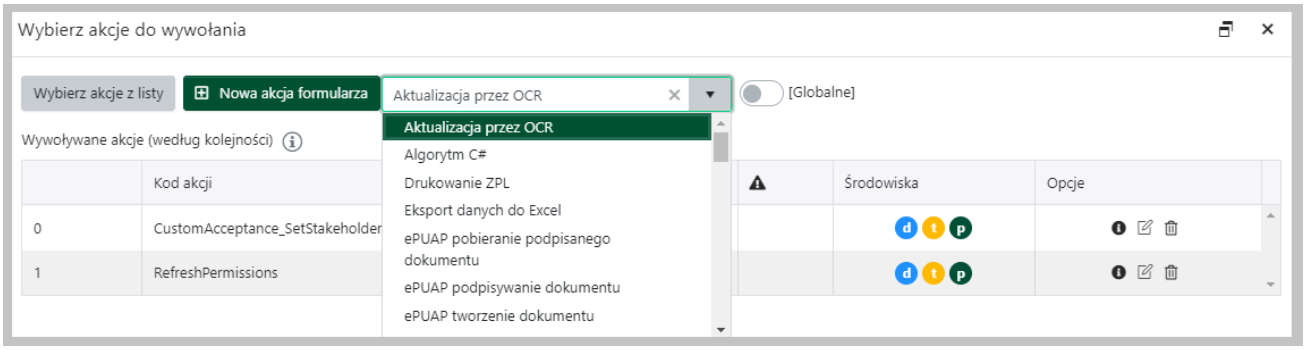


Przykład listy akcji Zapytania do Webserwisów

Akcje globalne są dostępne podczas projektowania interfejsu użytkownika, np. we właściwościach przycisków na formularzach i widokach list, a także podczas tworzenia modelu procesu biznesowego we właściwościach przejść na diagramie procesów.

Akcje globalne można modyfikować lokalnie w miejscu użycia i zapisywać zmiany jako lokalną kopię lub nadpisywać nimi akcję globalną.

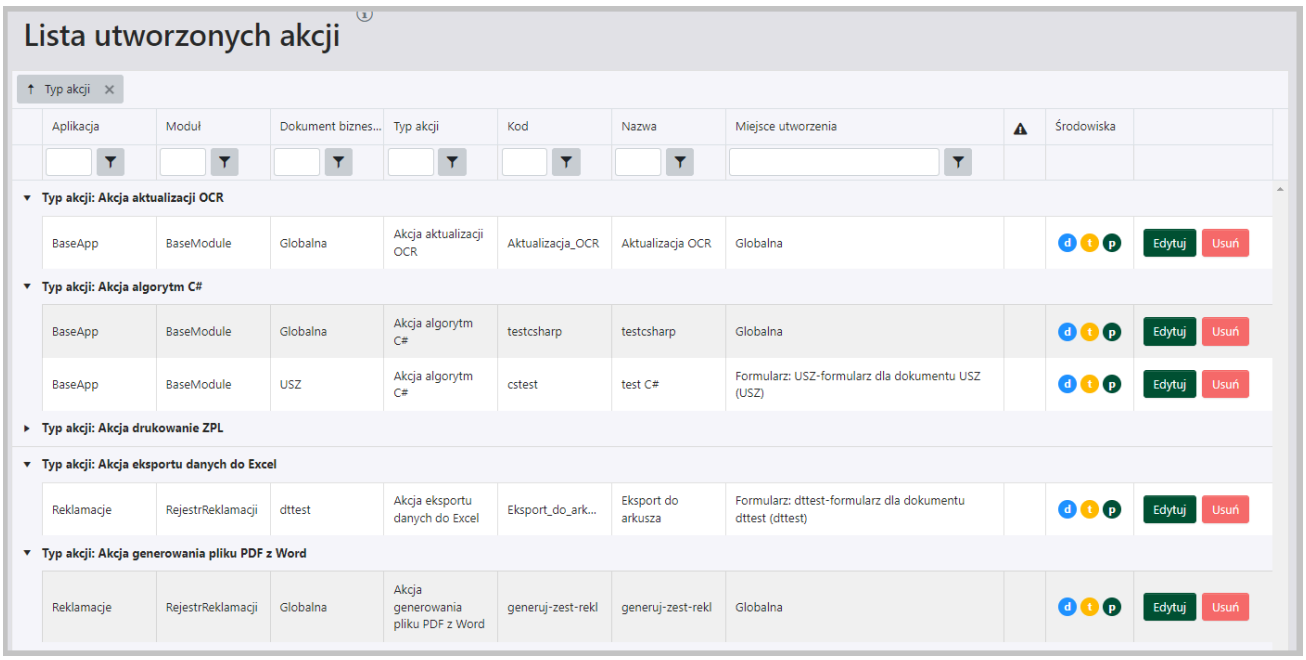
Nowe akcje globalne można także definiować wszędzie tam, gdzie można używać akcji, czyli na przykład we właściwościach przycisków i przejść na diagramie procesu.



Okno wyboru akcji we właściwościach przycisku formularza

Osobna kategoria akcji w nAxiom to *akcje systemowe*. Są to akcje, które nie mają żadnych parametrów konfiguracyjnych, na przykład akcja *Odśwież stronę* lub *Usuń rekord* lub których możliwości konfiguracji są ograniczone, na przykład *Zamknij formularz* lub *Zapisz rekord*. Akcje systemowe wybiera się z listy i ewentualnie konfiguruje wyłącznie w miejscu użycia. Takiej akcji nie można zdefiniować jako globalnej.

Pierwsza pozycja menu `Akcje` to `Lista utworzonych akcji`. Ta lista zawiera wykaz wszystkich akcji zdefiniowanych w danej witrynie nAxiomNa. Na liście można grupować, filtrować i wyszukiwać akcje. Z tej listy można przechodzić bezpośrednio do edycji akcji. Można na niej także usuwać akcje.



Lista utworzonych akcji

W przypadku akcji systemowych oraz akcji, które mają kopie lokalne, po kliknięciu przycisku `Edytuj` jest wyświetlana lista miejsc użycia danej akcji, z której można przejść do edycji akcji w miejscu użycia.

Miejsca użycia akcji

W celu zdefiniowania akcji należy określić szereg parametrów. Niektóre z tych parametrów, na przykład `Kod`, `Nazwa`, `Aplikacja`, `Moduł` i wiele innych powtarzają się w definicji wszystkich typów akcji. Te parametry opisano w osobnym temacie.

Wyzwalacze akcji w nAxiom to:

- kliknięcie przycisku na formularzu lub na liście — wyzwala akcje przypisane we właściwościach przycisku;
- zdarzenie *OnChange*, *OnLeave* i/lub *OnClick* dla pola formularza — wyzwala akcje JavaScript przypisane we właściwościach pola formularza;
- zmiana statusu dokumentu generuje dwa zdarzenia: przed zmianą statusu i po zmianie statusu, z którymi mogą być powiązane akcje (przypisane we właściwościach przejścia na diagramie procesu) — dotyczy to zarówno zmiany statusu na skutek kliknięcia przycisku na formularzu, wykonania akcji zmiany statusu, jak również żądania zmiany statusu przesłanego do interfejsu Public API, np. z wtyczki do programu Outlook;
- zadanie cykliczne — może uruchamiać niektóre typy akcji;
- akcja *Algorytm C#* — może uruchamiać niektóre typy akcji.

6.1.1. Wspólne parametry akcji

Podsumowanie:
Szereg parametrów powtarza się w definicji każdego typu akcji. Te parametry opisano osobno, aby poprawić przejrzystość opisu poszczególnych typów akcji.

☛ *AdminSPA* > `NARZĘDZIA` > `Akcje` > *typ akcji* > `Edytuj`

Dla każdej akcji należy skonfigurować następujące parametry:

`Kod akcji`: symbol akcji używany przez mechanizm asysty *smart numbers*; może zawierać tylko małe i wielkie litery (bez znaków diakrytycznych), cyfry i znaki podkreślenia (_); maksymalnie 50 znaków.

`Nazwa akcji`: nazwa akcji, która będzie widoczna na listach wyboru akcji; maksymalnie 250 znaków.

`Opis`: opis akcji; maksymalnie 1000 znaków.

`Aplikacja`: nazwa aplikacji, do której zostanie przypisany tworzony obiekt; wszystkie obiekty przypisane do aplikacji można wraz z nią wyeksportować, używając polecenia `Migrator aplikacji` w sekcji `ADMINISTRACJA` menu głównego.

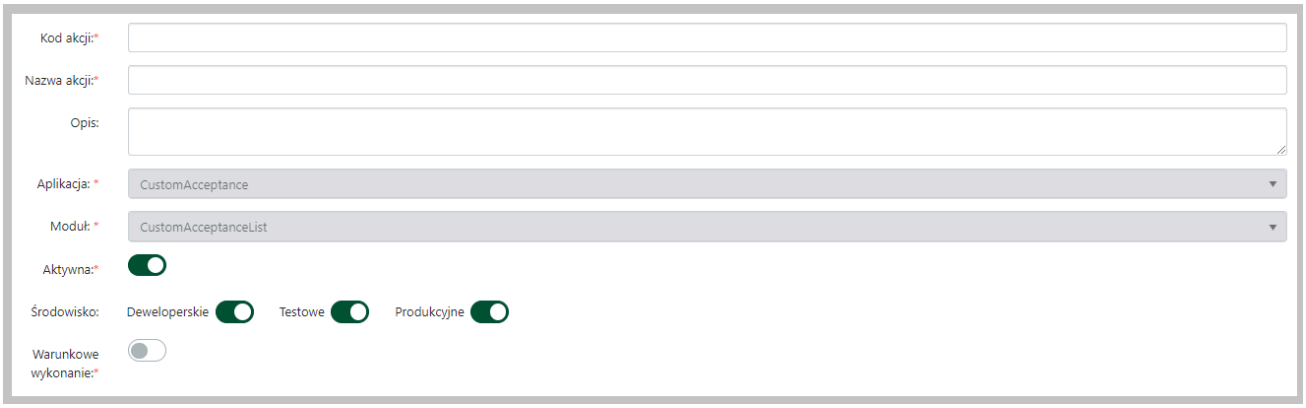
`Moduł`: nazwa modułu aplikacji, do którego zostanie przypisany tworzony obiekt.

`Aktywna`: określa, czy dana akcja jest wykonywana. Akcji nieaktywnej nie można także przypisać, chociaż jest dostępna w komponencie wyboru akcji.

`Środowisko`: przełączniki decydujące o wykonywaniu akcji w określonym środowisku; jeśli przełącznik jest wyłączony, wykonanie akcji zostanie pominięte (przy czym do aplikacji będzie zwracana informacja o pomyślnym wykonaniu); aby używać różnych wersji akcji w różnych środowiskach, należy zdefiniować te akcje, odpowiednio poustawiać przełączniki i przypisać je wszystkie np. do przycisku na formularzu; domyślnie wszystkie przełączniki są włączone:

- `Deweloperskie`: decyduje o wykonywaniu akcji w środowisku deweloperskim
- `Testowe`: decyduje o wykonywaniu akcji w środowisku testowym
- `Produkcyjne`: decyduje o wykonywaniu akcji w środowisku produkcyjnym

`Warunkowe wykonanie`: po włączeniu pozwala zdefiniować warunek wykonania akcji w formie zapytania SQL; kiedy zapytanie zwróci wartość *1*, akcja zostanie wykonana; w przeciwnym razie akcja nie zostanie wykonana, ale zostanie zwrócony status pomyślnego wykonania — kolejne akcje zostaną wykonane; aby przerwać wykonywanie sekwencji akcji, zapytanie musi zwrócić błąd, np. za pomocą funkcji *RAISERROR*.

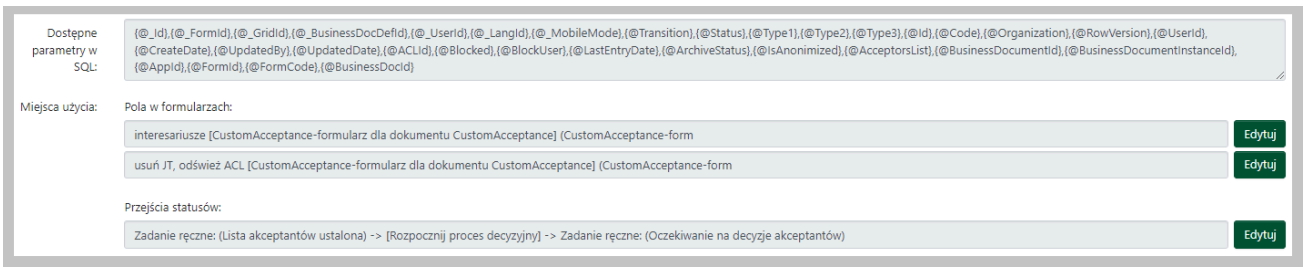


Wspólne parametry akcji

Dla niektórych typów akcji dostępne są także następujące informacje:

`Dostępne parametry w SQL`: pole tylko do odczytu, dla akcji przypisanych do formularzy wyświetla listę pól formularza, których można użyć w zapytaniu SQL jako zmiennych w formacie *{@PoleFormularza}*; dla akcji globalnych to pole jest puste.

`Miejsca użycia`: pole tylko do odczytu z wykazem pól (przycisków) i sekcji formularzy oraz przejść statusów, w których dana akcja została użyta; jeśli dana akcja nie została nigdzie przypisana, to pole nie jest wyświetlane; przycisk `Edytuj` z prawej strony pola otwiera definicję danego formularza lub diagramu procesu.



Informacje o dostępnych parametrach i miejscach użycia akcji

6.1.2. Tabela *core.Actions*

Podsumowanie:
Definicje akcji są przechowywane w tabeli systemowej *core.Actions*.

Nazwa kolumny	Typ danych	Zezwól NULL	Opis
Id	int	☐	Kolumna klucza głównego, identyfikator rekordu w tabeli
Name	nvarchar(250)	☑	Nazwa akcji
Command	nvarchar(MAX)	☑	Parametry akcji w formacie JSON lub zapytanie SQL
IsActive	bit	☐	Flaga określająca, czy akcja jest aktywna
Type	int	☐	Identyfikator typu akcji; patrz przypis 1 pod tabelą
DataSourceId	int	☑	Identyfikator źródła danych wskazanego w definicji akcji
FormDefinitionId	int	☑	Identyfikator formularza, w którym użyto danej akcji
RowGuid	uniqueidentifier	☐	Unikatowy identyfikator szablonu
ModuleDefinitionId	int	☐	Identyfikator modułu aplikacji, do którego jest przypisana akcja
CreatedOnWorkflow	bit	☐	Flaga informująca, czy akcja jest używana na diagramie procesów
RunInTransaction	bit	☐	Flaga informująca, czy akcja ma być wykonywana w transakcji
IsConditional	bit	☐	Flaga informująca, czy wykonanie akcji jest warunkowe
SqlCondition	nvarchar(MAX)	☑	Zapytanie SQL z warunkiem wykonania akcji
RunInEnvironment	int	☐	Identyfikator środowiska modułu aplikacji, do którego jest przypisana akcja; patrz przypis 2 pod tabelą
Code	nvarchar(50)	☐	Kod akcji
Description	nvarchar(1000)	☑	Nazwa akcji

1. Identyfikatory typu akcji
    - 0: SQL do bazy systemowej

- 1: Eksport danych do Excel
- 2: Import danych z Excel
- 3: JavaScript
- 4: Odświeżanie list
- 5: Akcja systemowa
- 6: Wyślij e-mail
- 7: Raporty (Crystal Reports)
- 8: SQL zapisu danych
- 9: Zapytania do WebSerwisów
- 10: Procedury bazodanowe
- 12: Drukowanie ZPL
- 13: Konwersja Word do PDF
- 14: Szyfrowanie plików
- 15: Odbierz e-mail
- 16: SQL do baz zewnętrznych
- 18: Akcja uprawnień
- 19: Przypisywanie zadań
- 20: Generowanie PDF z Word
- 21: Rozpoznawanie danych OCR
- 22: Aktualizacja przez OCR
- 23: Magazyn Local Storage
- 24: Przypisz do interesariusza — użytkownicy
- 25: Raportowanie
- 26: Sprawdzenia przed zapisem
- 27: Wyświetlanie komunikatów
- 28: SQL między bazami danych
- 29: Otwieranie formularza
- 30: Generowanie elementów aplikacji
- 32: Tworzenie pliku tekstowego
- 33: Zmiana statusu dokumentów
- 34: Wysyłanie powiadomień
- 35: Otwieranie linku zewnętrznego
- 36: Inicjowanie przekierowania
- 37: Przekierowania HTTP
- 38: Algorytm C#
- 39: Podpis dokumentu (eSign)
- 40: Weryfikacja podpisu (eSign)

    - 41: Status zlecenia (eSign)

- 42: ePUAP utwórz dokument
- 43: ePUAP wyślij dokument
- 44: ePUAP podpisz dokument
- 45: ePUAP pobierz podpisany dokument
- 46: ePUAP weryfikuj podpisany dokument

- 42: ePUAP utwórz dokument (wycofana)
- 43: ePUAP wyślij dokument (wycofana)
- 44: ePUAP podpisz dokument (wycofana)
- 45: ePUAP pobierz podpisany dokument (wycofana)
- 46: ePUAP weryfikuj podpisany dokument (wycofana)

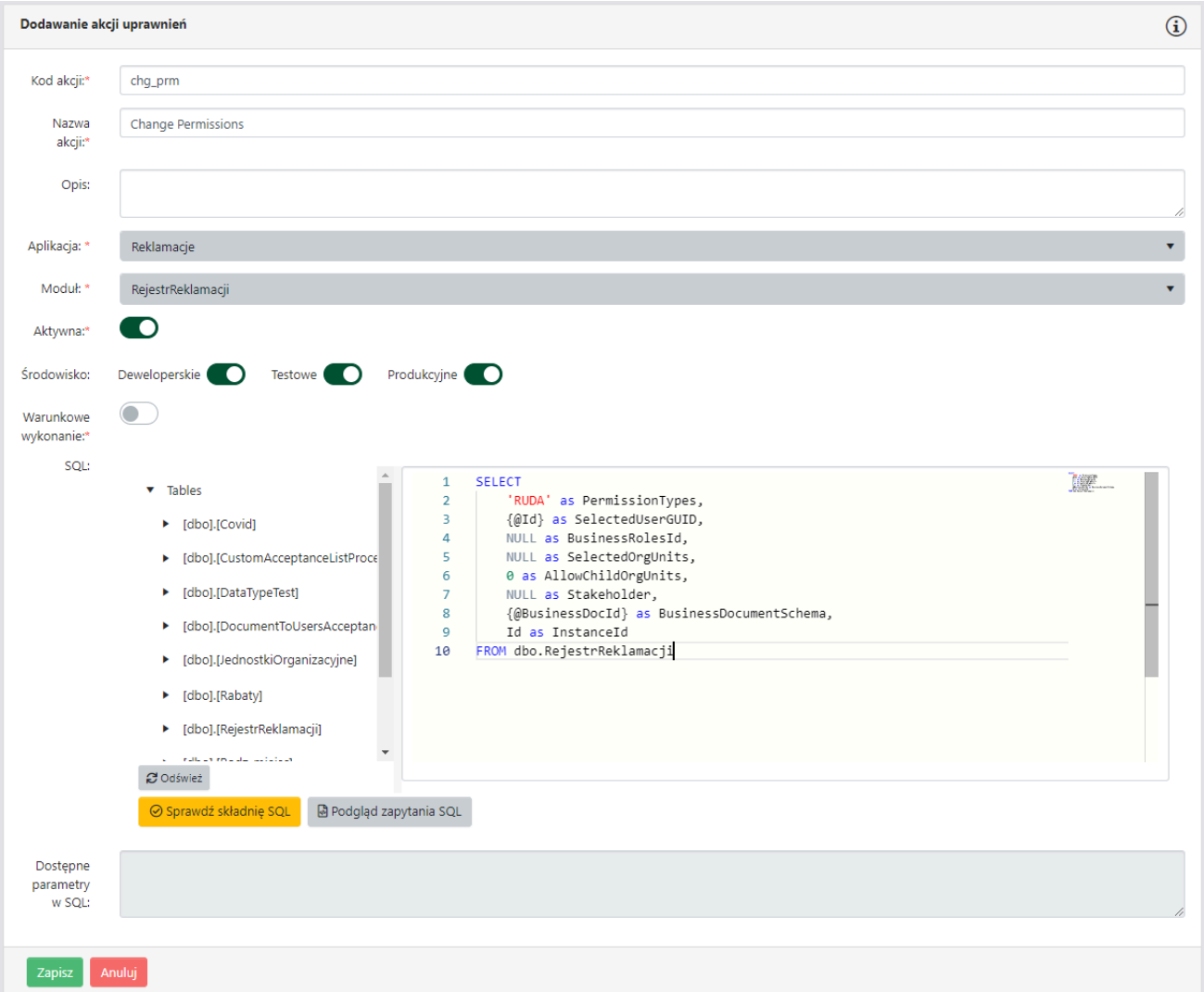
- 47: Skaner kodów QR/BC
- 48: Odświeżanie uprawnień
- 49: Otwieranie listy
- 50: Uruchom proces BPMN
- 51: Importowanie dokumentów
- 52: Przypisz do interesariusza — role w jednostkach
- 53: Przypisz do interesariusza — role globalne

1. Identyfikatory środowiska
    - 1: deweloperskie
    - 2: testowe
    - 4: produkcyjne

6.1.3. Uprawnienia do rekordów

Podsumowanie:
Akcja uprawnień pozwala nadawać uprawnienia ACL do istniejących dokumentów. W akcji można wskazać dowolnego użytkownika (użytkowników), role biznesowe lub jednostkę organizacyjną, a także wybrać określone dokumenty. Akcja uprawnień pozwala dynamicznie sterować uprawnieniami ACL w aplikacji *FrontSPA* w środowisku produkcyjnym w odniesieniu do istniejących dokumentów. Uprawnienia do nowych dokumentów są określane na podstawie szablonów zdefiniowanych przez konsultanta w aplikacji *AdminSPA*.

☛ *AdminSPA* > `NARZĘDZIA` > `Akcje` > `Uprawnienia do rekordów`



Akcja uprawnień

Oprócz właściwości wspólnych dla wszystkich akcji, w akcji uprawnień należy określić następujące właściwości:

`SQL`: zdefiniowane w akcji zapytanie SQL musi zwrócić jeden lub więcej rekordów w następujących kolumnach; w zapytaniu obsługiwana jest także operacja *UNION*:

- **PermissionTypes** (varchar): rodzaj nadawanego uprawnienia (RUDA: R=odczyt, U=modyfikacja, D=usuwanie, A=administracja).
- **SelectedUserGUID** (uniqueidentifier): identyfikator użytkownika, któremu zostanie przypisane uprawnienie (dozwolony NULL).
- **BusinessRolesId** (varchar): identyfikator roli, której zostaną przypisane uprawnienia (użyj składni *smart numbers*, dozwolony NULL).
- **SelectedOrgUnits** (varchar): identyfikator jednostki organizacyjnej, z której mogą pochodzić użytkownicy o wskazanych rolach (np. '1', '123' itp., dozwolony NULL); jeśli zapytanie zwraca dane w tej kolumnie, musi także zwrócić rolę w kolumnie *BusinessRolesId*. W takim przypadku uprawnienia zostaną nadane użytkownikom, którym przypisano określoną rolę w określonej jednostce organizacyjnej.

- **AllowChildOrgUnits** (bit): flaga dostępu użytkowników z podrzędnych jednostek organizacyjnych (przyjmuje wartości 0=NIE lub 1=TAK, dozwolony NULL).
- **Stakeholders** (varchar): Parametr nieobsługiwany (wersja 1.14.0.15).

Opcjonalnie zapytanie może zwrócić kolumny:

- **BusinessDocumentSchema** (int): identyfikator definicji dokumentu (np *15*, dozwolony NULL); w przypadku braku kolumny w zapytaniu lub wartości NULL uprawnienia dotyczą definicji dokumentu, do której należy bieżący dokument.
- **InstanceId** (int): identyfikator dokumentu (dozwolony NULL); w przypadku braku kolumny w zapytaniu lub wartości NULL uprawnienia dotyczą bieżącego dokumentu.

UWAGA
Kolumny muszą być zwracane w podanej kolejności; jeśli wartości w kolumnach typu *varchar* będą zwracane przez SQL, podstawiane za pomocą zmiennych *{@PoleFormularza}* lub przy użyciu mechanizmu *smart numbers*, konieczne będzie użycie funkcji SQL *CAST* do konwersji typu danych.

Jednostki organizacyjne mogą jedynie uzupełniać wybór użytkowników na podstawie ról, tak jak dotychczas w szablonach. Nie można przypisać uprawnień dla samej jednostki (można natomiast skonstruować zapytanie, które przypisze uprawnienia wszystkim użytkownikom należącym do określonej jednostki).

Zestaw wyników zwracanych przez zapytanie SQL może definiować wiele zestawów uprawnień. Zależnie od konfiguracji zapytania SQL może to być zestaw identycznych uprawnień dla wielu użytkowników, ról biznesowych, kategorii interesariuszy, definicji dokumentu biznesowego lub instancji dokumentu biznesowego. Natomiast przy użyciu operatora SQL *UNION* można jedną akcją utworzyć wiele różnych zestawów uprawnień.

PRZYKŁAD 1

Akcja nada następujące uprawnienia:

- *RUDA*: bieżąca instancja dokumentu biznesowego, użytkownik, który utworzył tę instancję

```
SELECT
     'RUDA' as PermissionTypes,
      {@CreatedBy} as SelectedUserGUID,
      NULL as BusinessRolesId,
      NULL as SelectedOrgUnits,
      0 as AllowChildOrgUnits,
```

```
    NULL as Stakeholder,
    {@BizDok} as BusinessDocumentSchema,
    {@Id} as InstanceId
```

PRZYKŁAD 2

Akcja nada bieżącemu uzytkownikowi uprawnienia *RUDA* do wszystkich instancji dokumentu biznesowego z tabeli *dbo.RejestReklamacji*

```
SELECT
    'RUDA' as PermissionTypes,
    {@UserId} as SelectedUserGUID,
    NULL as BusinessRolesId,
    NULL as SelectedOrgUnits,
    0 as AllowChildOrgUnits,
    NULL as Stakeholder,
    {@BusinessDocId} as BusinessDocumentSchema,
    Id as InstanceId
FROM dbo.RejestrReklamacji
```

PRZYKŁAD 3

Akcja nada uprawnienie *R* do wszystkich instancji dokumentu z tabeli *dbo.RejestrReklamacji* wszystkim użytkownikom z tabeli *core.UserProfiles*.

```
SELECT
    'R' as PermissionTypes,
    up.Id as SelectedUserGUID,
    NULL as BusinessRolesId,
    NULL as SelectedOrgUnits,
    0 as AllowChildOrgUnits,
    NULL as Stakeholder,
    {@BusinessDocId} as BusinessDocumentSchema,
    rr.Id as InstanceId
FROM dbo.RejestrReklamacji rr
CROSS JOIN core.UserProfiles up
```

PRZYKŁAD 4

Akcja nada uprawnienie *RU* do bieżącej instancji dokumentu dla każdego użytkownika należącego do jednostki organizacyjnej o podanym identyfikatorze.

```
SELECT
  'RU' as PermissionTypes,
  [UserProfileId] as SelectedUserGUID,
  NULL as BusinessRolesId,
  NULL as SelectedOrgUnits,
  0 as AllowChildOrgUnits,
  NULL as Stakeholder
FROM core.[OUInstanceUsers]
WHERE OUInstanceId = {@OrgUnitId}
```

PRZYKŁAD 5

Akcja nada uprawnienie *RUD* do bieżącej instancji dokumentu każdemu użytkownikowi z rolą *Kierownik* w jednostce organizacyjnej o podanym identyfikatorze.

```
SELECT
  'RUD' as PermissionTypes,
  NULL as SelectedUserGUID,
  {&WyjsciaPrywatne1.BusinessRoles.WyjsciaPrywatneModule.wyj_kier}
  as BusinessRolesId,
  {@OrgUnitId} as SelectedOrgUnits,
  0 as AllowChildOrgUnits,
  NULL as Stakeholder
```

Powiązane tematy:

- Wspólne parametry akcji
- Uprawnienia ACL do dokumentów
- Smart numbers

6.1.4. Przypisywanie użytkowników do grupy interesariuszy

Podsumowanie:
Akcja przypisywania użytkowników do grupy interesariuszy służy do dodawania/usuwania osób dla wybranej grupy interesariuszy oraz do usuwania wszystkich użytkowników z grupy interesariuszy. Akcja dotyczy bieżącego dokumentu (tego, w kontekście którego jest wykonywana) lub zestawu dokumentów należących do wskazanej definicji dokumentu.

☛ *AdminSPA* > `Narzędzia` > `Akcje` > `Przypisz do interesariusza` > `Użytkowników`

W celu zdefiniowania akcji interesariuszy należy określić następujące właściwości:

`SQL`: zapytanie SQL zwracające jeden lub więcej rekordów; zapytanie musi zwrócić trzy kolumny o podanych nazwach i w podanej kolejności (aby przypisać użytkowników do grupy interesariuszy dla określonych dokumentów, należy określić w zapytaniu dwie kolumny opcjonalne):

- *UserId* (uniqueidentifier): identyfikator użytkownika; wartość *NULL* w tej kolumnie i wartość *1* w kolumnie *IsRemoval* spowodują usunięcie wszystkich użytkowników z listy.
- *StakeholderId* (int): identyfikator grupy interesariuszy z tabeli *[core].[Stakeholders]*; o ile wartość nie jest pobierana zapytaniem, najlepiej użyć mechanizmu *smart numbers*, aby umożliwić prawidłową migrację.

  UWAGA
  Wskazana grupa interesariuszy musi należeć do definicji dokumentu odpowiedającego bieżącemu dokumentowi lub do definicji wskazanej w kolumnie *BusinessDocId*, w przeciwnym razie wykonanie akcji nie powiedzie się.

- *IsRemoval* (bit): typ operacji wykonywanej na liście interesariuszy; możliwe wartości:
  - *0*: dodanie użytkownika do listy interesariuszy.
  - *1*: usunięcie użytkownika z listy interesariuszy; jeśli kolumna *UserId* zwraca wartość *NULL*, zostaną usunięci wszyscy użytkownicy z listy.
- *BusinessDocId* (int): kolumna opcjonalna, identyfikator definicji dokumentu.
- *InstanceId* (int): kolumna opcjonalna, identyfikator dokumentu.

UWAGA

- W zapytaniu można użyć operatora *UNION*.
- Kolumny *BusinessDocId* i *InstanceId* umożliwiają określenie definicji dokumentu oraz zestawu dokumentów należących do tej definicji w celu zbiorczego przypisania użytkownika do grupy interesariuszy dla tej definicji. Jeśli te kolumny nie zostaną określone lub w obu zostanie zwrócona wartość *null*, użytkownik zostanie przypisany do grupy interesariuszy dla bieżącego dokumentu.
- Jeśli jedna z kolumn opcjonalnych zwróci identyfikator, a druga wartość *null*, wykonanie akcji nie powiedzie się i zostanie wyświetlony komunikat o błędzie.

PRZYKŁADY

Akcja skopiuje **wszystkich** użytkowników z grupy interesariuszy o kodzie *AcceptorInProcess* do grupy o kodzie *DefAcceptor*.

```
SELECT UserProfileId AS UserId,
  {&CstAccept.Stakeholders.CstAcceptDoc.AcceptorInProcess}
  AS StakeholderId,
  CAST(0 AS BIT) AS IsRemoval
FROM core.ACLStakeholderEntries AS ase
WHERE ase.StakeholderId = {&CstAccept.Stakeholders.CstAcceptDoc.DefAcceptor}
```

Akcja przypisze bieżącego użytkownika do grupy interesariuszy o kodzie *AcceptorInProcess* oraz usunie użytkowników zwróconych przez drugie zapytanie z grupy interesariuszy o kodzie *DefAcceptor*:

```
SELECT
  {@_UserId} AS UserId,
  {&CstAccept.Stakeholders.CstAcceptDoc.AcceptorInProcess}
  AS StakeholderId,
  CAST(0 AS BIT) AS IsRemoval
UNION
SELECT
  Id AS UserId,
  {&CstAccept.Stakeholders.CstAcceptDoc.DefAcceptor}
  AS StakeholderId,
  CAST(1 AS BIT) AS IsRemoval
FROM core.UserProfiles
```

```
WHERE LEFT(FirstName,1) IN ('a', 's')
```

Akcja przypisze do określonej grupy interesariuszy wszystkie osoby wybrane w kontrolce typu lista wielowyboru. W zapytaniu użyto funkcji *STRING_SPLIT* do wydzielenia poszczególnych identyfikatorów użytkowników. Warunkiem poprawnego działania zapytania jest wybranie co najmniej dwóch osób na liście.

```
SELECT
VALUE as UserId,
{&CstAccept.Stakeholders.CstAcceptDoc.DefAcceptor}
    AS StakeholderId,
CAST(0 AS BIT) AS IsRemoval
FROM STRING_SPLIT({@AcceptorsList}, '$')
WHERE value <> ','
```

Akcja przypisze zalogowanego użytkownika do grupy interesariuszy dla wszystkich dokumentów należących do wskazanej definicji dokumentu.

```
SELECT
  {@_UserId} AS UserId,
  {&CstAccept.Stakeholders.CstAcceptDoc.AcceptorInProcess}
  AS StakeholderId,
  CAST(0 AS BIT) AS IsRemoval
  (SELECT Id FROM core.BusinessDocuments
  WHERE [Table] = 'dbo.Reklamacje')
  AS BusinessDocId,
  Id AS InstanceId
FROM dbo.Reklamacje
```

6.1.5. Przypisywanie ról globalnych do interesariuszy

Podsumowanie:
Ta akcja służy do:

- dodawania ról globalnych do grupy interesariuszy,
- usuwania ról globalnych z grupy interesariuszy,
- usuwania wszystkich ról globalnych z grupy interesariuszy.

Akcja zmienia skład grupy interesariuszy dla:

- bieżącego dokumentu (tego, w kontekście którego jest wykonywana)
- zbioru dokumentów w ramach wskazanej definicji

☛ *AdminSPA* > `NARZĘDZIA` > `Akcje` > `Przypisz do interesariusza` > `Role w jednostkach`

W celu zdefiniowania akcji przypisywania ról globalnych do interesariuszy należy określić następujące właściwości:

`SQL`: zapytanie SQL zwracające jeden lub więcej rekordów; zapytanie musi zwrócić pięć kolumn o podanych nazwach i w podanej kolejności (aby zmienić interesariuszy dla określonych dokumentów we wskazanej definicji dokumentu, należy określić w zapytaniu dwie kolumny opcjonalne):

- *BusinessDocId* (int): kolumna opcjonalna, identyfikator definicji dokumentu.
- *RecordId* (int): kolumna opcjonalna, identyfikator dokumentu we wskazanej definicji dokumentu.
- *StakeholderId* (int): identyfikator grupy interesariuszy z tabeli *[core].[Stakeholders]*; o ile wartość nie jest pobierana zapytaniem, najlepiej użyć mechanizmu *smart numbers*, aby umożliwić prawidłową migrację.

  UWAGA
  Wskazana kategoria interesariuszy musi należeć do definicji dokumentu odpowiadającego bieżącemu dokumentowi lub do definicji wskazanej w kolumnie *BusinessDocId*, w przeciwnym razie wykonanie akcji nie powiedzie się.

- *Operation* (int): typ operacji wykonywanej na grupie interesariuszy; możliwe wartości:
  - *0*: dodanie roli globalnej do grupy interesariuszy.
  - *1*: usunięcie roli globlanej z grupy interesariuszy;

- *GlobalRoleId:* (int): identyfikator roli biznesowej; wymagane, oprócz sytuacji opisanej poniżej.
  Ustawienie parametrów *GlobalRoleId = null* oraz *Operation = 1* spowoduje usunięcie wszystkich ról globalnych z grupy interesariuszy dla wskazanego dokumentu lub bieżącego dokumentu.

UWAGA

- W zapytaniu można użyć operatora UNION.
- Kolumny *BusinessDocId* i *RecordId* umożliwiają określenie definicji dokumentu oraz zestawu dokumentów należących do tej definicji w celu zbiorczego przypisania ról globalnych do grupy interesariuszy dla tej definicji. Jeśli te kolumny nie zostaną określone lub w obu zostanie zwrócona wartość null, przypisanie ról globalnych do grupy interesariuszy nastąpi w odniesieniu do bieżącego dokumentu.

Przykłady:

Akcja usunie **wszystkie** role globalne dla interesariusza *Laborant* dla bieżącego dokumentu.

```
SELECT
NULL AS BusinessDocId,
NULL AS RecordId,
  {&Uprawnienia.Stakeholders.uprUpr.Laborant} AS StakeholderId,
  1 as Operation,
 NULL AS GlobalRoleId
```

Akcja dodaje role globalną *KierownikU* do grupy interesariuszy *Laborant* dla bieżącego dokumentu:

```
SELECT
NULL AS BusinessDocId,
NULL AS RecordId,
  {&Uprawnienia.Stakeholders.uprUpr.Laborant} AS StakeholderId,
  0 AS Operation,
  {&Uprawnienia.BusinessRoles.Uprawnienia.KierownikU} AS GlobalRoleId
```

Akcja usuwa **wszystkie** role globalne z grupy interesariuszy *Akceptant* dla bieżącego dokumentu, następnie dla dokumentów o id > 4000 (zbiór dokumentów w ramach wskazanej definicji dokumentu za pomocą warunku *WHERE*) dodaje rolę globalną *PracownikU* do grupy interesariuszy *Akceptant*.

```
SELECT
NULL AS BusinessDocId,
NULL AS RecordId,
  {&Uprawnienia.Stakeholders.uprUpr.Akceptant} AS StakeholderId,
  1 AS Operation,
  NULL AS GlobalRoleId

UNION ALL

SELECT
{@BusinessDocId} AS BusinessDocId,
{@Id} AS RecordId,
  {&Uprawnienia.Stakeholders.uprUpr.Akceptant} AS StakeholderId,
  0 AS Operation, {&Uprawnienia.BusinessRoles.Uprawnienia.PracownikU} AS GlobalRoleId
FROM [dbo].[AR_Uprawnienia]
WHERE id >4000
```

6.1.6. Przypisywanie ról w jednostkach organizacyjnych do interesariuszy

Podsumowanie:
Akcja przypisywania ról w jednostkach organizacyjnych do interesariuszy służy do:

- dodawania i usuwania ról w jednostkach organizacyjnych do grupy interesariuszy,
- usuwania wszystkich ról w jednostkach organizacyjnych z grupy interesariuszy.

Akcja zmienia skład grupy interesariuszy dla:

- bieżącego dokumentu (tego, w kontekście którego jest wykonywana),
- zbioru dokumentów w ramach wskazanej definicji.

☛ *AdminSPA* > `Narzędzia` > `Akcje` > `Przypisz do interesariusza` > `Role w jednostkach`

Określ następujące właściwości:

`SQL`: zapytanie SQL zwracające jeden lub więcej rekordów; zapytanie musi zwrócić sześć kolumn o podanych nazwach i w podanej kolejności (aby zmienić interesariuszy dla określonych dokumentów we wskazanej definicji dokumentu, należy określić w zapytaniu dwie kolumny opcjonalne):

- *BusinessDocId* (int): kolumna opcjonalna, identyfikator definicji dokumentu.
- *RecordId* (int): kolumna opcjonalna, identyfikator dokumentu we wskazanej definicji dokumentu.

- *StakeholderId* (int): identyfikator grupy interesariuszy z tabeli *[core].[Stakeholders]*; o ile wartość nie jest pobierana zapytaniem, najlepiej użyć mechanizmu *smart numbers*, aby umożliwić prawidłową migrację.

  > **UWAGA**
  > Wskazana kategoria interesariuszy musi należeć do definicji dokumentu odpowiadającego bieżącemu dokumentowi lub do definicji wskazanej w kolumnie *BusinessDocId*, w przeciwnym razie wykonanie akcji nie powiedzie się.

- *Operation* (int): typ operacji wykonywanej na grupie interesariuszy; możliwe wartości:
  - *0*: dodanie roli globalnej do grupy interesariuszy.
  - *1*: usunięcie roli globlanej z grupy interesariuszy;
- *OUId* (int): identyfikator jednostki organizacyjnej (można użyć funkcji *core.calculateOUId*, która oblicza ten identyfikator na podstawie kombinacji podanych parametrów — kodu jednostki odniesienia, identyfikatora użytkownika, poziomu struktury względem jednostki odniesienia).
- *RoleId* (int): identyfikator roli biznesowej; wymagane, oprócz sytuacji opisanej poniżej.
  Ustawienie parametrów *OUId/RoleId = null* oraz *Operation = 1* spowoduje usunięcie wszystkich ról w jednostkach z grupy interesariuszy dla wskazanego dokumentu lub bieżącego dokumentu.

> **UWAGA**
> - W zapytaniu można użyć operatora UNION.
> - Kolumny *BusinessDocId* i *RecordId* umożliwiają określenie definicji dokumentu oraz zestawu dokumentów należących do tej definicji w celu zbiorczego przypisania ról w jednostkach do grupy interesariuszy dla tej definicji. Jeśli te kolumny nie zostaną określone lub w obu zostanie zwrócona wartość null, przypisanie ról w jednostkach do grupy interesariuszy nastąpi w odniesieniu do bieżącego dokumentu.

PRZYKŁADY

Akcja usuwa **wszystkie** role w jednostkach z grupy interesariuszy *Rzeczoznawcy* dla bieżącego dokumentu.

```
SELECT
NULL AS BusinessDocId,
NULL AS RecordId,
  {&Uprawnienia.Stakeholders.uprUpr.Rzeczoznawcy} AS StakeholderId,
  1 as Operation,
 NULL AS OUId,
 NULL AS RoleId
```

Akcja dodaje rolę *Referent* we wskazanej jednostce do grupy interesariuszy *Rzeczoznawcy* dla bieżącego dokumentu:

```
SELECT
NULL AS BusinessDocId,
NULL AS RecordId,
  {&Uprawnienia.Stakeholders.uprUpr.Rzeczoznawcy} AS StakeholderId,
  0 AS Operation,
  10 AS OUId,
  {&Uprawnienia.BusinessRoles.Uprawnienia.Referent} AS RoleId
```

Akcja usuwa **wszystkie** role w jednostkach organizacyjnych z grupy interesariuszy *Rzeczoznawcy* dla bieżącego dokumentu oraz dodaje rolę *Kierownik* w we wskazanej jednostce do grupy interesariuszy *Akceptanci* dla dokumentów o id > 4000 (zbiór dokumentów w ramach wskazanej definicji dokumentu za pomocą warunku *WHERE*).

```
SELECT
NULL AS BusinessDocId,
NULL AS RecordId,
  {&Uprawnienia.Stakeholders.uprUpr.Rzeczoznawcy} AS StakeholderId,
  1 AS Operation,
  NULL AS OUId,
  NULL AS RoleId

UNION ALL

SELECT
{@BusinessDocId} AS BusinessDocId,
{@Id} AS RecordId,
  {&Uprawnienia.Stakeholders.uprUpr.Akceptanci} AS StakeholderId,
  0 AS Operation,
  10 AS OUId,
  {&Uprawnienia.BusinessRoles.Uprawnienia.Kierownik} AS RoleId
```

```
FROM [dbo].[AR_Uprawnienia]
WHERE id >4000
```

6.1.7. Odświeżanie uprawnień

Podsumowanie:
Akcja odświeżania uprawnień służy do aktualizacji uprawnień po zmianie interesariusza. Odświeżanie odbywa się w momencie wykonania akcji i nie wymaga przejścia do kolejnego statusu. Akcja odświeża uprawnienia zgodnie z konfiguracją uprawnień dla statusów i uprawnień dla instancji. Można jej używać dla uprawnień w modelu uproszczonym.

☛ `NARZĘDZIA` > `Akcje` > `Odświeżanie uprawnień`

Edycja akcji

Kod akcji:* RefreshPermissions

Nazwa akcji:* RefreshPermissions

Opis: RefreshPermissions

Aplikacja:* CustomAcceptance

Moduł:* CustomAcceptanceList

Aktywna:*

Środowisko: Deweloperskie Testowe Produkcyjne

Warunkowe wykonanie:*

Wykonaj dla bieżącej instancji dokumentu:*

Wybór instancji dokumentu (SQL):*

Tables
- [dbo].[Covid]
- [dbo].[CustomAccept
- [dbo].[DataTypeTest]
- [dbo].[DocumentToU
- [dbo].[JednostkiOrga
- [dbo].[Jira]
- [dbo].[KatalogUsun]
- [dbo].[Rozliczenia]

Odśwież

```
1   SELECT Id
2       , (SELECT Id FROM core.BusinessDocuments
3          WHERE [Table] = 'dbo.Reklamacje')
4   FROM dbo.Reklamacje
5   WHERE Status = {&Reklamacje.BusinessStatuses.rklRjs.rk
6
7   UNION
8
9   SELECT Id
10      , (SELECT Id FROM core.BusinessDocuments
11         WHERE [Table] = 'dbo.Zwroty')
12  FROM dbo.Zwroty
```

Sprawdź składnię SQL | Maksymalizuj edytor

Zapytanie musi zwrócić dwie kolumny:
recordId (int) - id instancji dokumentu
businessDocumentId (int) - id definicji dokumentu biznesowego do którego należy dana instancja
Przykład:

```
SELECT Id, (SELECT Id From core.BusinessDocuments Where [Table] = 'dbo.Reklamacje') FROM dbo.Reklamacje WHERE Status = {@Stat
UNION
SELECT Id, 13 FROM dbo.Zwroty
```

Dostępne parametry w SQL:

Miejsca użycia: Pola w formularzach:

interesariusze [CustomAcceptance-formularz dla dokumentu CustomAcceptance] (CustomAcceptance-form — Edytuj

usuń JT, odśwież ACL [CustomAcceptance-formularz dla dokumentu CustomAcceptance] (CustomAcceptance-form — Edytuj

Przejścia statusów:

Zadanie ręczne: (Lista akceptantów ustalona) -> [Rozpocznij proces decyzyjny] -> Zadanie ręczne: (Oczekiwanie na decyzje — Edytuj

Zapisz | Anuluj

Akcja odświeżania uprawnień

Oprócz właściwości wspólnych dla wszystkich akcji, w akcji odświeżania uprawnień należy określić następujące właściwości:

`Wykonaj dla bieżącej instancji dokumentu`: (domyślnie włączony) akcja odświeży uprawnienia tylko dla bieżącej instancji dokumentu.

`Wybór instancji dokumentu (SQL)`: zdefiniowane w akcji zapytanie SQL musi zwrócić dwie kolumny, które będą zawierały identyfikatory instancji dokumentów (kolumna *Id* z tabeli dokumentu) oraz z identyfikator definicji dokumentu (kolumna *Id* w tabeli *core.BusinessDocuments*). W zapytaniu można użyć operatora *UNION*. Tego trybu można użyć w przypadku wsadowej operacji odświeżania uprawnień dla wielu dokumentów.

WAŻNE
Wyniki zapytania SQL są używane jako parametry w zapytaniu aktualizującym uprawnienia. Ze względu na ograniczenia SQL Server ich maksymalna dozwolona liczba wynosi 2100. Jeśli zapytanie zwróci więcej identyfikatorów dokumentów, akcja zakończy się błędem.

PRZYKŁAD

Poniżej podano przykład zapytania dla trybu wsadowego akcji odświeżania uprawnień. Akcja zaktualizuje uprawnienia dla dokumentów z tabeli *dbo.Reklamacje* w określonym statusie oraz dla wszystkich dokumentów z tabeli *dbo.Zwroty*.

```
SELECT Id
    , (SELECT Id FROM core.BusinessDocuments
       WHERE [Table] = 'dbo.Reklamacje')
FROM dbo.Reklamacje
WHERE Status = {&Reklamacje.BusinessStatuses.rklRjs.rklRjs_utworzony}

UNION

SELECT Id
     , (SELECT Id FROM core.BusinessDocuments
        WHERE [Table] = 'dbo.Zwroty')
FROM dbo.Zwroty
```

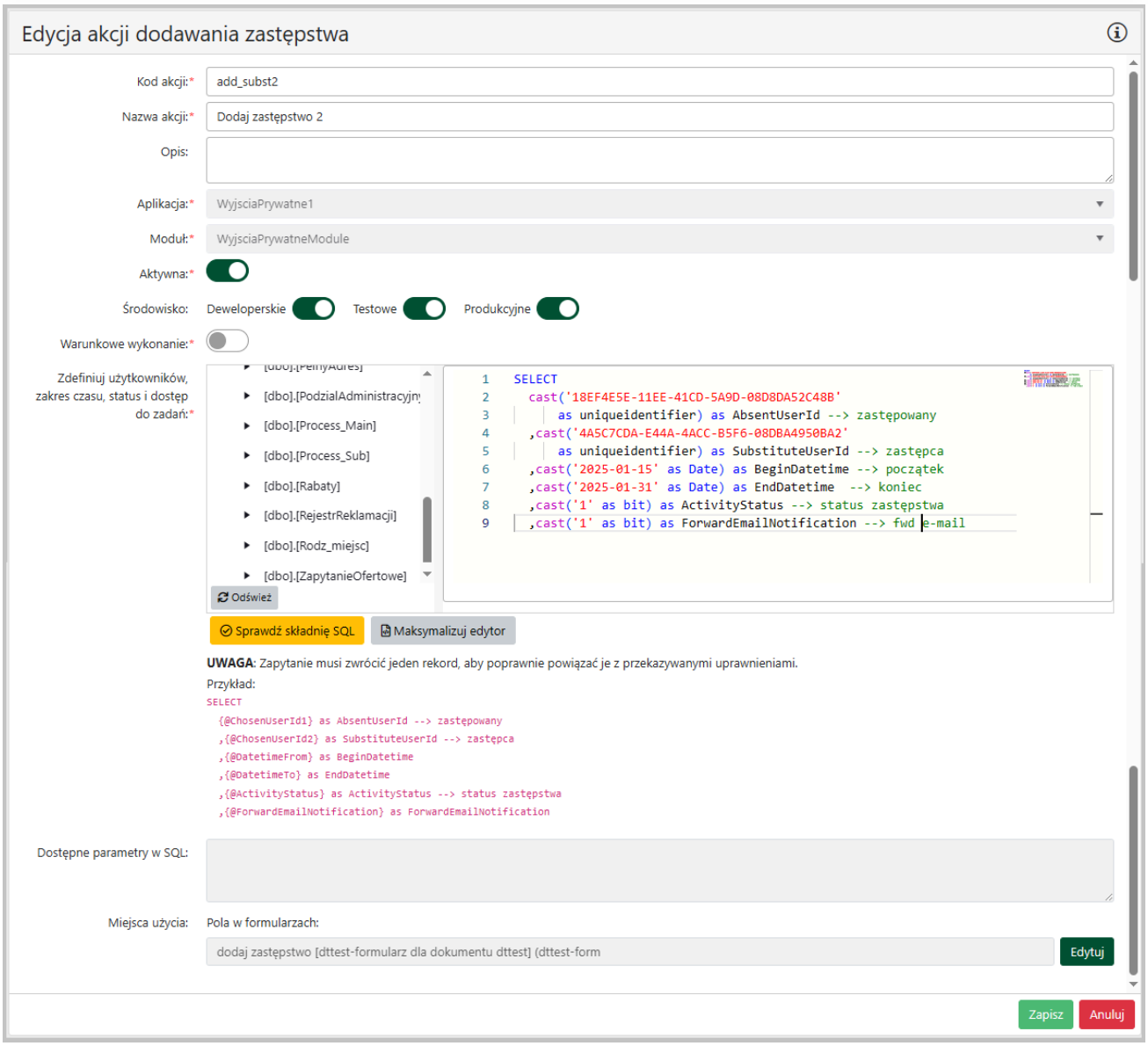
Powiązane tematy:

- Wspólne parametry akcji
- Uprawnienia ACL do dokumentów
- Przypisywanie użytkowników do grupy interesariuszy
- Smart numbers

6.1.8. Dodaj zastępstwo

Podsumowanie:
Akcja dodaje nowe zastępstwo na podstawie podanych danych, to jest dodaje rekord do tabeli *core.PermissionDelegations*. Akcja jest przeznaczona do wykorzystania w funkcjonalnościach obsługi zastępstw budowanych samodzielnie przez konsultanta.

☛ *AdminSPA* > `Narzędzia` > `Akcje` > `Zastępstwa` > `Dodaj zastępstwo`



Akcja dodawania zastępstwa

Oprócz właściwości wspólnych dla wszystkich akcji, w akcji dodawania zastępstwa należy określić zapytanie SQL zwracające ogólne atrybuty zastępstwa; to zapytanie musi zwrócić tylko jeden rekord danych w następujących kolumnach:

- *AbsentUserId* (*uniqueidentifier*): identyfikator użytkownika zastępowanego,

- *SubstituteUserId* (*uniqueidentifier*): identyfikator użytkownika zastępującego; musi być różny od *AbsentUserId*,
- *BeginDatetime* (*datetime*): data rozpoczęcia zastępstwa; jeśli zostanie podana godzina, zostanie zamieniona na *00:00:00.0000000*; musi być wcześniejsza niż data *EndDatetime*,
- *EndDatetime* (*datetime*): data zakończenia zastępstwa; jeśli zostanie podana godzina, zostanie zamieniona na *23:59:59.0000000*; musi być późniejsza niż data *BeginDatetime*,
- *ActivityStatus* (*bit*): flaga określająca, czy zastępstwo jest aktywne, czy nie; dozwolone wartości *0* lub *1*,
- *ForwardEmailNotification* (*bit*): flaga określająca, czy e-maile wysyłane z nAxiom do zastępowanego mają być wysyłane również do zastępującego; dozwolone wartości *0* lub *1*,

Wystąpienie błędów walidacji podczas wykonywania akcji powoduje wyświetlenie komunikatu o błędzie i przerwanie wykonywania akcji.

Powiązane tematy:

- Wspólne parametry akcji
- core.SubstitutionsPerformedByUser
- core.GetAllSubstitutions

6.1.9. Edytuj zastępstwo

Podsumowanie:
Akcja umożliwia zmianę parametrów zdefiniowanego zastępstwa, to jest aktualizuje rekord w tabeli *core.PermissionDelegations*. Akcja wymaga podania wszystkich parametrów wymaganych w definicji zastępstwa, nawet tych, których wartości pozostają bez zmian. Akcja jest przeznaczona do wykorzystania w funkcjonalnościach obsługi zastępstw budowanych samodzielnie przez konsultanta.

☛ *AdminSPA* > `Narzędzia` > `Akcje` > `Zastępstwa` > `Edytuj zastępstwo`

Oprócz właściwości wspólnych dla wszystkich akcji, w akcji edycji zastępstwa należy określić zapytanie SQL zwracające ogólne atrybuty zastępstwa; to zapytanie musi zwrócić tylko jeden rekord danych w następujących kolumnach:

- *AbsentUserId* (*uniqueidentifier*): identyfikator użytkownika zastępowanego,
- *SubstituteUserId* (*uniqueidentifier*): identyfikator użytkownika zastępującego; musi być różny od *AbsentUserId*,

- *BeginDatetime* (*datetime*): data rozpoczęcia zastępstwa; jeśli zostanie podana godzina, zostanie zamieniona na *00:00:00.0000000*; musi być wcześniejsza niż data *EndDatetime*,
- *EndDatetime* (*datetime*): data zakończenia zastępstwa; jeśli zostanie podana godzina, zostanie zamieniona na *23:59:59.0000000*; musi być późniejsza niż data *BeginDatetime*,
- *ActivityStatus* (*bit*): flaga określająca, czy zastępstwo jest aktywne, czy nie; dozwolone wartości *0* lub *1*,
- *ForwardEmailNotification* (*bit*): flaga określająca, czy e-maile wysyłane z nAxiom do zastępowanego mają być wysyłane również do zastępującego; dozwolone wartości *0* lub *1*,

Wystąpienie błędów walidacji podczas wykonywania akcji powoduje wyświetlenie komunikatu o błędzie i przerwanie wykonywania akcji.

Powiązane tematy:

- Wspólne parametry akcji
- core.SubstitutionsPerformedByUser
- core.GetAllSubstitutions

6.1.10. Usuń zastępstwo

Podsumowanie:
Akcja umożliwia usunięcie zdefiniowanego zastępstwa, to jest usuwa rekord z tabeli *core.PermissionDelegations*. Akcja wymaga podania jednego lub kilku identyfikatorów zastępstw do usunięcia. Akcja jest przeznaczona do wykorzystania w funkcjonalnościach obsługi zastępstw budowanych samodzielnie przez konsultanta.

☛ *AdminSPA* > `Narzędzia` > `Akcje` > `Zastępstwa` > `Usuń zastępstwo`

Oprócz właściwości wspólnych dla wszystkich akcji, w akcji usunięcia zastępstwa należy określić zapytanie SQL zwracające jeden lub kilka rekordów w kolumnie:

- *SubstitutionId* (*int*): identyfikator zastępstwa, którego dotyczy akcja.

Wystąpienie błędu walidacji podczas wykonywania akcji powoduje wyświetlenie komunikatu o błędzie i przerwanie wykonywania akcji.

Powiązane tematy:

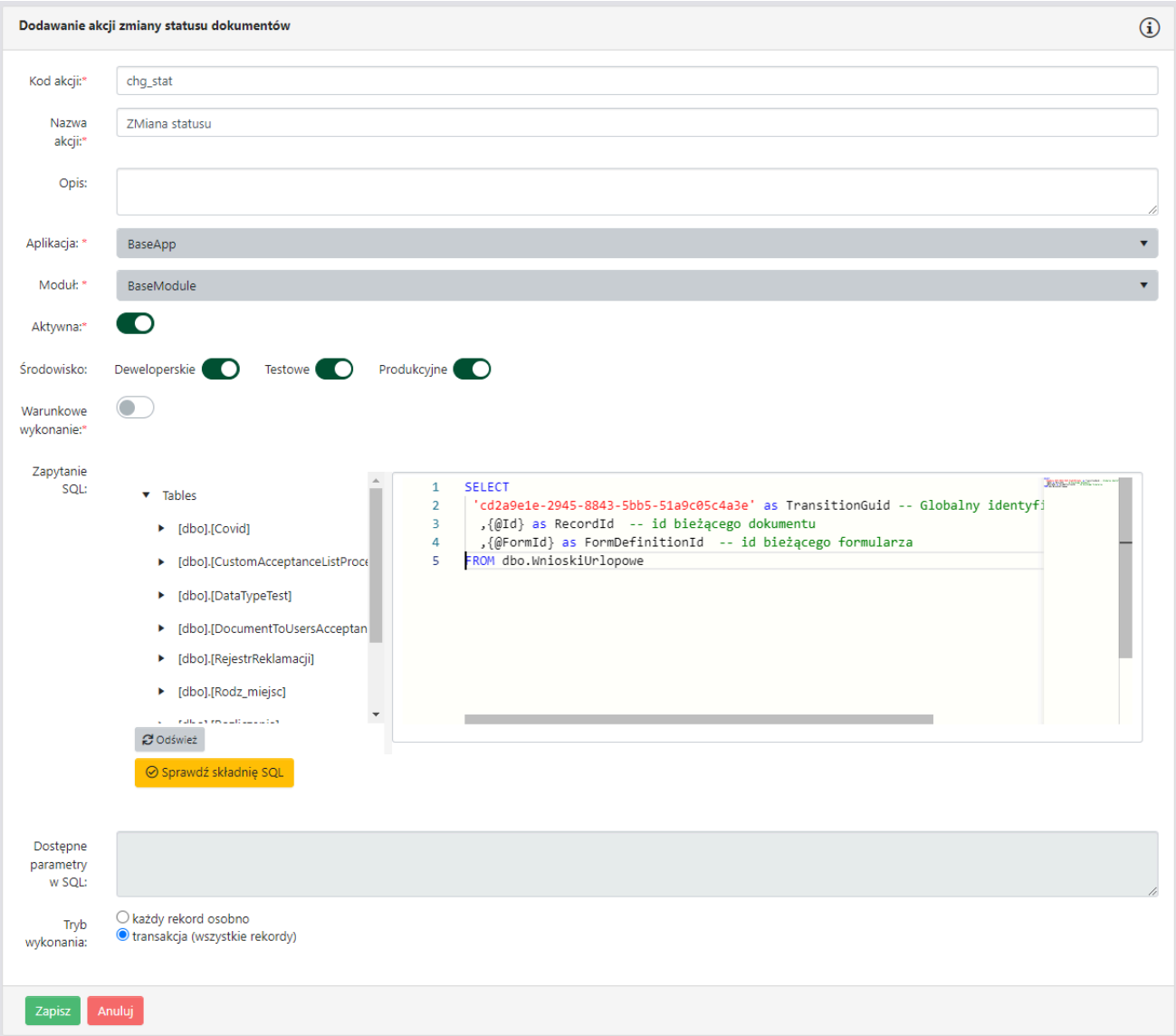
- Wspólne parametry akcji
- core.SubstitutionsPerformedByUser
- core.GetAllSubstitutions

6.1.11. Zmiana statusu dokumentów

Podsumowanie:
Akcja zmiany statusu dokumentów umożliwia zautomatyzowane wykonanie wskazanego przejścia diagramu procesu dla zestawu dokumentów biznesowych wybranych zapytaniem SQL. Akcja wykonując przejście, uwzględnia wszystkie jego elementy, w tym akcje wykonywane przed i po zmianie statusu oraz zmianę statusu. Funkcjonalnie akcja zmiany statusu jest tożsama z kliknięciem przycisku reprezentującego dane przejście dla pojedynczego dokumentu.

Akcja może być wykonywana w dwóch trybach: indywidualnie dla poszczególnych dokumentów albo transakcyjnie dla całej partii zwróconej przez zapytanie.

☛ *AdminSPA* > `NARZĘDZIA` > `Akcje` > `Zmiana statusu dokumentów`



Akcja zmiany statusu dokumentów

Oprócz właściwości wspólnych dla wszystkich akcji, w akcji zmiany statusu należy określić następujące właściwości:

`Zapytanie SQL`: zapytanie SQL zwracające zestaw dokumentów do zmiany statusu; musi zwrócić następujące kolumny (kolejność nie ma znaczenia):

- `TransitionId/TransitionName/TransitionGuid`: identyfikator, nazwa lub globalny identyfikator przejścia, które ma zostać wykonane dla wybranych dokumentów.
- `RecordId`: identyfikator dokumentu biznesowego z tabeli dokumentu biznesowego.
- `FormDefinitionId`: identyfikator definicji formularza powiązanego z tabelą dokumentu biznesowego.

`Tryb wykonania`:

- `każdy rekord osobno`: rekordy są przetwarzane indywidualnie, każdy w osobnej transakcji; błąd dla indywidualnego dokumentu powoduje przerwanie działania akcji — kolejne rekordy nie zostaną przetworzone; jeśli błąd dla dokumentu wystąpi w akcji po, zmiana statusu tego dokumentu nie zostanie wycofana; w pozostałych przypadkach status dokumentu nie zostanie zmieniony.
- `transakcja (wszystkie rekordy)`: rekordy są przetwarzane w ramach jednej transakcji; błąd dla indywidualnego dokumentu (akcja przed, zmiana statusu lub akcja po) powoduje wycofanie wszelkich zmian i przerwanie wykonywania akcji. Jest to zachowanie odmienne od wykonywania przejść poza tą akcją, kiedy to błędy akcji po nie powodują wycofania zmiany statusu.

> **UWAGA**
> Należy wziąć pod uwagę, że w przypadku akcji uruchamianej w trybie transakcyjnym wpisy uprawnień ACL są aktualizowane zbiorczo po zakończeniu transakcji. W rzadkich przypadkach może to powodować błędy uprawnień w akcjach po (np. w akcji przypisania właściciela zadania). W takiej sytuacji należy zmienić tryb wykonywania akcji na `każdy rekord osobno`.

> **OSTRZEŻENIE**
> Od wersji nAxiom 1.13.2.0 wprowadzono zmiany optymalizujące wydajność akcji zmiany statusu dla wielu dokumentów. W wyniku tych zmian podczas wykonywania przejść w tej akcji nie będą dostępne parametry systemowe dodawane do kontekstu formularza: *@FormId*, *@BusinessDocId*, *@FormCode* i *@AppId*. Dotyczy to akcji wykonywanych na przejściu, warunków wykonania przejścia, warunku wyświetlania przycisku, eskalatorów itp. Parametry *@FormId* i *@BusinessDocId* można zastąpić ich systemowymi odpowiednikami: *@_FormId* i *@_BusinessDocDefId*.

Przykład:

Przykład zapytania SQL dla akcji zmiany statusu, która odrzuci wszystkie wnioski o urlop z datą rozpoczęcia w lipcu lub sierpniu.

```
SELECT 'Odrzuć wniosek' as TransitionName -- nazwa przejścia
      ,Id as RecordId                     -- id dokumentu
      ,20 as FormDefinitionId             -- id formularza
FROM dbo.WnioskiUrlopowe
WHERE MONTH(StartDate) in (7,8)
```

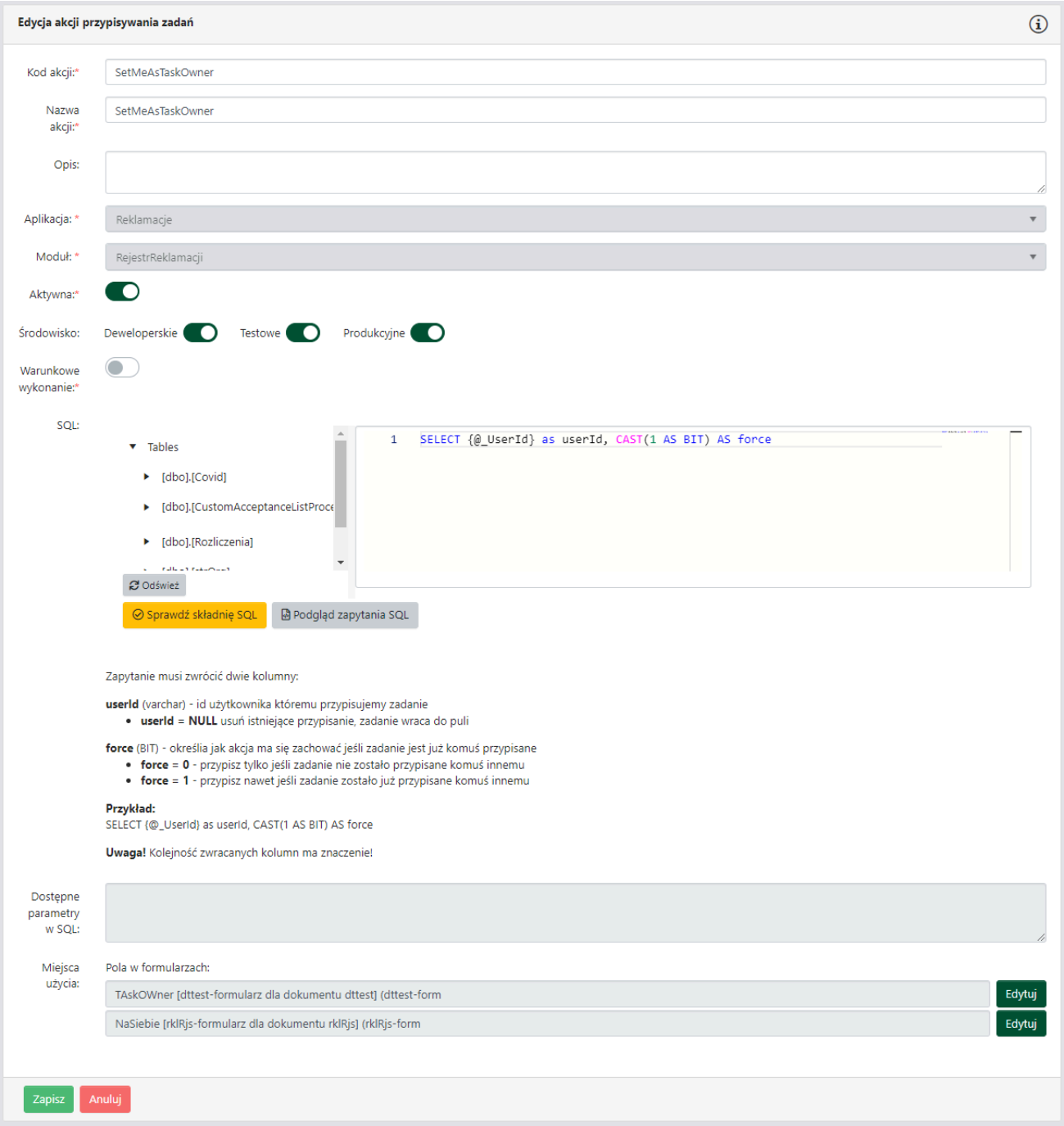
Powiązane tematy:

- Wspólne parametry akcji

6.1.12. Przypisywanie zadań

Podsumowanie:
Akcje przypisywania zadań pozwalają wyznaczać konkretnego użytkownika do wykonania czynności dotyczących dokumentu w określonym statusie (zadanie). Akcja wymaga zdefiniowania odpowiedniego zapytania SQL, które musi zwrócić identyfikator użytkownika oraz wartość parametru określającego, czy bieżące przypisanie ma nadpisywać wcześniejsze przypisanie.

☛ *AdminSPA* > `NARZĘDZIA` > `Akcje` > `Przypisywanie zadań`



Akcja przypisywania zadań

Oprócz właściwości wspólnych dla wszystkich akcji, w akcji przypisywania zadań należy określić następujące właściwości:

`SQL`: zapytanie SQL zwracające dwie kolumny:

- `userId` (varchar): identyfikator użytkownika, któremu przypisujemy zadanie; wartość *NULL* usuwa istniejące przypisanie, zadanie wraca do puli zadań nieprzypisanych.
- `force` (bit): określa zachowanie akcji w przypadku, gdy do zadania już przypisano użytkownika:

- `0`: przypisuje zadanie, tylko wtedy, gdy nie jest przypisane.
- `1`: przypisuje zadanie nawet wtedy, gdy jest już przypisane.

> **UWAGA**
> Kolejność zwracanych kolumn musi być taka jak podana na liście

Informacje o właścicielach zadań są zapisywane jako wiersze w systemowej tabeli *[core].[TaskOwnerships]*. Ta tabela składa się następujących kolumn:

Nazwa kolumny	Typ danych	Zezwól NULL	Opis
Id	int	☐	Kolumna klucza głównego, identyfikator rekordu w tabeli
StatusId	int	☐	Identyfikator statusu, dla którego wyznaczono właściciela zadania; odpowiada kolumnie *Status* w tabeli dokumentu biznesowego
UserId	uniqueidentifier	☐	Identyfikator użytkownika wyznaczonego jako właściciel zadania; odpowiada kolumnie *UserProfileId* z tabeli *[core].[UserProfiles]*
RecordGuid	uniqueidentifier	☐	Identyfikator dokumentu biznesowego, dla którego wyznaczono właściciela zadania; odpowiada kolumnie *ACLId* w tabeli dokumentu biznesowego (StatusId->BusinessStatuses->BussinesDocuments->Table->ACLId)

6.1.13. Zapytania do webserwisów

> **Podsumowanie:**
> Akcje zapytań do usług webservice umożliwiają API wymianę danych między nAxiom i interfejsami innych serwisów, a także między tenantami w obrębie interfejsu Public API danego środowiska nAxiom. Dane zwracane przez te akcje mogą być zapisywane w bazie danych, używane do wyboru przejścia dla bloku decyzyjnego na workflow albo wyświetlane w specjalnej kontrolce na formularzu: lista rozwijana (WebService).

> ☛ Menu *Admin SPA* > `NARZĘDZIA` > `Akcje` > `Zapytania do WebSerwisów`

W TYM TEMACIE

- Opis akcji
- Przykład 1: Komunikacja między tenantami
- Przykład 2: Żądanie API do NBP
- Przykład 3: Własny warunek obsługi odpowiedzi
- Przykład 4: Żądanie SOAP API

OPIS AKCJI

W ramach akcji należy zdefiniować żądanie API do zdefiniowanego osobno źródła danych oraz zapytanie SQL, które wyodrębni odpowiednie dane z odpowiedzi. W zapytaniu dostępne są następujące zmienne:

- *@_Response*: treść (*body*) odpowiedzi; jeśli dla zwracanych danych określono format JSON, w zapytaniu użyj funkcji *OPENJSON*.
- *@_HTTPResponseStatusCode*: status odpowiedzi (*int*) np. *404*.
- *@_HTTPResponseStatusName*: nazwa statusu odpowiedzi np. *NotFound* dla statusu *404*.
- *@_HTTPResponseHeader*: nagłówki odpowiedzi w formacie JSON o następującej strukturze:

  ```
  {
  "header_1": ["value_1"],
  "header_2": ["value_2", "value_3"]
  }
  ```

  gdzie *header_x* to nazwa nagłówka odpowiedzi, a *value_x* to jego wartość. Wartości nagłówków można pozyskać używając funkcji SQL *OPENJSON*. W przypadku, kiedy nazwa nagłówka zawiera myślnik (-), odpowiadający mu klucz należy ująć w cudzysłów, na przykład:

  ```
  (SELECT top(1) value
   FROM OPENJSON({@_HTTPResponseHeader}, '$."Content-Type"')
  ```

W przypadku żądań *POST* dodatkowo konieczne jest określenie danych, które mają zostać wysłane w sekcji *body*. Można do tego użyć zapytania SQL, które zwróci odpowiednie wartości, a następnie w wysyłanych danych odwołać się do tych wartości, używając zmiennych *{@_QueryBody_SQLQueryData}*, gdzie *SQLQueryData* to nazwa kolumny zwróconej przez zapytanie.

Oprócz właściwości wspólnych dla wszystkich akcji, w akcji zapytań do webserwisów należy określić następujące właściwości:

- `Źródło danych`: zdefiniowane osobno źródło danych; patrz opis źródeł danych w Leksykonie nAxiom.
- `Metoda wywołania`: metoda wywołania API; dostępne wartości: `GET`, `POST`, `PUT`, `PATCH` i `DELETE`.
- `Typ ciała żądania`: określa format w jakim wysyłane są załączniki; pole dostępne dla metod typu *POST*, *PUT* i *PATCH*; dostępne wartości:
    - *RAW*: (wartość domyślna) dane w *body* żądania są tekstem, w przypadku załącznika są kodowane jako Base64; w tym trybie można użyć parametru *{@_AttachmentContent_FILEID}* do zakodowania danych załącznika jako Base64; przykład użycia przedstawiono poniżej.
    - *MULTIPART/FORM-DATA*: dla tego ustawienia zapytanie musi zwrócić następujące kolumny (w podanej kolejności):
        - *Type*: typ zawartości żądania. Akceptowane wartości to *Text* lub *File*
        - *Key*: klucz w zawartości żądania, pod którym zostanie umieszczona podana zawartość
        - *Content*: tekst dla zawartości typu *Text*
        - *FileId*: identyfikator załącznika z tabeli *core.Attachments* dla zawartości typu *File*
        - *ContentType*: format danych żądania dla zawartości o typie *Text*. Wartość domyślna to *application/json*

> **WSKAZÓWKA**
> W przypadku źródła danych typu *Tenant w środowisku nAxiom* pole `Rodzaj zawartości` jest wyświetlane w trybie tylko do odczytu z domyślną wartością *RAW*. W tym przypadku również możliwe jest korzystanie z parametru *{@_AttachmentContent_FILEID}*.

- `Identyfikator tenanta`: identyfikator tenanta, do którego ma zostać wysłane zapytanie; pole wyświetlane tylko w przypadku wyboru źródła danych typu Tenant w środowisku nAxiom ; w polu obsługiwane są referencje smart numbers, zmienne formularza oraz zapytania SQL. Patrz także przykład poniżej.
- `Ścieżka żądania`: specyfikacja punktu końcowego (endpoint) i parametrów żądania; jako parametrów można używać zmiennych typu *{@PoleFormularza}*. W przypadku żądania SOAP API ścieżka jest pobierana z definicji źródła i jest tylko do odczytu, a samo żądanie definiuje się osobno.
- `Content-Type`: pole dostępne w przypadku, gdy `Typ ciała żądania` to *RAW*; określa format danych tekstowych; w przypadku źródła danych typu *Tenant*... pole

jest tylko do odczytu i zawiera wartość domyślną *application/json*.

- `Nagłówki żądania`: dodatkowa specyfikacja nagłówków żądania; kliknij przycisk `Nowy nagłówek` i wpisz nazwę nagłówka i jego wartość (może być zwracana przez zapytanie SQL).
- `SQL wysyłanych danych`: (tylko żądania SOAP API i żądania REST API typu POST, PUT i PATCH) zapytanie SQL zwracające dane do wysłania w żądaniu, w zapytaniu są dostępne parametry z kontekstu bieżącego formularza w przypadku akcji wykonywanych po kliknięciu przycisku.
- `Wysyłane dane`: (tylko żądania SOAP API i żądania REST API typu POST, PUT i PATCH) dane wysyłane w żądaniu; w treści można korzystać z wartości z modelu formularza oraz wartości zwróconych przez zapytanie SQL; w tym drugim przypadku nazwę kolumny zwróconej przez zapytanie należy poprzedzić przedrostkiem *_QueryBody_*. Na przykład jeśli zapytanie zwraca kolumnę o nazwie *Name*, w wysyłanych danych można się do tej kolumny odwołać parametrem *{_QueryBody_Name}*

- `Tryb obsługi odpowiedzi`: dostępne wartości:
  - `domyślny`: zapytanie SQL określone jako procedura obsługi odpowiedzi jest wykonywane w przypadku, gdy zostanie zwrócony status HTTP z zakresu 200-299; w przeciwnym razie akcja zwróci wyjątek.
  - `własny warunek obsługi odpowiedzi`: w ramach procedury odpowiedzi możesz obsłużyć każdy status HTTP odpowiedzi; patrz przykład poniżej.

`Procedura obsługi odpowiedzi`: w przypadku akcji na formularzu zapytanie SQL UPDATE, które wyodrębnia dane z odebranej odpowiedzi i zapisuje je w odpowiednich polach tabeli źródłowej; w przypadku akcji użytej we właściwościach bloku decyzyjnego na workflow, zapytanie SQL SELECT, które zwróci **nazwę** przejścia. W zapytaniu można korzystać ze zmiennych *{@_Response}*, *{@_HTTPResponseStatusCode}*, *{@_HTTPResponseStatusName}* i *{@_HTTPResponseHeader}*.

PRZYKŁAD 1: KOMUNIKACJA MIĘDZY TENANTAMI

W środowisku wielu tenantów można używać akcji zapytań do webservice do wysyłania żądań do serwisu PublicAPI dowolnego tenanta. Jeśli w akcji zostanie użyte źródło danych typu Tenant w środowisku nAxiom . Wystarczy podać identyfikator docelowego tenanta oraz określić ścieżkę, parametry i treść żądania. Dzięki zastosowaniu uwierzytelniania opartego o tokeny TOTP, nie trzeba osobno obsługiwać uwierzytelniania.

Poniżej podano przykład żądania do punktu końcowego *Document/CreateWithAttachments*, który obsługuje tworzenie dokumentu wraz z zapisaniem załączników.

- Źródło danych: typu *Tenant w środowisku nAxiom*

- Metoda wywołania: **POST**
- Ścieżka żądania:

```
Document/CreateWithAttachments?documentDefinitionId=631FF...&skipWorkflowActions=true
```

- SQL Wysyłanych danych:

```
SELECT CONCAT('{@_AttachmentContent_', {@AttachmentId}, '}') AS [AttachmentContent]
```

gdzie *{@AttachmentId}* to identyfikator załącznika z tabeli *core.Attachments*.

- Wysyłane dane:

```
{
  "Model": {
      "Id": 0,
      "Temat": "<treść tematu>",
      "Pismo": "<treść pisma>"
  },
  "Attachments": [
      {
          "Base64Content": "{@_QueryBody_AttachmentContent}",
          "FileName": "<nazwa pliku załącznika>",
          "Description": "<opis pliku załącznika>",
          "AttachmentCategoryId": "<rowGuid kategorii załączników>"
      }
  ]
}
```

Zmienna *{@_QueryBody_AttachmentContent}* zawiera dane załącznika zakodowane jako Base64.

PRZYKŁAD 2: ŻĄDANIE API DO NBP

Źródło danych: `http://api.nbp.pl/`

Ścieżka żądania: `api/exchangerates/rates/a/{@Waluta}/{@Data}/?format=json`

Żądanie zwraca średni kurs wymiany z tabeli A Narodowego Banku Polskiego dla waluty o symbolu przekazanym zmienną *{@Waluta}* na dzień określony wartością zmiennej *{@Data}*. Odpowiedź jest przesyłana w formacie JSON. Przykładowa odpowiedź dla waluty *USD* na dzień *2021-11-16* to:

```
{
"table":"A",
"currency":"dolar amerykański",
"code":"USD",
"rates":[
        {
          "no":"221/A/NBP/2021",
          "effectiveDate":"2021-11-16",
          "mid":4.1061
        }
        ]
}
```

Procedura obsługi odpowiedzi:

- aktualizacja pola formularza:

```
UPDATE [dbo].[Rozliczenia]
SET Kurs = (SELECT [mid] FROM OPENJSON({@_Response}, '$.rates')
WITH ([no] varchar(20),
    [effectiveDate] varchar(20),
    [mid] varchar(20)))
WHERE Id={@Id}
```

- wybór nazwy przejścia dla bloku decyzyjnego:

```
SELECT
 CASE
 WHEN (
   SELECT [mid]*{@Kwota} from openjson({@_Response}, '$.rates')
      WITH ([no] varchar(20), [effectiveDate] varchar(20), [mid] decimal(18,4))
      ) > 300
 THEN 'do kontroli'
 ELSE 'zatwierdź automatycznie'
 END
```

Literały *do kontroli* i *zatwierdź automatycznie* to wartości właściwości `Nazwa przejścia` określone dla przejść wychodzących z bloku decyzyjnego.

PRZYKŁAD 3: WŁASNY WARUNEK OBSŁUGI ODPOWIEDZI

Przykładowa procedura obsługi odpowiedzi z własnym warunkiem.

```
DECLARE @msg AS nvarchar(1000)
IF({@_HTTPResponseStatusCode} >= 200 AND {@_HTTPResponseStatusCode} <= 299)
update [dbo].[testy_WS]
set [Odp] = (SELECT * FROM OPENJSON({@_Response})
    WITH (requestedValue nvarchar(4000) '$.description')),
[StatusOdp] = concat({@_HTTPResponseStatusCode}, ' ',
                                        {@_HTTPResponseStatusName}),
[Naglowki] = {@_HTTPResponseHeader},
[ContentType] = (SELECT top(1) value FROM OPENJSON({@_HTTPResponseHeader}, '$."Content-Type"'))
where Id = {@_Id}
ELSE
BEGIN
SELECT @msg = concat('[MSG] ERROR ', {@_HTTPResponseStatusCode},' ',
                                         {@_HTTPResponseStatusName});
RAISERROR(@msg,11,1);
END
```

PRZYKŁAD 4: ŻĄDANIE SOAP API

Przykład uzyskania wartości z żądania dla SoapApi za pomocą metody nodes():

https://www.dataaccess.com/webservicesserver/NumberConversion.wso

```
DECLARE @xmlRes XML =
  REPLACE(ISNULL({@_Response},'ERROR'),'<?xml version="1.0" encoding="utf-8"?>','');
DECLARE @result NVARCHAR(MAX) = (
SELECT n.value('.','NVARCHAR(MAX)') as CODE
FROM @xmlRes.nodes('declare namespace s="http://schemas.xmlsoap.org/soap/envelope/";
declare namespace x="http://www.dataaccess.com/webservicesserver/";
/s:Envelope/s:Body/x:NumberToWordsResponse/x:NumberToWordsResult') as a(n)
);
UPDATE [dbo].[Invoice]
SET [PriceInWords] = ISNULL(@result,' Pusta'),
WHERE Id={@_Id}
```

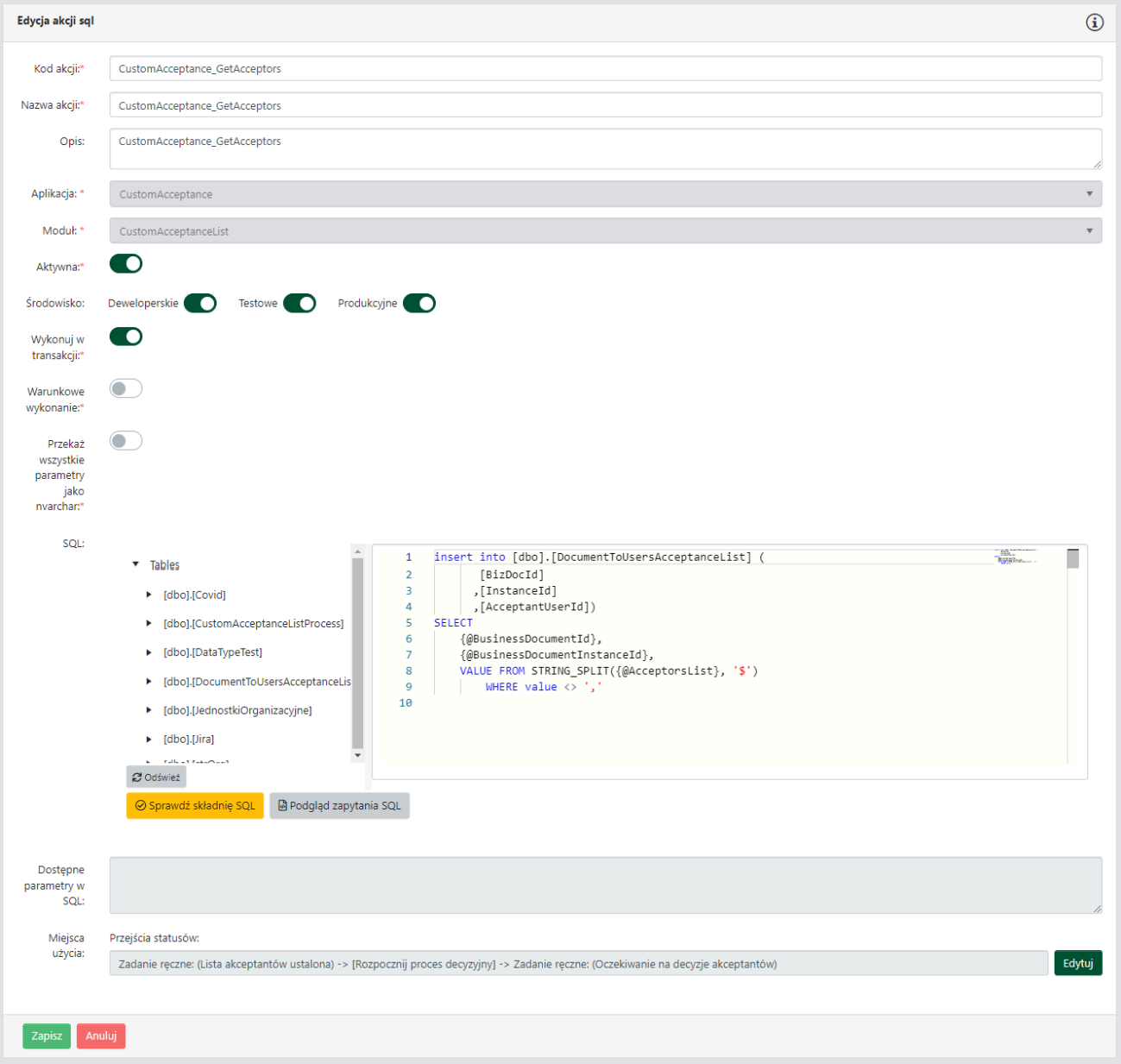
Powiązane tematy:

- Lista wyboru (WebService)
- Blok decyzyjny

6.1.14. SQL do bazy systemowej

Podsumowanie:
Akcje SQL do bazy systemowej służą do wykonywania zapytań modyfikujących, które zawierają instrukcje *INSERT INTO*, *UPDATE*, *DELETE* lub *ALTER*, do uruchamiania procedur bazy danych instrukcją *EXEC* oraz na przykład do obsługi walidacji danych z użyciem funkcji *RAISERROR*.

☛ *AdminSPA* > `Narzędzia` > `Akcje` > `SQL do bazy systemowej`



Akcja SQL do bazy systemowej

Oprócz właściwości wspólnych dla wszystkich akcji, w akcji SQL do bazy systemowej należy określić następujące właściwości:

`Wykonuj w transakcji`: (domyślnie włączone) powoduje, że akcja jest wykonywana w jednej transakcji z kolejnymi akcjami, które mają ten przełącznik włączony (mogą to być inne akcje tego rodzaju, akcje procedur bazodanowych oraz sytemowa akcja zapisu); w takim przypadku wykonanie każdej akcji musi zakończyć się powodzeniem, aby transakcja została zatwierdzona. Jeśli wynik bieżącej akcji ma zostać zatwierdzony niezależnie od wyniku wykonania pozostałych akcji, należy wyłączyć ten przełącznik.

`Przekaż wszystkie parametry jako nvarchar`: przełącznik zapewniający kompatybilność wsteczną z wcześniejszymi wersjami, kiedy wartości zmiennych typu *{@nazwa}* (z kontekstu formularza i systemowe) w zapytaniu SQL były domyślnie przekazywane jako tekst (typ *nvarchar*); od wersji 1.4.1.3 parametry w zapytaniach SQL są przekazywane zgodnie z ich typem natywnym.

`SQL`: instrukcje SQL, które mają zostać wykonane po uruchomieniu akcji; przycisk `Sprawdź składnię SQL` sprawdza instrukcje SQL pod względem składniowym i wyświetla odpowiedni komunikat; przycisk `Podgląd zapytania SQL` wyświetla okno dialogowe z kodem zapytania SQL oraz dostępnymi parametrami.

PRZYKŁAD 1

```
DELETE FROM [dbo].[WyjsciaRejestr]
WHERE [dbo].[WyjsciaRejestr].Id = JSON_VALUE({@Event}, '$.id')
```

PRZYKŁAD 2

```
BEGIN
 IF (SELECT COUNT(1) FROM [ERO].[MIG_T_RZR_ZGLOSZENIA]) > 0
  IF EXISTS (
      SELECT 1
        FROM sys.triggers
       WHERE parent_class = 1
         AND OBJECT_NAME(parent_id) = 'T_RZR_ZGLOSZENIA'
         AND name like 'audit_delete_%'
            )
 ALTER TABLE [ERO].[T_RZR_ZGLOSZENIA] DISABLE TRIGGER [audit_delete_T_RZR_ZGLOSZENIA]
END
```

PRZYKŁAD 3

```
UPDATE [WNM].[NOWY_MATERIAL]
SET [CENY_UZUPELNIONE] = CAST(1 AS BIT)
WHERE Id IN
  (SELECT [value] FROM OPENJSON({@KodSekcji_Selected}))
```

W tym zapytaniu akcja aktualizuje pola w rekordach z tablicy *Selected* w modelu formularza. Ta tablica zawiera identyfikatory rekordów zaznaczonych na liście osadzonej w sekcji formularza o podanym kodzie.

OSTRZEŻENIE
Składnia *{@_Section_KodSekcji_Selected}* może być używana wyłącznie w warunku zależności od sekcji. W innych przypadkach (np. w akcji SQL), aby odwołać się do listy zaznaczonych rekordów na liście osadzonej w formularzu, należy użyć zmiennej *{@KodSekcji_Selected}*.

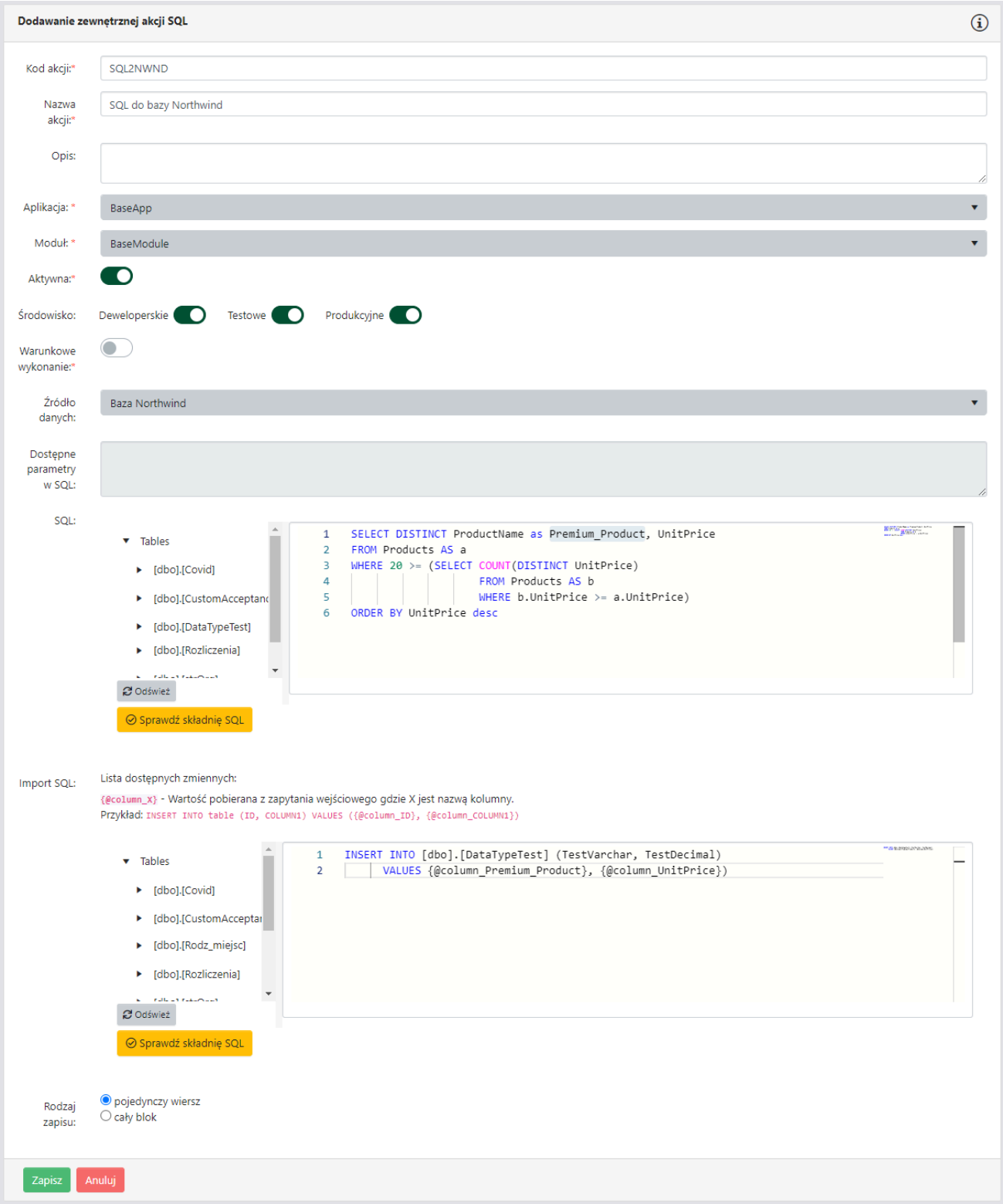
Powiązane tematy:

- Wspólne parametry akcji

6.1.15. SQL do baz zewnętrznych

Podsumowanie:
Akcja *Zewnętrzny SQL* umożliwia pobranie danych z zewnętrznego źródła danych i zapisanie ich w tabeli bazy danych nAxiom. Ta akcja wymaga określenia dwóch zapytań: pobierającego dane i zapisującego.

☛ *AdminSPA* > `NARZĘDZIA` > `Akcje` > `SQL do baz zewnętrznych`



SQL do baz zewnętrznych

Oprócz właściwości wspólnych dla wszystkich akcji, w akcji SQL do baz zewnętrznych należy określić następujące właściwości:

`Źródło danych`: zdefiniowane osobno źródło danych; patrz opis źródeł danych w Leksykonie nAxiom.

`SQL`: zapytanie pobierające dane z zewnętrznego źródła; na przykład:

```
SELECT ColumnA FROM ExternalTable
```

`Import SQL`: zapytanie zapisujące dane pobrane z zewnętrznego źródła w tabeli bazy danych nAxiom; na przykład:

```
INSERT INTO LocalTable (ColumnB)
VALUES ({@column_ColumnA})
```

`Rodzaj zapisu`: poziom zatwierdzania transakcji podczas zapisu zaimportowanych rekordów do bazy danych; dostępne wartości:

- `Pojedynczy wiersz`: zatwierdza transakcję po zapisaniu każdego rekordu ze zwróconego zestawu; błędy podczas zapisu indywidualnych rekordów nie przerywają operacji.
- `Cały blok`: zatwierdza transakcję po zapisaniu całego bloku zwróconych rekordów; błędy podczas zapisu indywidualnego rekordu przerywa operację.

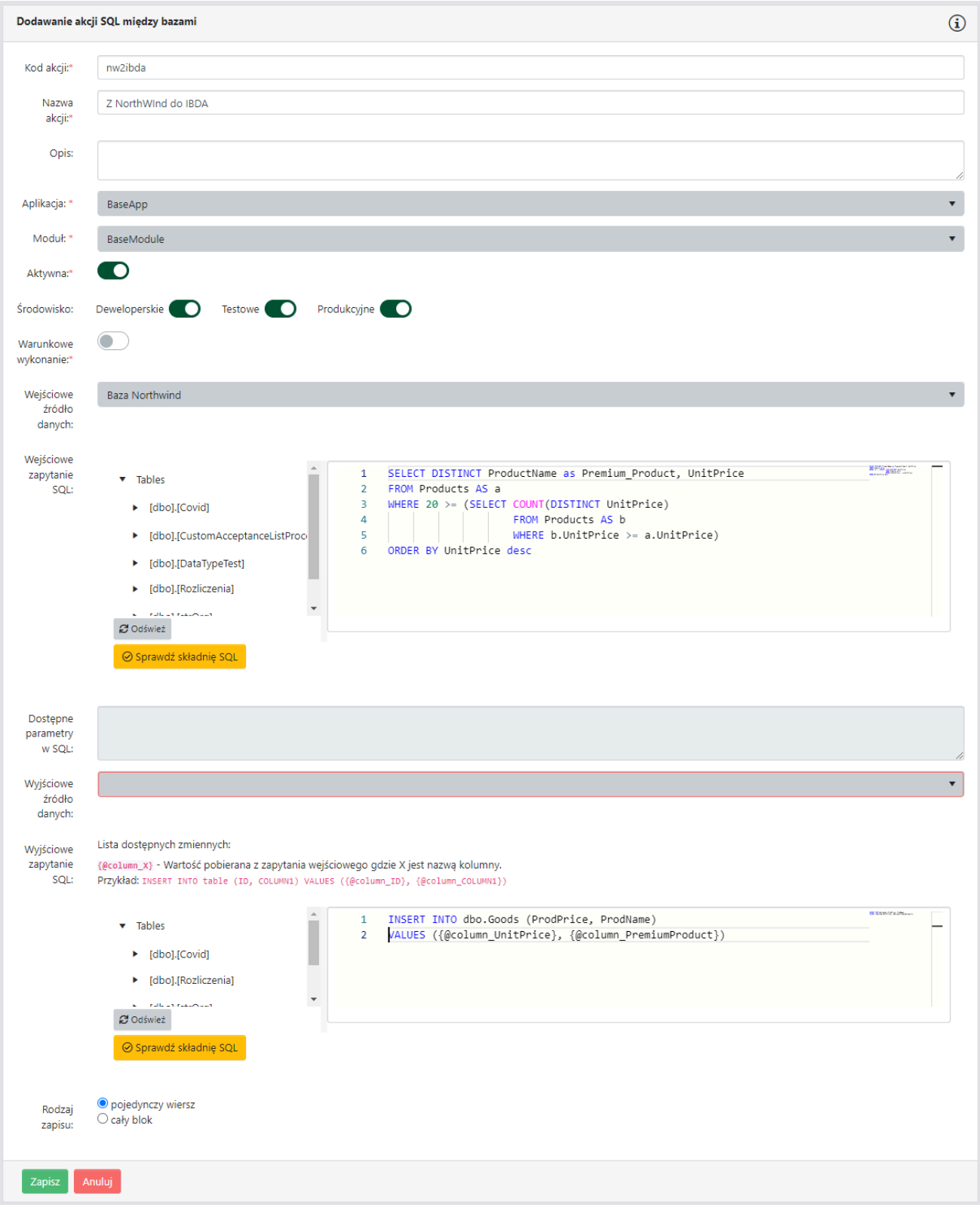
Powiązane tematy:

- Źródła danych

6.1.16. SQL między bazami danych

Podsumowanie:
Akcja SQL między bazami danych umożliwia pobranie danych z zewnętrznej bazy danych i zapisanie ich w innej zewnętrznej bazie danych. Bazy danych źródłowa i docelowa muszą zostać zdefiniowane jako źródła danych. Ta akcja wymaga określenia dwóch zapytań: pobierającego dane (typu *SELECT*) i zapisującego dane (typu *INSERT INTO* lub *UPDATE*).

☛ *AdminSPA* > `NARZĘDZIA` > `Akcje` > `SQL między bazami danych`



SQL między bazami danych

Oprócz właściwości wspólnych dla wszystkich akcji, w akcji SQL między bazami należy określić następujące właściwości:

- `Wejściowe źródło danych`: zdefiniowane osobno źródło danych z którego zostaną pobrane dane.

- `Wejściowe zapytanie SQL`: zapytanie pobierające dane z zewnętrznego źródła.
- `Wyjściowe źródło danych`: zdefiniowane osobno źródło danych, w którym zostaną zapisane dane pobrane ze źródła wejściowego.
- `Wyjściowe zapytanie SQL`: zapytanie wstawiające dane do zewnętrznego źródła; dane pobrane ze źródła wejściowego są reprezentowane przez zmienne *{@column_NazwaKolumny}*, gdzie *NazwaKolumny* to nazwa kolumny zwracanej przez zapytanie wejściowe.
- `Rodzaj zapisu`: poziom zatwierdzania transakcji podczas zapisu zaimportowanych rekordów do bazy danych; dostępne wartości:
- `Pojedynczy wiersz`: zatwierdza transakcję po zapisaniu każdego rekordu ze zwróconego zestawu; błędy podczas zapisu indywidualnych rekordów nie przerywają operacji.
- `Cały blok`: zatwierdza transakcję po zapisaniu całego bloku zwróconych rekordów; błędy podczas zapisu indywidualnego rekordu przerywa operację.

PRZYKŁAD

Zapytanie zwracające dane z wejściowego źródła danych:

```
SELECT DISTINCT ProductName AS Premium_Product, UnitPrice
FROM Products AS a
WHERE 20 >= (SELECT COUNT(DISTINCT UnitPrice)
                    FROM Products AS b
                    WHERE b.UnitPrice >= a.UnitPrice)
ORDER BY UnitPrice desc
```

Zapytanie wstawiające zwrócone dane w docelowym źródle danych:

```
INSERT INTO dbo.Goods (ProdPrice, ProdName)
VALUES ({@column_UnitPrice}, {@column_PremiumProduct})
```

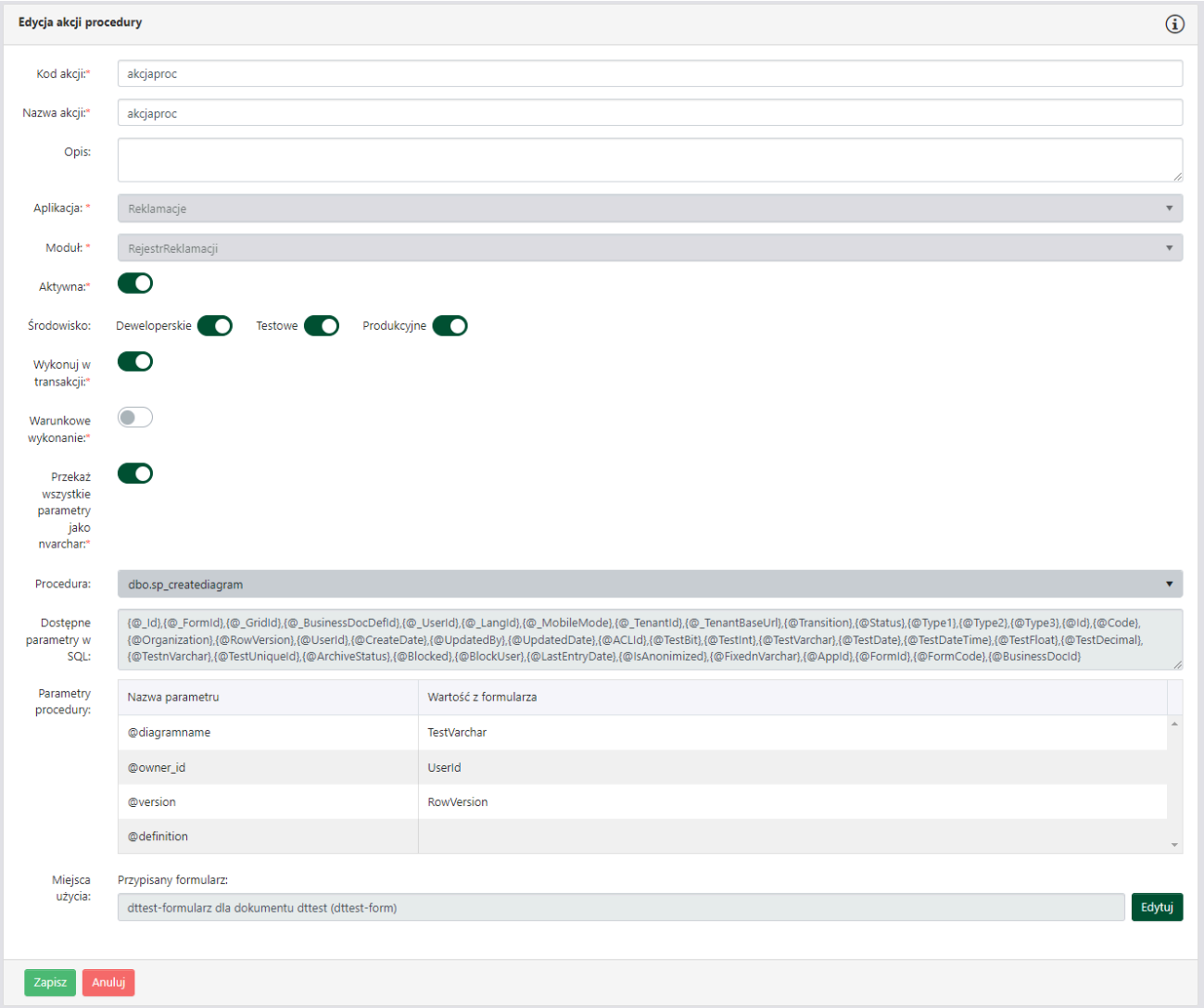
Powiązane tematy:

- Wspólne parametry akcji
- Źródła danych

6.1.17. Procedury bazodanowe

Podsumowanie:
Akcje z kategorii `Procedury bazodanowe` służą do uruchamiania *procedur składowanych SQL* (*stored procedures*) wraz z odpowiednimi parametrami.

☛ *AdminSPA* > `Narzędzia` > `Akcje` > `Procedury bazodanowe`



Akcja procedur bazodanowych

Oprócz właściwości wspólnych dla wszystkich akcji, w akcji procedury bazodanowej należy określić następujące właściwości:

`Wykonuj w transakcji`: (domyślnie włączone) powoduje, że akcja jest wykonywana w jednej transakcji z kolejnymi akcjami, które mają ten przełącznik włączony (mogą to być inne akcje tego rodzaju, akcje procedur bazodanowych oraz sytemowa akcja zapisu); w takim przypadku wykonanie każdej akcji musi zakończyć się powodzeniem, aby transakcja

została zatwierdzona. Jeśli wynik bieżącej akcji ma zostać zatwierdzony niezależnie od wyniku wykonania pozostałych akcji, należy wyłączyć ten przełącznik.

`Przekaż wszystkie parametry jako nvarchar`: przełącznik zapewniający kompatybilność wsteczną z wcześniejszymi wersjami, kiedy wartości zmiennych typu *{@nazwa}* (z kontekstu formularza i systemowe) w zapytaniu SQL były domyślnie przekazywane jako tekst (typ *nvarchar*); od wersji 1.4.1.3 parametry w zapytaniach SQL są przekazywane zgodnie z ich typem natywnym.

`Procedura`: nazwa procedury składowanej zdefiniowanej w bazie danych.

`Parametry procedury`: tabela parametrów używanych przez procedurę; kolumna `Nazwa parametru` zawiera nazwy parametrów w formacie *@NazwaParametru*, kolumna `Wartość z formularza` zawiera pola formularza, których wartości zostaną podstawione jako wartości parametrów. Obecnie nie są obsługiwane wywołania procedur bez parametrów.

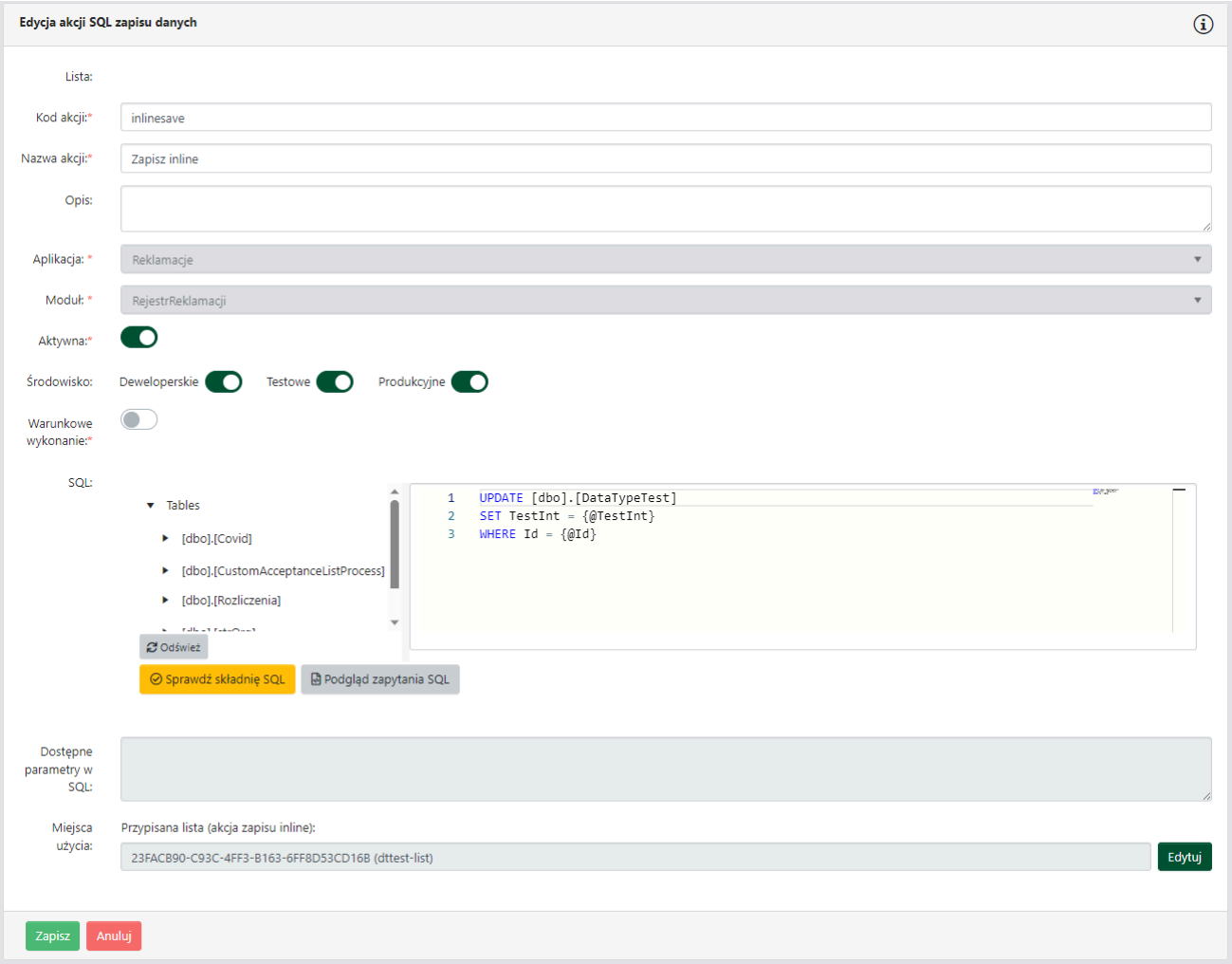
Powiązane tematy:

- Wspólne parametry akcji

6.1.18. SQL zapisu danych

Podsumowanie:
Akcje SQL zapisu danych służą do zapisywania danych w bazie przy użyciu instrukcji `INSERT INTO` i `UPDATE`.

☛ *AdminSPA* > `Narzędzia` > `Akcje` > `SQL zapisu danych`



Akcja zapisu

Oprócz właściwości wspólnych dla wszystkich akcji, w akcji SQL zapisu danych należy określić następujące właściwości:

`SQL`: instrukcje SQL, które mają zostać wykonane po uruchomieniu akcji; przycisk `Sprawdź składnię SQL` sprawdza instrukcje SQL pod względem składniowym i wyświetla odpowiedni komunikat; przycisk `Podgląd zapytania SQL` wyświetla okno dialogowe z kodem zapytania SQL oraz dostępnymi parametrami.

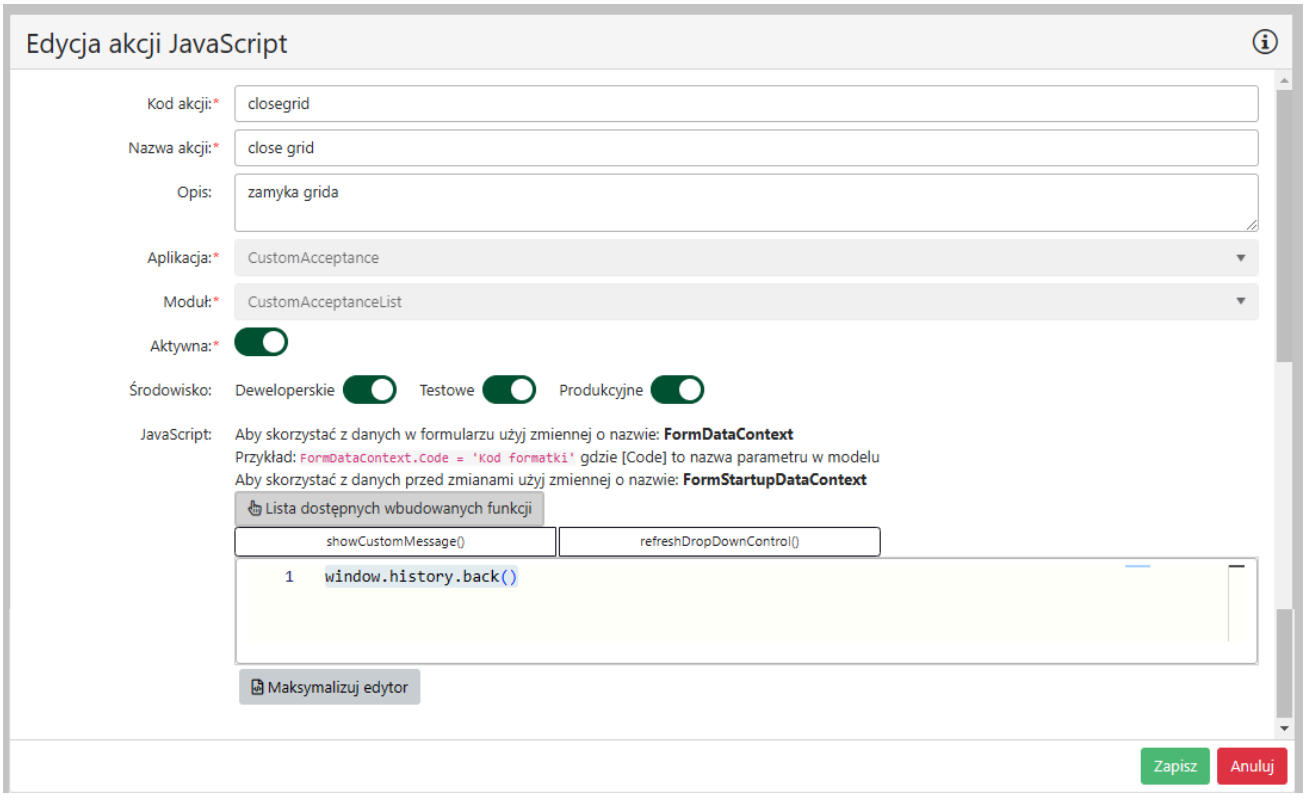
Powiązane tematy:

- Wspólne parametry akcji

6.1.19. JavaScript

Podsumowanie:
Akcje JavaScript umożliwiają uruchamianie kodu JavaScript w aplikacjach nAxiom. W akcjach można odwoływać się do pól formularza, używając składni *FormDataContext.PoleFormularza*, gdzie *PoleFormularza* to wartość właściwości `Pole formularza`. Możliwe jest także odwoływanie się do początkowych wartości z kontekstu (po otwarciu formularza). Należy wówczas użyć selektora *FormStartupDataContext*. W akcji JavaScript dostępna jest także funkcja odświeżania listy wyboru: *refreshDropDownControl('KodListy')*, gdzie *KodListy* to wartość właściwości *Kod* kontrolki listy wyboru.

☛ *AdminSPA* > `NARZĘDZIA` > `Akcje` > `JavaScript`



Akcja JavaScript

Oprócz właściwości wspólnych dla wszystkich akcji, w akcji JavaScript należy określić następujące właściwości:

`JavaScript`: funkcja JavaScript, która będzie wykonywana po wywołaniu akcji.

WSKAZÓWKA

Powyżej okna kodu znajduje się przycisk `Lista dostępnych wbudowanych funkcji`. Po kliknięciu wyświetlane są dostępne wbudowane funkcje Javascript. Kliknięcie funkcji powoduje wstawienie jej do okna kodu.

ODWOŁANIA DO KONTROLEK FORMULARZA

Począwszy od wersji 1.15.7.0, w związku z aktualizacją biblioteki Kendo, zmienił się sposób renderowania kontrolek w aplikacji Front SPA. Obecnie kontrolki Kendo renderowane są jako element hostujący zawierający właściwy element input.

W przypadkach zastosowania w formularzu własnego kodu JavaScript mającego na celu wstawianie danych do inputów kontrolek (np. z urządzeń peryferyjnych komputera) może być konieczne jego dostosowanie polegające na wprowadzeniu odwołań bezpośrednio do inputów kontrolek formularza.

Dla pola tekstowego

```
<kendo-textbox
  id="SerialNumber560"
  data-code="RPT-StockRotation-ccs75">
  <input
    class="k-input-inner"
    type="text"
    placeholder="Miejsce na kod SN">
</kendo-textbox>
```

Zalecanym rozwiązaniem jest wyszukiwanie kontrolki za pomocą stabilnego atrybutu *data-code* przechowującego kod kontrolki, a następnie pobranie znajdującego się w niej elementu input:

```
const textbox = document.querySelector('[data-code="RPT-StockRotation-ccs75"]');
const input = textbox.querySelector("input");
```

Takie podejście jest niezależne od dynamicznie generowanych identyfikatorów i zapewnia poprawny dostęp do wartości kontrolki we wszystkich środowiskach.

Nieprawidłowa (zdeprecjonowana) instrukcja:

```
const sn_input = document.getElementById("SerialNumber560");
```

zwraca element *kendo-textbox* (host), a nie pole wejściowe *input*. Operacje wykonywane bezpośrednio na wartości pola mogą więc działać nieprawidłowo.

6.1.20. Wyślij e-mail

> **Podsumowanie:**
> Ten rodzaj akcji e-mail służy do wysyłania wiadomości e-mail generowanych na podstawie zdefiniowanego osobno szablonu wiadomości. Adresat wiadomości może zostać wpisany na stałe lub być zwracany zapytaniem SQL.

> ☛ *AdminSPA* > `NARZĘDZIA` > `Akcje` > `Wyślij/Odbierz e-mail`

Oprócz właściwości wspólnych dla wszystkich akcji, w akcji e-mail należy określić następujące właściwości:

`Tytuł wiadomości`: temat wysyłanej wiadomości e-mail.

`Szablon email`: nazwa zdefiniowanego szablonu e-mail (`NARZĘDZIA` > `Szablony` > `Szablony e-mail`).

`Edytuj szablon`: wyświetla okno edycji wybranego szablonu e-mail.

`Nowy szablon`: wyświetla okno tworzenia definicji szablonu; utworzony szablon zostaje automatycznie ustawiony jako wybrany w akcji.

`Wyślij od`: adres e-mail nadawcy wiadomości.

`Wyślij do każdego adresata „Do” osobno`: włączenie powoduje, że do każdego adresata określonego na karcie `Adresaci Do` wysyłana jest osobna kopia wiadomości; każda taka wiadomość zostanie również wysłana do adresatów określonych na kartach `Adresaci DW` i adresaci `UDW`.

`Obsługa zastępstw`: włączenie powoduje, że kopia wiadomości jest wysyłana także do osób zastępujących adresatów. Obsługa zastępstw dotyczy tylko adresatów definiowanych na kartach:

- Interesariusze,
- Posiadacze ról globalnych,
- Użytkownicy wskazani przez zapytanie SQL.

W takim przypadku do wiadomości dodawane są informacje:

- dla zastępowanego: informacja o tym, że kopia wiadomości została wysłana do

zastępującego i dane tej osoby,
- dla zastępującego: informacja o tym, że ta wiadomość to kopia oryginalnej wiadomości wysłanej do zastępowanego, informacje o zastępstwie, informacje o oryginalnych adresatach wiadomości.

W przypadku adresatów podanych w polu UDW (BCC), kopia wiadomości wysyłana do osób zastępujących nie zawiera żadnych dodatkowych informacji.

> **WSKAZÓWKA**
> Obsługa zastępstw dotyczy tylko tech zastępstw, w których włączono przełącznik `Przekazuj powiadomienia email z systemu do zastępcy`.

`Adresaci Do`, `Adresaci DW`, `Adresaci UDW`: karty na których można zdefiniować adresatów wiadomości. Każda karta ma pięć kart podrzędnych, na których można definiować adresatów odrębnymi metodami.

`Interesariusze`: jako adresaci zostaną dodani użytkownicy przypisani do wybranych grup interesariuszy dla instancji dokumentu zwróconych przez zapytanie:

- `Dokument biznesowy`: definicja dokumentu, dla której zdefiniowano grupy interesariuszy.
- `Instancje`: zapytanie SQL zwracające identyfikatory instancji dokumentu (wymagane jest pełne zapytanie ze wskazaniem tabeli w klauzuli *FROM*); jako adresaci zostaną wybrani użytkownicy przypisani do wybranych grup interesariuszy dla tych instancji.
- Lista grup interesariuszy dla wybranej definicji dokumentu; z listy należy wybrać grupy interesariuszy, z których będą wybierani adresaci.

`Posiadacze ról globalnych`: jako adresaci zostaną dodani użytkownicy, którym przypisano role wybrane z listy ról dostępnej na karcie.

`Użytkownicy wskazani przez zapytanie SQL`: jako adresaci zostaną dodani użytkownicy, których identyfikatory zostaną zwrócone przez zapytanie SQL.

`Adresy e-mail wskazane przez zapytanie SQL`: jako adresaci zostaną dodani właściciele adresów e-mail zwróconych zapytaniem SQL; zapytanie może zwrócić wyniki w formie kolumny lub ciągu wartości rozdzielonych znakiem średnika.

`Adresy e-mail wskazane bezpośrednio`: jako adresaci zostaną dodani właściciele adresów e-mail wpisanych w kontrolce wielowyboru (obsługiwana jest notacja smart numbers).

Na każdej z tych kart znajduje się przełącznik `Włączone`, którym można dezaktywować konfigurację adresatów bez konieczności usunięcia jej. Zakładki z aktywną konfiguracją posiadają pogrubiony tekst i systemowy ciemno-zielony kolor.

Powiązane tematy:

- Szablony e-mail

6.1.21. Import danych z Excel

Podsumowanie:
Akcje importu danych z pliku programu Excel służą do importowania danych z arkusza kalkulacyjnego do tabel użytkownika. W przypadku importu do tabel w modelu standardowym i uproszczonym należy brać pod uwagę, że dla zaimportowanych dokumentów nie zostaną utworzone wpisy uprawnień ACL. W takim przypadku należy zaimportować dane do tabeli pomocniczej, a następnie użyć akcji importu dokumentów.

Po uruchomieniu akcja wyświetla okienko, w którym należy wybrać źródłowy plik Excel.

☛ *AdminSPA* > `NARZĘDZIA` > `Akcje` > `Import danych z Excel`

Akcje importu danych z pliku programu Excel służą do importowania do tabeli dokumentu biznesowego danych z arkusza kalkulacyjnego. Po uruchomieniu akcja wyświetla okienko, w którym należy wybrać źródłowy plik.

Edycja akcji importu do formatu Excel

Kod akcji:* Miejscowosci

Nazwa akcji:* Miejscowosci

Opis:

Aplikacja: * BaseApp

Moduł: * BaseModule

Aktywna:*

Środowisko: Deweloperskie Testowe Produkcyjne

Warunkowe wykonanie:*

Ilość pomijanych wierszy: 2

SQL importu:

Tables
- [dbo].[Covid]
- [dbo].[CustomAcceptanceListProcess]
- [dbo].[Rozliczenia]

Odśwież

Sprawdź składnię SQL

```
1   INSERT INTO [dbo].[Miejscowosci] ([Woj])
2 SELECT
3 {@Column_A} AS [Woj]
```

Dostępne parametry w SQL:

Mapowanie kolumn:

Nowe mapowanie

Nazwa kolumny	Nazwa w SQL	
Brak wyników do wyświetlenia		

Lista dostępnych zmiennych:

`@Column_X` - Wartość pobierana z kolumny w tabeli Excel, gdzie X jest to nazwa kolumny w arkuszu Excel (np. `@Column_B`).
`@Mapper_X` - Wartość pobierana z tabeli mapowanych kolumn, gdzie 'nazwa kolumny' jest to nazwą kolumny w arkuszu Excel (np. A, D, F), a 'nazwa w SQL' jest to dowolna nazwa zmiennej z przedrostkiem 'Mapper_'.

Uwaga: W przypadku importu danych z plików CSV, pierwsza linia jest traktowana jako nagłówek, w którym znajduje się lista kolumn.

Przykład:

```
INSERT INTO [dbo].[Example]([ImportDate], [Region], [ItemName], [Units])
SELECT
  GETDATE(),
  {@Mapper_Region} AS [Region],
  {@Column_B} AS [Item],
  {@Column_D} AS [Unit]
```

Miejsca użycia: Pola w formularzach:

Importuj [dttest-formularz dla dokumentu dttest] (dttest-form Edytuj

Zapisz Anuluj

Definiowanie akcji importu

Oprócz właściwości wspólnych dla wszystkich akcji, w akcji importu danych z Excel należy określić następujące właściwości:

- `Ilość pomijanych wierszy`: liczba początkowych wierszy arkusza, która ma zostać pominięta podczas wczytywania danych.
- `SQL Importu`: zapytanie SQL, które wstawia do określonych kolumn tabeli dane wybranie podzapytaniem z arkusza; aby odwołać się do kolumny *A* w arkuszu, należy użyć zmiennej *{@Column_A}* lub *{@Mapper_nazwa}*. W pierwszym przypadku akcja

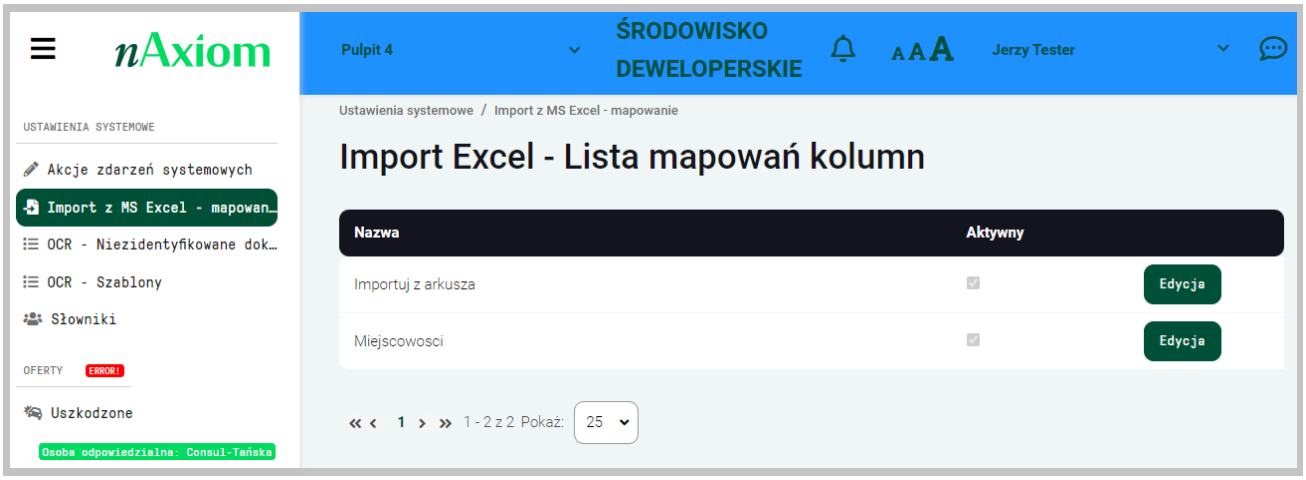
będzie na sztywno przypisana do konkretnych kolumn arkusza, w drugim możliwe będzie przypisywanie różnych kolumn do tej samej zmiennej, co pozwoli importować dane z arkuszy o różnym układzie kolumn; patrz opis poniżej.

- `Mapowanie kolumn`: pozwala zdefiniować mapowanie kolumn arkusza; mapowanie można zmieniać w aplikacji FrontSPA i w ten sposób dostosowywać akcję importu do nazw kolumn w arkuszu źródłowym; patrz opis poniżej. Aby zdefiniować mapowanie, kliknij przycisk `Nowe mapowanie` i wpisz nazwę kolumny arkusza w polu `Nazwa kolumny` oraz dowolną nazwę zmiennej w polu `Nazwa w SQL`; tej nazwy zmiennej można użyć w zapytaniu SQL w formacie *{@Mapper_nazwazmiennej}*; kliknij przycisk `Dodaj`, aby zapisać mapowanie.

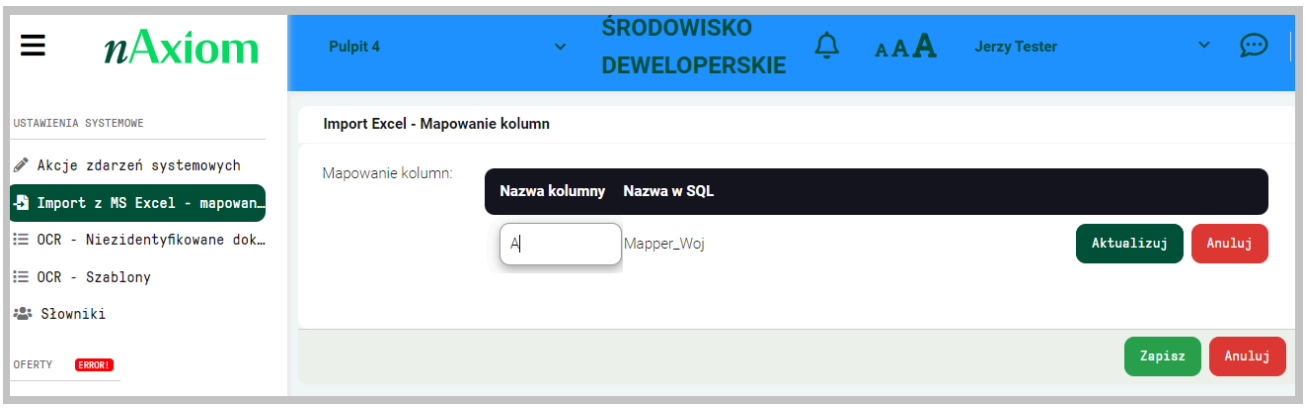
MAPOWANIE KOLUMN

Jeśli w akcji import z arkusza Excel zdefiniowano mapowanie kolumn arkusza na zmienne, uprawnieni użytkownicy będą mogli dostosowywać akcję importu do arkusza źródłowego w aplikacji FrontSPA. Służy do tego polecenie `Import danych z Excel - mapowanie kolumn` (sekcja menu `Ustawienia systemowe`). Dostęp do tego polecenia menu jest chroniony uprawnieniem PBA o takiej samej nazwie, które znajduje się w sekcji `Użytkowanie aplikacji`.

Po wybraniu polecenia wyświetlana jest lista zdefiniowanych akcji importu, a po kliknięciu przycisku `Edycja` wyświetlane są mapowania zdefiniowane w tej akcji. Po kliknięciu przycisku `Edytuj` obok mapowania, można zmienić nazwę kolumny w źródłowym arkuszu Excel. Dzięki temu można używać jednej akcji importu do importowania danych z arkuszy o różnym układzie kolumn.



FrontSPA: Import Excel - Lista mapowań kolumn



FrontSPA: Edycja mapowania kolumny

Powiązane tematy:

- Obszar Użytkowanie aplikacji

6.1.22. Eksport danych do Excel

Podsumowanie:
Akcja eksportu danych do pliku programu Excel pozwala wyeksportować dane z tabeli dokumentu biznesowego do nowego lub do istniejącego pliku w formacie Excel. W tym drugim przypadku plik musi zostać wskazany jako *szablon Excel*.

☛ *AdminSPA* > `NARZĘDZIA` > `Akcje` > `Eksport danych do Excel`

Oprócz właściwości wspólnych dla wszystkich akcji, w akcji eksportu do pliku Excel należy określić następujące właściwości:

`Nazwa pliku`: nazwa pliku wynikowego; może to być ustalona nazwa lub wartość zwrócona przez zapytanie SQL (wymaga przełączenia typu pola wprowadzania przez kliknięcie przycisku z prawej strony pola); niedozwolone znaki w zwróconej wartości zostaną zastąpione łącznikami (-).

- `Eksportuj jako`:
  - `Nowy plik`: eksportuje dane do nowego pliku.
  - `Uzupełniony istniejący szablon`: dołącza wyeksportowane dane do pliku wskazanego jako szablon i zapisuje pod podaną nazwą.

> **UWAGA**
> Aby dopisywanie eksportowanych danych do istniejącego pliku działało, po wyzwoleniu akcji w aplikacji FrontSPA użytkownik musi wybrać format *xlsx*, w przeciwnym razie dane zostaną wyeksportowane do nowego pliku w formacie *CSV*.

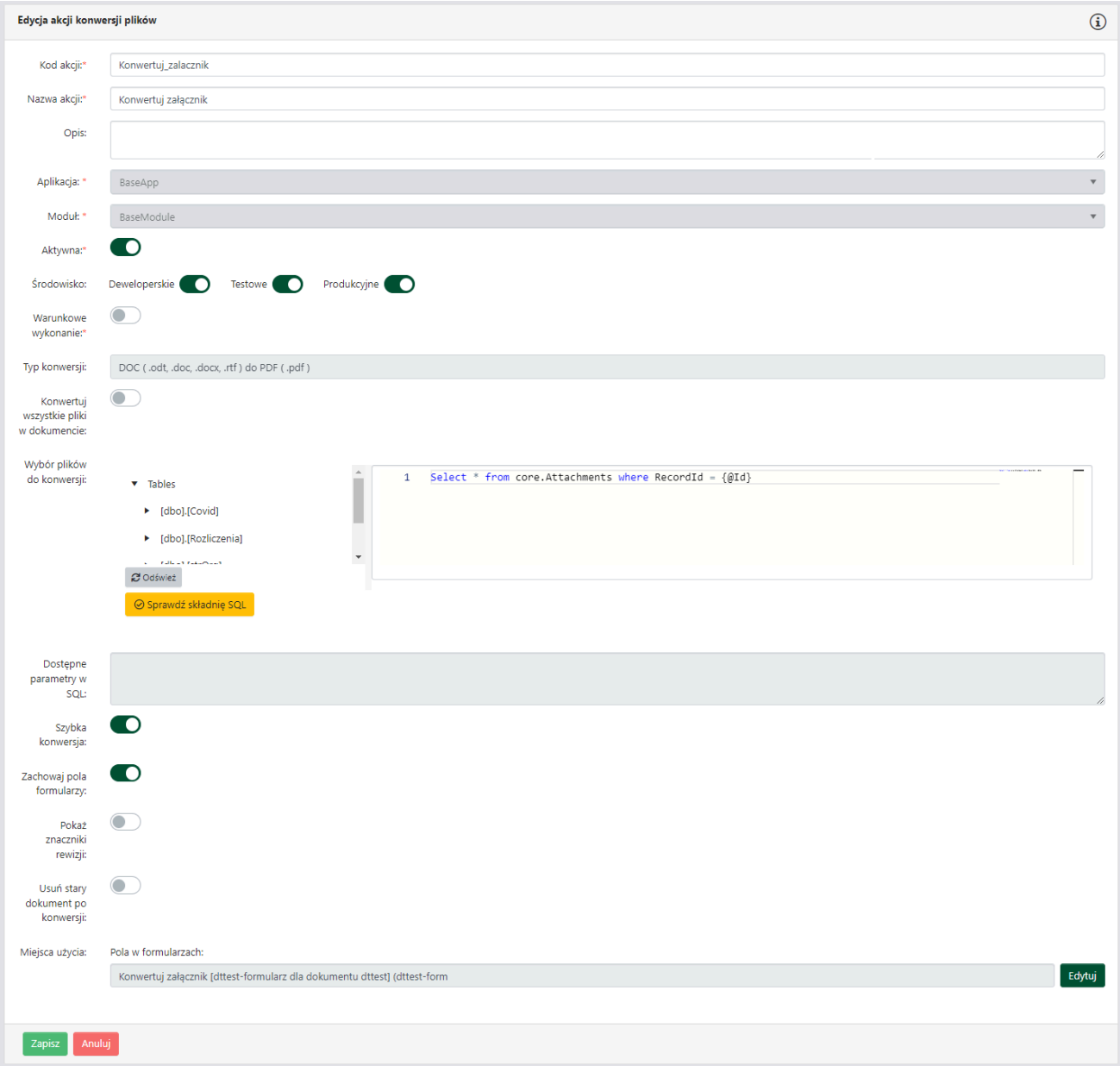
- `Szablon Excel`; kod szablonu zadeklarowanego w menu `NARZĘDZIA` > `Szablony` > `Szablony Excel`.
- `Nazwa arkusza do eksportu`: nazwa arkusza w pliku programu Excel, na którym zostaną zapisane wyeksportowane dane (musi istnieć w szablonie); w przypadku eksportu do nowego pliku może być również wartością zwróconą przez SQL.
- `Komórka początku eksportu`: adres komórki na arkuszu kalkulacyjnym, od której będą dopisywane eksportowane dane (lewa, górna komórka zakresu eksportowanych danych).
- `SQL eksportu`: zapytanie SQL wybierające dane z tabeli w bazie danych, które mają zostać wyeksportowane do arkusza kalkulacyjnego.
- `Niestandardowe typy`: pozwala zdefiniować formatowanie niestandardowych typów danych z tabeli w arkuszu kalkulacyjnym; po kliknięciu przycisku `Nowy format` wpisz nazwę kolumny z tabeli i formatowanie w arkuszu kalkulacyjnym, a następnie kliknij przycisk `Dodaj`, aby zapisać formatowanie; na przykład ustawienie formatu `[$-pl-PL]dddd, d mmmm rrrr` dla kolumny spowoduje, że będą one zapisywane w arkuszu jako data długa (np. *czwartek, 16 grudzień 2023*); pod tabelą znajduje się link do przykładu niestandardowego formatowania danych w programie Excel.

6.1.23. Konwersja Word do PDF

> **Podsumowanie:**
> Ten rodzaj akcji służy do konwertowania załączonych do dokumentu plików w popularnych formatach edytorów tekstu (.odt, .doc, .docx, .rtf) do formatu PDF. Plik wynikowy konwersji jest zapisywany w folderze załączników.

> ☛ *AdminSPA* > `NARZĘDZIA` > `Akcje` > `Konwersja Word do PDF`

Aby możliwe było załączanie plików w formacie edytorów tekstu do dokumentów, należy dopisać odpowiednie rozszerzenia w ustawieniach nAxiom: `Ustawienia` > `Załączniki` > `Dozwolone typy plików`.



Akcja konwersji

Oprócz właściwości wspólnych dla wszystkich akcji, w akcji konwersji należy określić następujące właściwości:

- `Typ konwersji`: rodzaj konwersji; jedyny obsługiwany rodzaj konwersji to `DOC (.odt, .doc, .docx, .rtf) do PDF (.pdf)`.
- `Konwertuj wszystkie pliki w dokumencie`: po włączeniu konwertuje wszystkie załączniki w dokumencie.
- `Wybór plików do konwersji`: zapytanie SQL zwracające rekordy plików z tabeli *core.Attachments*, na przykład:

```
SELECT * FROM core.Attachments WHERE RecordId = {@Id}
```

- `Szybka konwersja`: pozwala szybciej konwertować pliki źródłowe dzięki użyciu metody bezpośredniego renderowania pliku PDF (zamiast renderowania EMF).
- `Zachowaj pola formularzy`: (domyślnie włączone) zachowuje pola formularza w pliku źródłowym jako pola formularza w wynikowym pliku PDF; umożliwia generowanie formularzy do wypełniania w formacie PDF.
- `Pokaż znaczniki rewizji`: zachowuje znaczniki śledzenia zmian z pliku źródłowego w wynikowym pliku PDF.
- `Usuń stary dokument po konwersji`: po wykonaniu konwersji usuwa plik źródłowy.

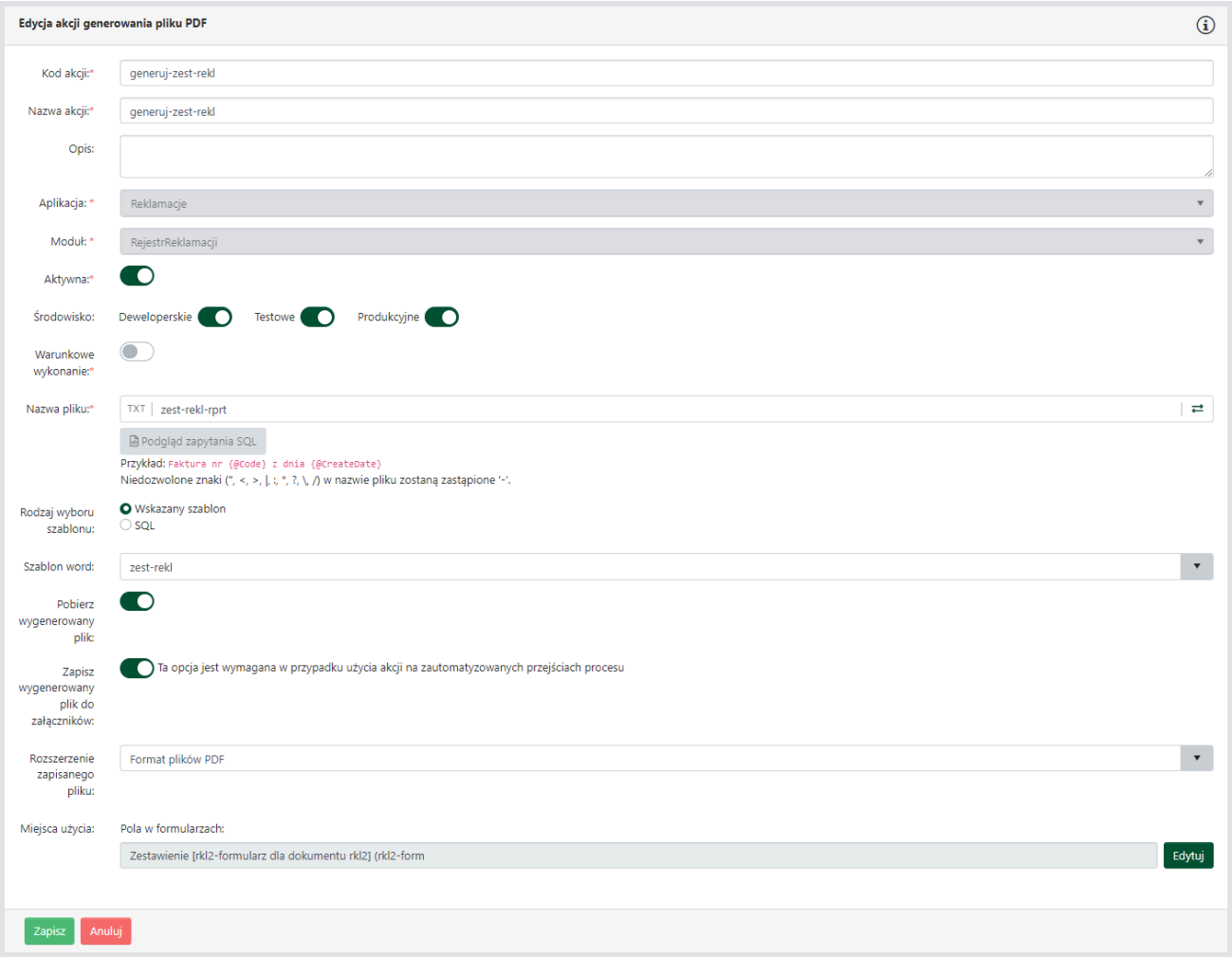
Powiązane tematy:

- Ustawienia systemu, sekcja Załączniki

6.1.24. Generowanie PDF z Word

Podsumowanie:
Akcje generowania pliku PDF służą do generowania pliku PDF na podstawie szablonu zdefiniowanego w programie Word, który jest zasilany danymi z formularza.

☛ *AdminSPA* > `NARZĘDZIA` > `Akcje` > `Generowanie PDF z Word`



Akcja generowania pliku PDF

Oprócz właściwości wspólnych dla wszystkich akcji, w akcji generowania PDF należy określić następujące właściwości:

- `Rodzaj wyboru szablonu`:
  - `Wskazany szablon`: wybierz szablon z listy rozwijanej `Szablon word`.
  - `SQL`: po włączeniu wyświetla standardowe okno zapytanie SQL; zapytanie musi zwrócić zwrócić identyfikator tego szablonu z tabeli *[core].[WordTemplates]*, np.

```
SELECT Id
FROM core.WordTemplates
WHERE Id = {@IdSzablonu}
```

- `Szablon word`: lista rozwijana, z której można wybrać zdefiniowany szablon Word (według kodu); lista jest wyświetlana tylko wtedy, gdy w grupie `Rodzaj wyboru szablonu` zaznaczono opcję `Wskazany szablon`.
- `Pobierz wygenerowany plik`: (domyślnie włączone) powoduje pobranie przez

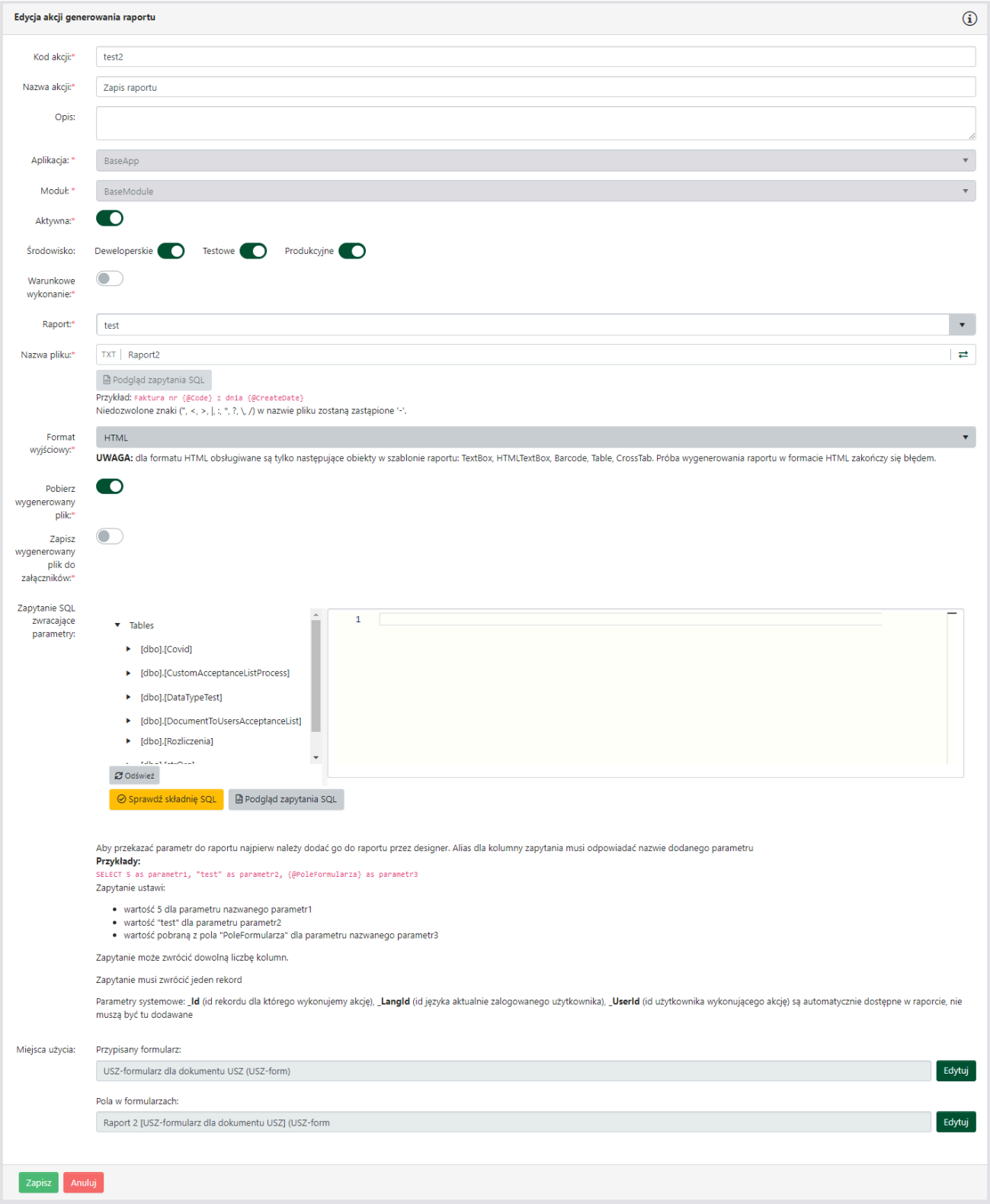
przeglądarkę wygenerowanego pliku.

- `Zapisz wygenerowany plik do załączników`: powoduje zapisanie wygenerowanego pliku jako załącznika do bieżącego dokumentu; w przypadku lokalnej edycji akcji w definicji formularza dostępne są dodatkowo ustawienia:
- `Kategoria załączników`: kategoria, do której ma zostać dodany załącznik; domyślnie `Domyślna kategoria`.
- `Typ zapisu`: tryb dodawania załącznika do dokumentu biznesowego; dostępne wartości:
  - `Zawsze jako nowy plik lub nowa wersja`: jest to jedyne obsługiwane ustawienie, ponieważ obecnie mail monitor obsługuje tylko utworzenie nowego dokumentu biznesowego;
  - `Zawsze jako update poprzedniego pliku lub wersji`: (nieobsługiwane) jeśli dokument ma już załącznik o takiej samej nazwie, nadpisuje go lub nadpisuje wersję, jeśli jest włączona kontrola wersji.
  - `Blokada podczas zapisu`: (nieobsługiwane) jeśli dokument ma już załącznik o takiej samej nazwie, nowy załącznik nie zostanie zapisany.
- `Rozszerzenie zapisanego pliku`: format zapisywanego pliku; dostępne wartości:
  - `Format plików PDF`.
  - `Oryginalny format przesłanego szablonu (plik Word)`.

6.1.25. Raportowanie

Podsumowanie:
Akcje raportów służą do generowania raportów zdefiniowanych w module firmy Telerik w wybranym formacie wyjściowym. W akcji można przekazać do raportu wartości parametrów zdefiniowanych w szablonie raportu

☛ Menu *back-office* > `NARZĘDZIA` > `Akcje` > `Raportowanie`



Akcje raportów Telerik

Oprócz właściwości wspólnych dla wszystkich akcji, w akcji raportów należy określić następujące właściwości:

- `Raport`: lista rozwijana, z której należy wybrać nazwę zdefiniowanego raportu.
- `Nazwa pliku`: nazwa pliku wynikowego; może to być ustalona nazwa lub wartość zwrócona przez zapytanie SQL (wymaga przełączenia typu pola wprowadzania przez

kliknięcie przycisku z prawej strony pola); niedozwolone znaki w zwróconej wartości zostaną zastąpione łącznikami (-).

- `Format wyjściowy`: format pliku wygenerowanego raportu; dostępne wartości to: PDF, CSV, TIFF, RTF, Excel (.xlsx), Word (.docx) i HTML.

> **UWAGA**
> W przypadku eksportu do formatu HTML obsługiwane są tylko następujące obiekty w szablonie raportu: *TextBox*, *HTMLTextBox*, *Barcode*, *Table*, *CrossTab*. Próba wygenerowania raportu zawierającego inne rodzaje obiektów zakończy się błędem.

- `Pobierz wygenerowany plik`: (domyślnie włączone) powoduje pobranie przez przeglądarkę wygenerowanego pliku.
- `Zapisz wygenerowany plik do załączników`: powoduje zapisanie wygenerowanego pliku jako załącznika do bieżącego dokumentu; w przypadku lokalnej edycji akcji w definicji formularza dostępne są dodatkowo ustawienia:
- `Kategoria załączników`: kategoria, do której ma zostać dodany załącznik; domyślnie `Domyślna kategoria`.
- `Typ zapisu`: tryb dodawania załącznika do dokumentu biznesowego; dostępne wartości:
  - `Zawsze jako nowy plik lub nowa wersja`: jest to jedyne obsługiwane ustawienie, ponieważ obecnie mail monitor obsługuje tylko utworzenie nowego dokumentu biznesowego;
  - `Zawsze jako update poprzedniego pliku lub wersji`: (nieobsługiwane) jeśli dokument ma już załącznik o takiej samej nazwie, nadpisuje go lub nadpisuje wersję, jeśli jest włączona kontrola wersji.
  - `Blokada podczas zapisu`: (nieobsługiwane) jeśli dokument ma już załącznik o takiej samej nazwie, nowy załącznik nie zostanie zapisany.
- `Zapytanie SQL zwracające parametry`: zapytanie zwracające wartości parametrów zdefiniowanych w raporcie; aliasy poszczególnych wartości muszą odpowiadać nazwom parametrów w raporcie, np. jeśli w projekcie raportu zdefiniowano parametr ***rpt_param1***, zapytanie:

```
SELECT {@PoleFormularza} AS rpt_param1
```

  przekaże do raportu wartość tego parametru ustawioną na formularzu w polu `PoleFormularza`. W raporcie automatycznie dostępne są wartości parametrów:

  - `_Id`: identyfikator bieżącego dokumentu, w kontekście którego jest wykonywana akcja
  - `_LangId`: identyfikator języka dla zalogowanego użytkownika

- `_UserId`: identyfikator zalogowanego użytkownika

UWAGA
Zapytanie może zwrócić dowolną liczbę kolumn, ale tylko jeden rekord danych.

Powiązane tematy:
- Projektowanie raportów

6.1.26. Skanowanie kodów QR/BC (WebAPI)

Podsumowanie:
Akcja umożliwia odczytanie wartości reprezentowanej przez kod paskowy lub QR za pomocą kamery laptopa lub telefonu w trybie PWA i zapisanie jej w aplikacji nAxiom. Akcja wykorzystuje interfejs API używanej przeglądarki i jest zgodna z większością popularnych przeglądarek. Tabela zgodności znajduje się w oknie definicji akcji.

☛ *AdminSPA* > `NARZĘDZIA` > `Akcje` > `Skaner kodów QR/BC (WebApi)`

Dodawanie akcji skanowania kodów QR/BC (WebApi)

Kod akcji:*
Nazwa akcji:*
Opis:
Aplikacja:*
Moduł:*
Aktywna:*
Środowisko: Deweloperskie Testowe Produkcyjne

Procedura obsługi odpowiedzi

Procedura obsługi odpowiedzi:

Tables
[dbo].[Covid]
[dbo].[CustomAcceptanceListPr
[dbo].[Rozliczenia]
Odśwież

1

Sprawdź składnię SQL | Maksymalizuj edytor

System zapisuje zwracane dane do zmiennej `@_Response`.
Zmienna `@_Response` to obiekt w postaci JSON o następującej strukturze:

```
{
  "code": "1234567"
}
```

Gdzie `code` to treść skanowanego kodu kreskowego / kodu QR.

Rodzaje skanowanych kodów

Rodzaj kodów kreskowych i/lub QR: aztec, code_128, code_39, code_93, codabar, data_matrix, ean_13, ean_8, itf, pdf417, qr_code, upc_a, upc_e

Opcje dodatkowe

Wyświetl przełącznik latarki: *(Opcja niedostępna w przeglądarce Firefox for Android)*

Zawsze włączaj latarkę po rozpoczęciu skanowania i wyłączaj po zakończeniu: *(Opcja niedostępna w przeglądarce Firefox for Android)*

Wyświetl przełącznik urządzeń:

Zgodność z przeglądarką

	Desktop					Mobile						
	Chrome	Edge	Firefox	Opera	Safari	Chrome Android	Firefox for Android	Opera Android	Safari on iOS	Samsung Internet	WebView Android	WebView on iOS
Wersja przeglądarki	✓ 47+	✓ 12+	✓ 33+	✓ 30+	✓ 11+	✓ 47+	✓ 36+	✓ 30+	✓ 11+	✓ 5.0+	✓ 47+	✓ 11+

Zapisz | Anuluj

Akcja skanowania kodów QR/BC

Oprócz właściwości wspólnych dla wszystkich akcji, w akcji skanowania kodów QR/BC należy określić następujące właściwości:

`Procedura obsługi odpowiedzi`: zapytanie SQL, które zapisze w bazie wartość zwróconą przez akcję. Dane odczytane z zeskanowanego kodu są zwracane w zmiennej *@_Response*, która jest obiektem JSON o strukturze:

```
{
  "code": "odczytana wartość kodu QR/BC"
```

```
}
```

Przykładowe zapytanie, które zapisze odczytaną przez akcję wartość klucza *code* w polu bieżącego rekordu:

```
UPDATE [dbo].[DataTypeTest]
SET TestVarchar = (SELECT JSON_VALUE ({@_Response}, '$.code'))
WHERE Id = {@Id}
```

`Rodzaje kodów kreskowych i/lub QR`: lista wielowyboru, w której należy wybrać obsługiwane standardy kodów QR i BC. Domyślnie wybrane są wszystkie pozycje.

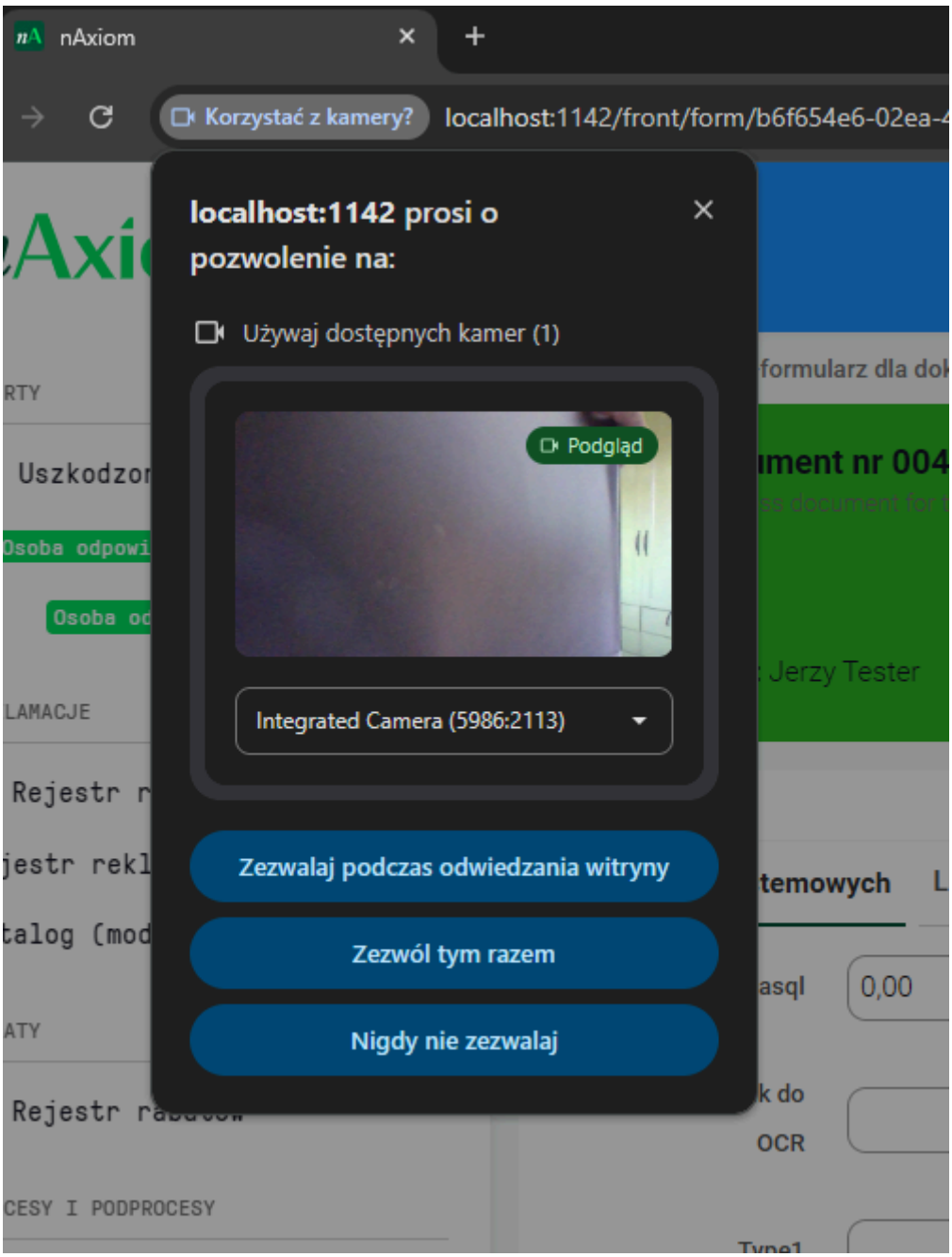
`Wyświetl przełącznik latarki`: wyświetla przełącznik latarki w okienku skanowania kodu.

`Zawsze włączaj latarkę po rozpoczęciu skanowania i wyłączaj po zakończeniu`: umożliwia automatyczną obsługę latarki podczas skanowania.

`Wyświetl przełącznik urządzeń`: wyświetla w okienku skanowania przełącznik kamer dostępnych w urządzeniu.

`Zgodność z przeglądarką`: tabelka zgodności akcji z przeglądarkami na urządzenia mobilne (Android, iOS) i komputery.

Podczas pierwszego użycia konieczne jest przyznanie aplikacji uprawnień do korzystania z kamery.



Przyznanie uprawnień

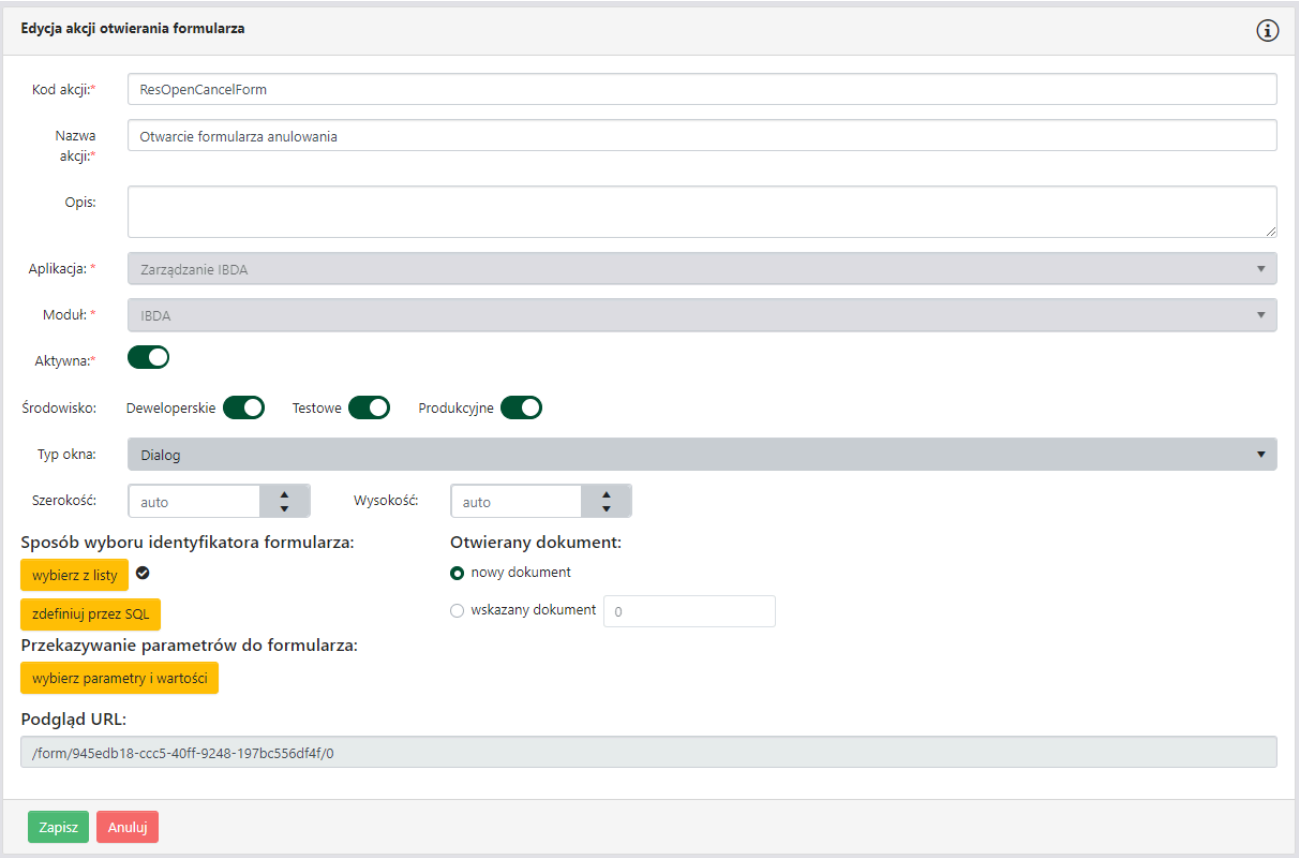
Po uruchomieniu akcja wyświetla okienko, w którym oczekuje kodu do odczytania. Po rozpoznaniu kodu, okienko jest zamykane.

Okienko skanowania

6.1.27. Otwieranie formularza

Podsumowanie:
Akcja otwierania formularza umożliwia otworzenie wskazanego formularza oraz przekazanie parametrów do tego formularza.

☛ *AdminSPA* > `NARZĘDZIA` > `Akcje` > `Otwieranie formularza`



Akcja otwierania formularza

Oprócz właściwości wspólnych dla wszystkich akcji, w akcji otwierania formularza należy określić następujące właściwości:

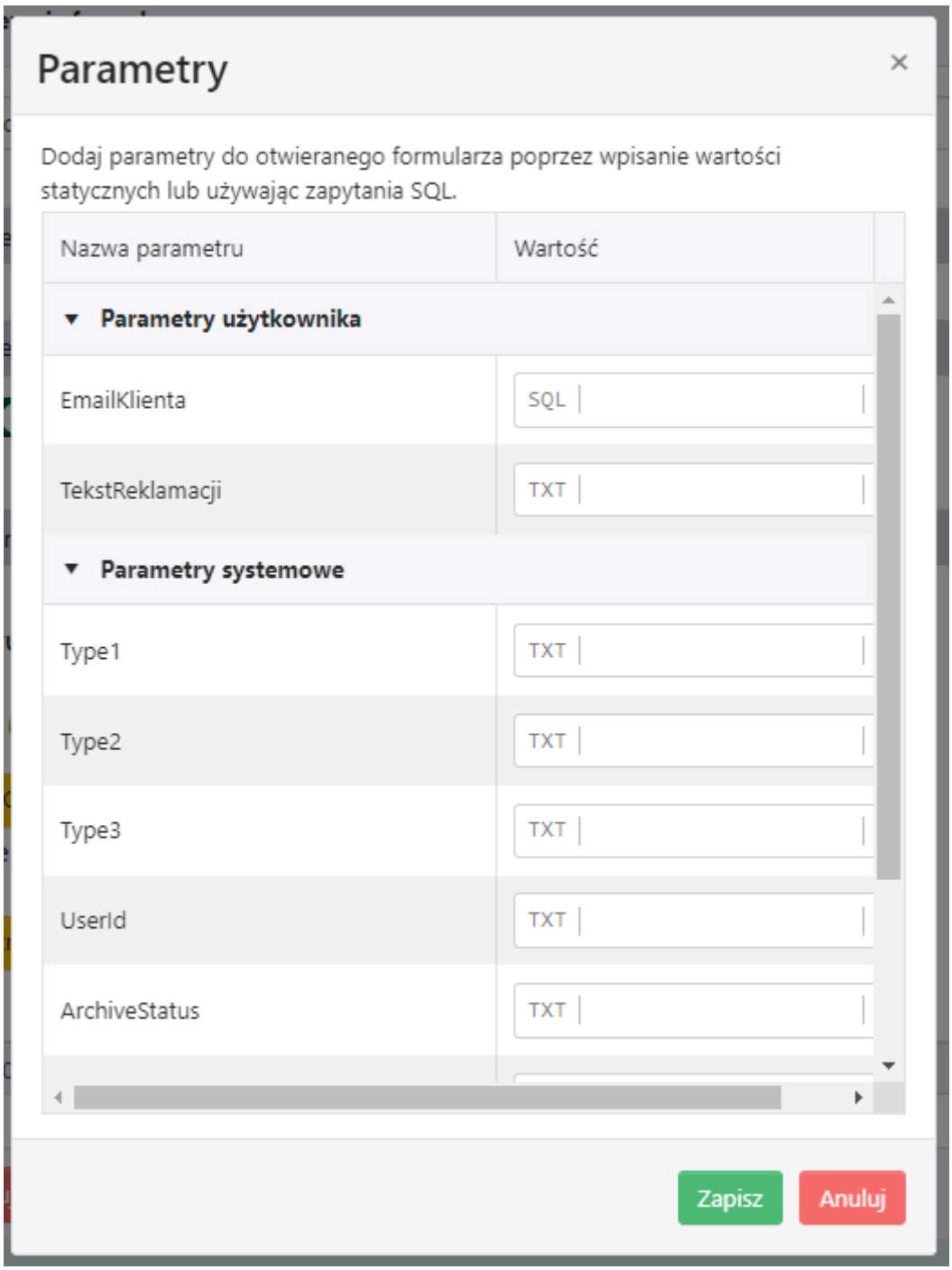
- `Typ okna`: określa rodzaj okna, w którym zostanie wyświetlony formularz:
  - `Dialog`: formularz zostanie otwarty w trybie okna dialogowego (*pop-up*)
  - `Okno`: formularz zostanie otwarty w trybie okna
  - `Link wewnętrzny`: formularz zostanie wyświetlony w trybie linku wewnętrznego
  - `Panel boczny`: formularz zostanie otwarty w trybie panelu bocznego, to jest jego prawa krawędź będzie zadokowana do prawej krawędzi okna przeglądarki; ten tryb jest przeznaczony do wyświetlania formularza przez kliknięcie rekordu w widoku listy. Akcja, w której zostanie wybrany ten tryb, musi zostać przypisana w we właściwości `Wywołaj akcje po interakcji z rekordem listy` na karcie `Ustawienia listy`.
- `Szerokość/Wysokość`: (tryby `Dialog` i `Okno`) określona w pikselach szerokość i wysokość okna formularza.
- `Pełny ekran`: (tryb `Okno`) po włączeniu wyświetla formularz, wykorzystując całe dostępne miejsce w oknie przeglądarki.
- `Szerokość`: (tryb `Panel boczny`) określona w procentach szerokość formularza względem miejsca dostępnego w oknie przegladarki.

- `Sposób wyboru identyfikatora formularza`:
  - `wybierz z listy`: wyświetla okno dialogowe z listą zdefiniowanych formularzy
  - `zdefiniuj przez SQL`: wyświetla okno edycji SQL w celu zdefiniowania zapytania, które zwróci identyfikator GUID formularza
- `Otwierany dokument`:
  - `nowy dokument`: otwiera pusty formularz
  - `wskazany dokument`: wpisz identyfikator dokumentu, który ma zostać otwarty; w polu obsługiwane są zmienne odwołujące się do kontekstu bieżącego formularza *{@NazwaPola}*.
- `Przekazywanie parametrów do formularza`: kliknij przycisk `wybierz parametry i wartości` i w wyświetlonym oknie dialogowym wpisz parametry, czyli wartości, które zostaną wstawione w odpowiednie pola formularza (dostępne są pola użytkownika i pola systemowe); w polach wprowadzania można wpisać stałe wartości, zmienne typu *{@Nazwa}* oraz zapytania SQL zwracające wartości.

UWAGA
Akcja otwierania formularza (w trybie *dialog*) jest wykonywana także podczas odświeżania formularza, który został otwarty przy użyciu tej akcji. Jednak w tym przypadku kontekstem domyślnym jest kontekst formularza docelowego. Dlatego ważne jest odpowiednie skonfigurowanie parametrów odwołujących się do kontekstu formularza źródłowego (tego, z którego jest inicjowana akcja).
Właściwa składnia to *{@*KodFormularza.PoleFormularza}*. Alternatywnie można skorzystać z funkcji
JSON_VALUE({@OpenedForms},'$[0].PoleFormularza')
Należy przy tym pamiętać (w obu przypadkach) o konieczności przestawienia trybu pola wprowadzania na *SQL*.
Zastosowanie zmiennej *{@PoleFormularza}* spowoduje błąd lub nieoczekiwane zachowanie po odświeżeniu formularza docelowego, ponieważ system będzie szukał wartości takiej zmiennej w kontekście odświeżanego formularza.

Zależnie od obiektu źródłowego, z którego wykonywana jest akcja `Otwórz formularz`, odwołania do wartości z tego obiektu powinny wyglądać jak w poniższej tabelce.

Obiekt źródłowy	Odwołanie	Przekazywana wartość
Formularz	*{@PoleFormularza}*	Kontekst formularza źródłowego
Widok listy	*{@KolumnaListy}*	Przecięcie kolumny z wierszem, z którego wywołano akcję
Sekcja Lista	*{@KolumnaListy}* *{@*KodFormularza.PoleFormularza}*	Przecięcie kolumny z wierszem, z którego wywołano akcję Kontekst formularza o podanym kodzie z tablicy *OpenedForms*
Sekcja Harmonogram	*{@KodSekcjiHarmonogramu_SelectedEvent.id}*	Identyfikator zdarzenia harmonogramu, z którego jest otwierany formularz



Parametry akcji otwierania formularza

`Podgląd URL`: wyświetla podgląd linku do otwieranego formularza

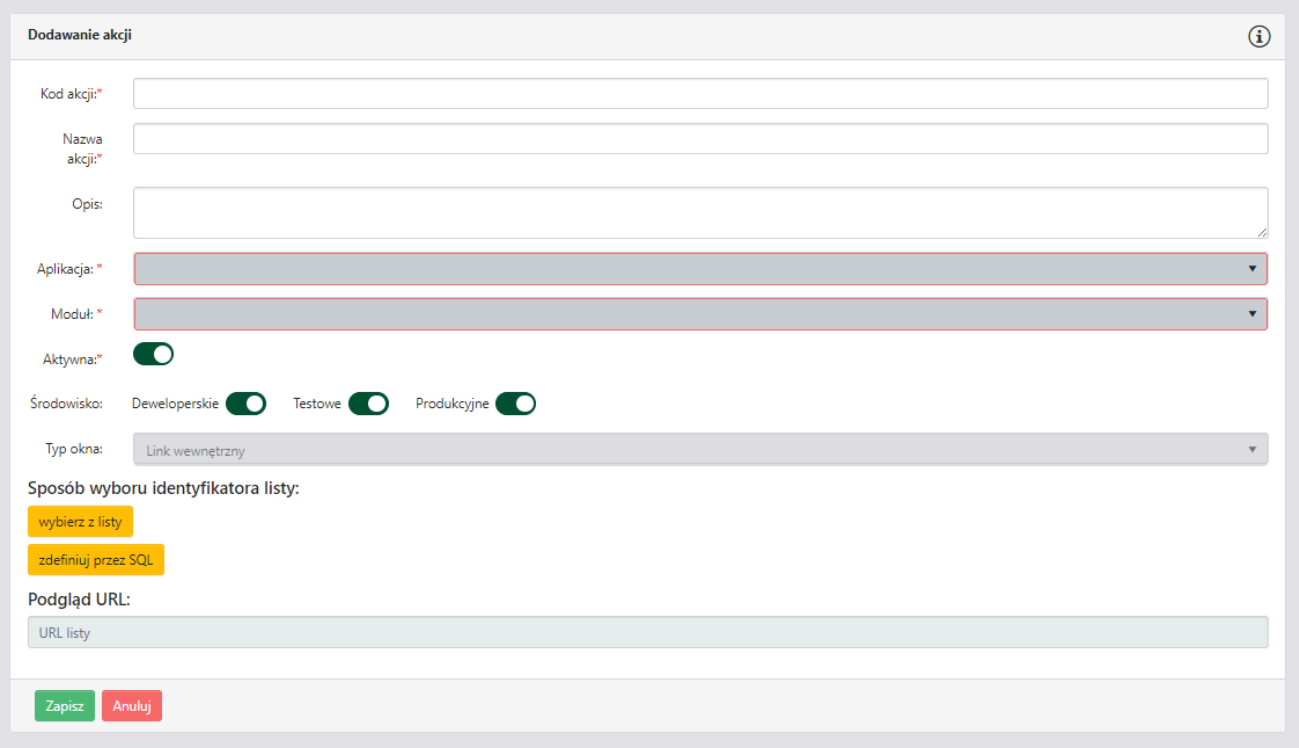
Powiązane tematy:

- Wspólne parametry akcji

6.1.28. Otwieranie listy

Podsumowanie:
Akcje otwierania listy umożliwia wyświetlenie w aplikacji *FrontSPA* wybranego widoku listy.

☛ *AdminSPA* > `NARZĘDZIA` > `Akcje` > `Otwieranie listy`



Akcja otwierania listy

Oprócz właściwości wspólnych dla wszystkich akcji, w akcji otwierania listy należy określić następujące właściwości:

- `Typ okna`: jedyna dostępna wartość to `Link wewnętrzny`.
- `Sposób wyboru identyfikatora listy`:
  - `wybierz z listy`: wyświetla okno dialogowe z wykazem zdefiniowanych widoków listy.
  - `zdefiniuj przez SQL`: wyświetla okno edytora SQL; zapytanie musi zwrócić prawidłowy identyfikator widoku listy do wyświetlenia.
- `Podgląd URL`: (tylko do odczytu) po wskazaniu listy wyświetla podgląd linku, który będzie używany przez akcję.

Powiązane tematy:

- Wspólne parametry akcji
- Otwieranie formularza

6.1.29. Odśwież listę u wszystkich użytkowników

Podsumowanie:
Ta akcja służy do odświeżenia widoku listy wyświetlanego w danej chwili przez wszystkich użytkowników w aplikacjach nAxiom. Skutkiem akcji może być odświeżenie widoku listy lub poinformowanie uzytkownika o konieczności odświeżenia.

Oprócz właściwości wspólnych dla wszystkich akcji, w tej akcji należy określić następujące właściwości:

- `Odświeżana lista`: nazwa listy, której widok ma odświeżyć akcja; aby akcja zadziałała poprawnie, wybrana lista musi mieć zaznaczoną opcję `Automatyczne` na karcie `Odświeżanie listy`; patrz także opis właściwości listy .
- `Tryb automatycznego odświeżania`: dostępne wartości:
  - `Wymuszone`: odświeża widok listy u wszystkich użytkowników,
  - `Informacja o aktualizacji danych`: sygnalizuje konieczność odświeżenia listy poprzez pulsowanie przycisku odświeżania widoku listy.

WSKAZÓWKA
Akcja nie sprawdza, czy dane wyświetlane w widoku listy faktycznie się zmieniły. Do zaimplementowania takiego mechanizmu można wykorzystać opcję warunkowego wykonania akcji.

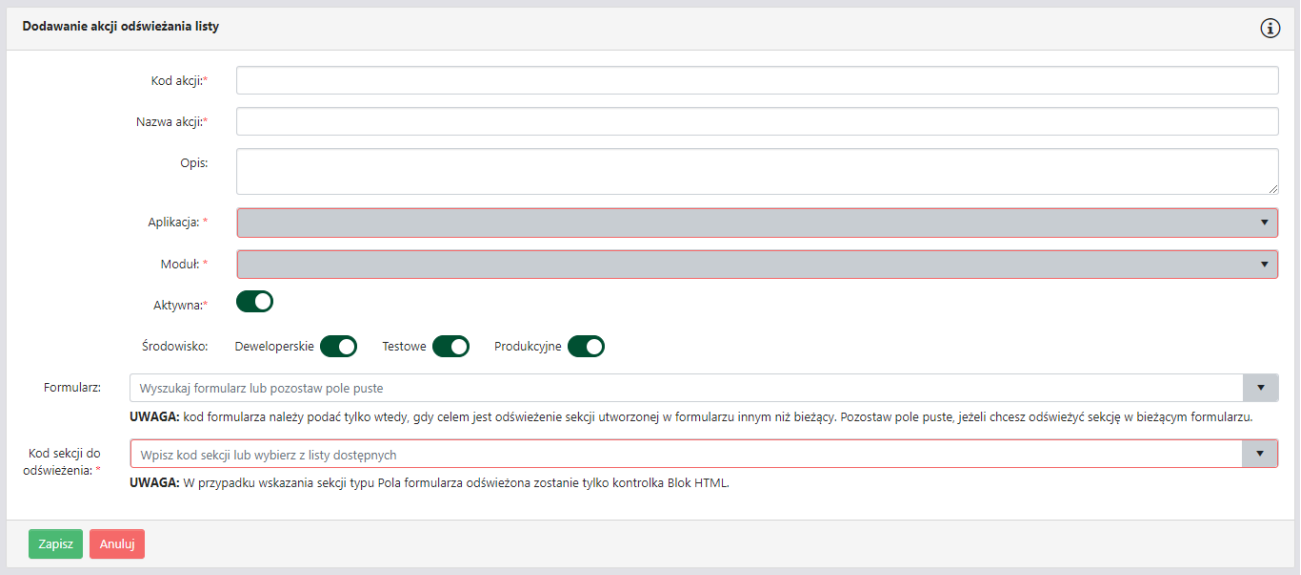
Powiązane tematy:

- Modele decyzyjne DMN

6.1.30. Odświeżanie sekcji formularza

Podsumowanie:
Akcja odświeżania sekcji formularza umożliwia odświeżenie zawartości dowolnej sekcji na formularzu. Odświeżenie polega na ponownym wyrenderowaniu sekcji, w tym ponownym wykonaniu zapytania będącego źródłem danych dla sekcji, aby wyświetlane dane uwzględniały ostatnie zmiany. W przypadku sekcji *pola formularza* odświeżana jest tylko zawartość kontrolki *Blok HTML*.

☛ *AdminSPA* > `NARZĘDZIA` > `Akcje` > `Odświeżanie sekcji formularza`



Akcja odświeżania listy

Oprócz właściwości wspólnych dla wszystkich akcji, w akcji odświeżania listy należy określić następujące właściwości:

`Formularz`: zacznij wpisywać kod formularza i wybierz go z listy.

UWAGA
Jeśli akcja ma odświeżyć sekcję na bieżącym formularzu, w definicji akcji nie należy podawać kodu formularza.

`Kod sekcji do odświeżenia`: wpisz lub wybierz z listy kod sekcji na wybranym formularzu.

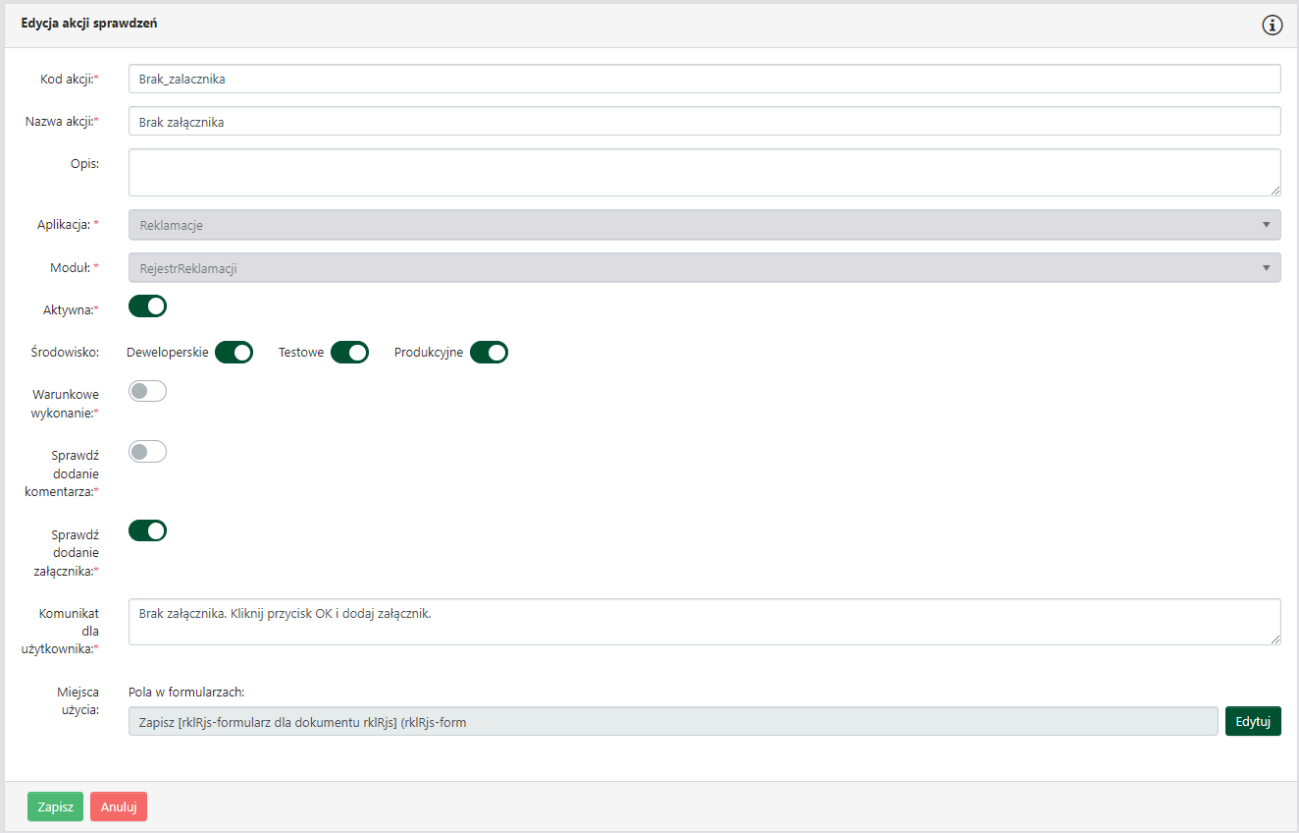
Powiązane tematy:

- Wspólne parametry akcji
- Formularz, sekcja typu lista

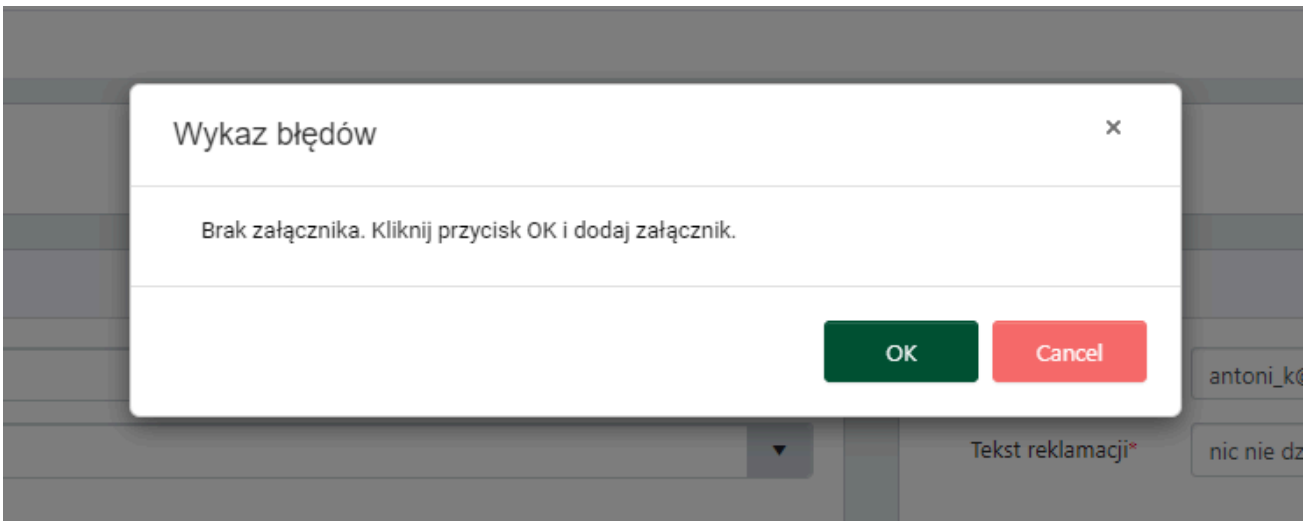
6.1.31. Sprawdzenia przed zapisem

Podsumowanie:
Akcje sprawdzeń umożliwiają sprawdzanie, czy od ostatniej zmiany statusu dokumentu został dodany komentarz lub załącznik. Dla akcji lokalnych jest możliwy wybór kategorii załączników, dla akcji globalnych sprawdzane jest dodanie załącznika w dowolnej kategorii. W razie braku komentarza lub załącznika, akcja powoduje wyświetlenie komunikatu i przerywa operację. Jeśli sprawdzany warunek nie zostanie spełniony, operacja zostanie przerwana i zostanie wyświetlony komunikat zdefiniowany w akcji.

☛ *AdminSPA* > `NARZĘDZIA` > `Akcje` > `Sprawdzenia przed zapisem`



Akcje sprawdzeń



Komunikat o braku załącznika

Oprócz właściwości wspólnych dla wszystkich akcji, w akcji sprawdzeń należy określić następujące właściwości:

- `Sprawdź dodanie komentarza`: po włączeniu sprawdza, czy od ostatniej zmiany statusu do dokumentu dodano komentarz.
- `Sprawdź dodanie załącznika`: po włączeniu sprawdza, czy od ostatniej zmiany statusu do dokumentu dodano załącznik.
- `Komunikat dla użytkownika`: tekst komunikatu wyświetlanego w przypadku, gdy co najmniej jeden z warunków sprawdzania nie zostanie spełniony (brak komentarza lub brak załącznika).

Powiązane tematy:

- Wspólne parametry akcji

6.1.32. Wysyłanie powiadomień

Podsumowanie:
Akcja powiadomień umożliwia konfigurowanie powiadomień wysyłanych do wybranych użytkowników aplikacji nAxiom. Elementy akcji wymagające zdefiniowania to:

- odbiorcy powiadomienia; za pomocą zapytania SQL,
- nagłówek powiadomienia; jako tekst lub zapytanie SQL; w nagłówku można odwoływać się do zmiennych *{@PoleFormularza}*,
- treść powiadomienia; jako tekst z markupem HTML z możliwością wstawiania w tekście wyników zapytania SQL; w treści można także używać linków do dokumentów w widoku formularza lub do widoku listy.

☛ *AdminSPA* > NARZĘDZIA > Akcje > Wysyłanie powiadomień

Edycja akcji powiadomień

Kod akcji:* Zmiana_statusu_wniosku

Nazwa akcji:* Zmiana statusu wniosku

Opis:

Aplikacja: * BaseApp

Moduł: * BaseModule

Aktywna:*

Środowisko: Deweloperskie Testowe Produkcyjne

Warunkowe wykonanie:*

Odbiorcy:* **Przykład:** `SELECT [Id] FROM [core].[UserProfiles] WHERE FirstName = 'Anna'`

Tables
- [dbo].[Covid]
- [dbo].[CustomAcceptanceListProc
- [dbo].[DataTypeTest]
- [dbo].[DocumentToUsersAcceptan

Odśwież

```
1  SELECT ase.[ACLId]
2        ,[UserProfileId] -- identyfikator odbiorcy powiadomienia
3  FROM [core].[ACLStakeholderEntries] AS ase
4  WHERE ase.StakeholderId = 2 -- identyfikator kategorii interesariusza Benefic
5  AND ase.ACLId =
6     (SELECT ACLId
7      FROM [dbo].[Wnioski]
8      WHERE Id = {@Id}) -- identyfikator bieżącego dokumentu
```

Sprawdź składnię SQL | Sprawdź zapytanie | Podgląd zapytania SQL

Ikona powiadomienia: * far fa-lig

Nagłówek:* TXT Wniosek nr {@Code} zmienił status na {@Status}

Podgląd zapytania SQL

Przykład: `Faktura nr {@Code} z dnia {@CreateDate}`

Treść:* Aby wstawić zwrócone wartości do treści powiadomienia, należy użyć ich wewnątrz RowTemplate:

`<RowTemplate> {@Query_NazwaKolumny} </RowTemplate>`

gdzie *NazwaKolumny* to nazwa (lub alias) kolumny zwróconej przez zapytanie SQL. Taki RowTemplate zostanie wygenerowany dla każdego rekordu które zapytanie SQL dla treści zwróci.

Przykład: `<RowTemplate><p>Twój wniosek z dn. {@Query_DataWniosku} zmienił status. Kliknij <a href="{#Link/form/222408f5.../{@Query_IdDokumentu}}">tutaj</a>, aby przejść do wniosku.</p></RowTemplate>>`

Aby dodać link i ręcznie użyć zmiennych kliknij przycisk: Wklej link

Aby dodać link i wybrać istniejący formularz/listę kliknij przycisk: Utwórz link

```
1  <p>Twój wniosek z dn. <RowTemplate>{@Query.DataWniosku}</RowTemplate> zmienił status. Kliknij
2  <a href="{#Link/form/222408f5/<RowTemplate>{@Query.IdDokumentu}</RowTemplate>}">tutaj</a>,
3  aby przejść do wniosku.</p>
```

SQL dla treści:* **Przykład:** `SELECT Id AS IdDokumentu, CreateDate AS DataWniosku FROM dbo.Wnioski WHERE Id = {@Id}`

Tables
- [dbo].[Covid]
- [dbo].[CustomAcceptanceListProc

Odśwież

```
1  SELECT Id AS IdDokumentu
2        ,CreateDate AS DataWniosku
3  FROM dbo.Wnioski
4  WHERE Id = {@Id}
```

Sprawdź składnię SQL | Sprawdź zapytanie | Podgląd zapytania SQL

Dostępne parametry w SQL:

Zapisz | Anuluj

Akcja powiadomień

Oprócz właściwości wspólnych dla wszystkich akcji, w akcji powiadomień należy określić następujące właściwości:

- `Odbiorcy`: zapytanie SQL zwracające identyfikatory GUID użytkowników, którzy mają otrzymać powiadomienie.
- `Ikona powiadomienia`: wybierz ikonę wyświetlaną w nagłówku powiadomienia; kliknij przycisk z ikoną lupy, aby wyświetlić okno dialogowe z ikonami do wybrania; wpisz słowo kluczowe (w języku angielskim), aby znaleźć ikonę powiązaną z tym hasłem.
- `Nagłówek`: tekst wyświetlany w nagłówku powiadomienia; może zawierać zmienne *{@PoleFormularza}* z kontekstu formularza; może być zwracany przez zapytanie SQL (kliknij napis `TXT` z lewej stron pola, aby zmienić go na `SQL`).
- `Treść`: treść powiadomienia; wpisuj treść, począwszy od wiersza z numerem 1; w treści możesz używać linków do dokumentów w widoku formularza i/lub do widoków list:
  - `Wklej link`: w wierszu treści wkleja ogólny format linku *{#Link/TYP/ID/RECORDID}*, który należy samodzielnie zmodyfikować.
  - `Utwórz link`: wyświetla okno wyboru formularza/widoku listy, w celu utworzenia linku do wskazanego obiektu.
- `SQL dla treści`: zwraca wartości, które mają zostać użyte w treści powiadomienia; aby wstawić zwrócone wartości do treści powiadomienia, należy użyć następującej składni:

```
<RowTemplate>
    {@Query.NazwaKolumny}
</RowTemplate>
```

  gdzie *NazwaKolumny* to nazwa (lub alias) kolumny zwróconej przez zapytanie SQL.

- **Kolory elementów powiadomienia**:
  - `Kolory ikony i nagłówka`: `Własne` lub `Wg kreatora stylów`
  - `Kolory czcionki dla linku w treści`: `Własne` lub `Wg kreatora stylów`
  - `Kolory tła i ramki kafla`: `Własne` lub `Wg kreatora stylów`

PRZYKŁAD

Powiadomienie beneficjenta (interesariusz *Beneficjent*) o zmianie statusu wniosku i przesłanie mu linku do wniosku.

Odbiorca:

```
SELECT ase.[ACLId]
      ,[UserProfileId] -- identyfikator odbiorcy powiadomienia
FROM [core].[ACLStakeholderEntries] AS ase
WHERE ase.StakeholderId = 2 -- identyfikator kategorii interesariusza Beneficjent
AND ase.ACLId =
   (SELECT ACLId
    FROM [dbo].[Wnioski]
        WHERE Id = {@Id}) -- identyfikator bieżącego dokumentu
```

Nagłówek:

```
Wniosek nr {@Code} zmienił status na {@Status}
```

Treść:

```
<p>Twój wniosek z dn. <RowTemplate>{@Query.DataWniosku}</RowTemplate>
zmienił status. Kliknij
<a href="{#Link/form/222408f5.../<RowTemplate>{@Query.IdDokumentu}</RowTemplate>}">
  tutaj
</a>, aby przejść do wniosku.</p>
```

Zapytanie SQL dla treści:

```
SELECT Id AS IdDokumentu
      ,CreateDate AS DataWniosku
FROM dbo.Wnioski
WHERE Id = {@Id}
```

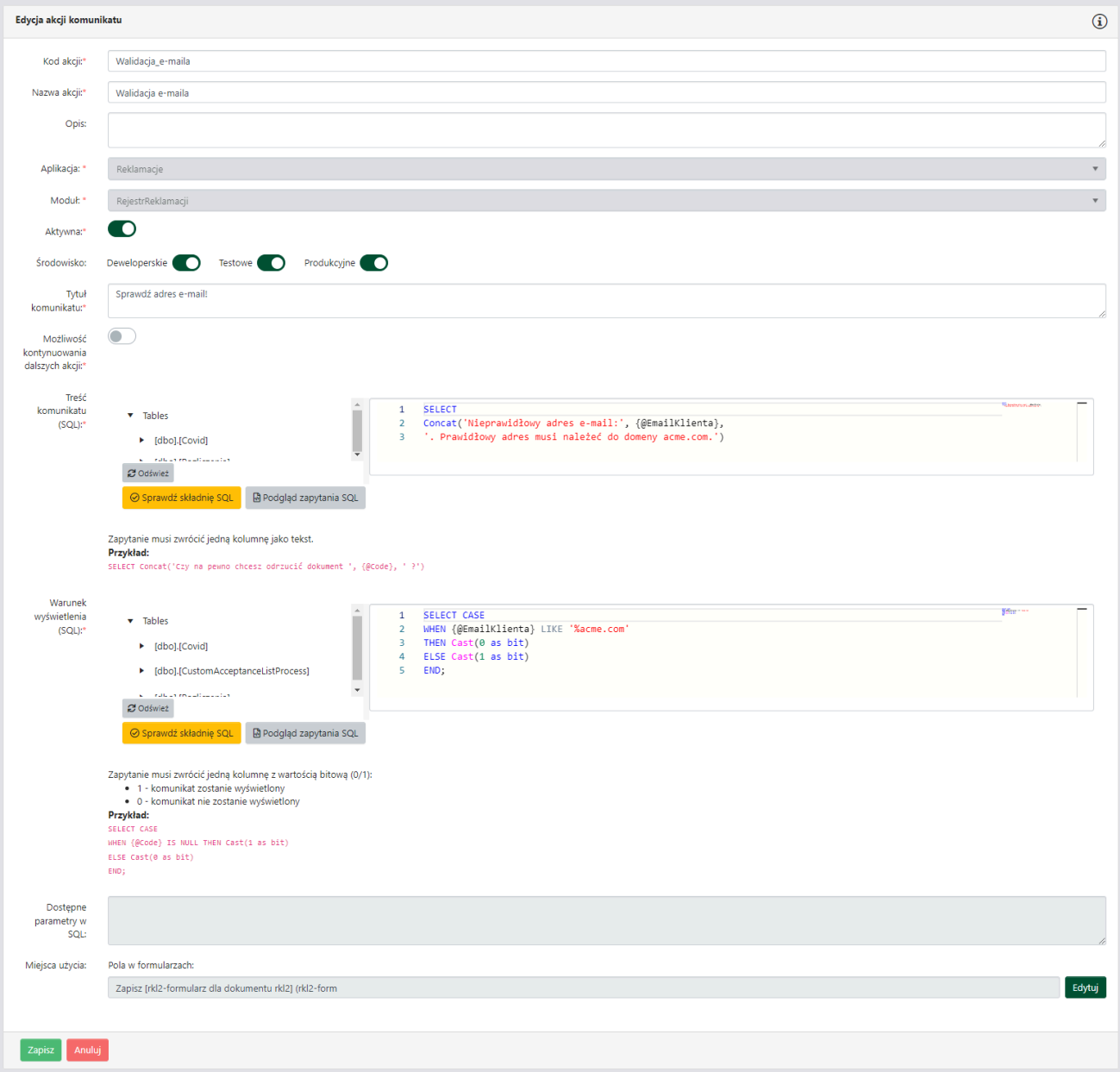
Powiązane tematy:

- Wspólne parametry akcji

6.1.33. Wyświetlanie komunikatów

Podsumowanie:
Akcje komunikatów umożliwiają definiowanie komunikatów, które są wyświetlane po spełnieniu określonego warunku. Dodatkowo dla akcji komunikatu można określić, czy wyświetlenie komunikatu przerywa bieg procesu (błąd), czy nie (informacja). W komunikatach można używać wartości z formularza, podobnie jak w kryteriach wyświetlenia komunikatu. W obu przypadkach w zapytaniu należy użyć standardowej składni *{@PoleFormularza}*.

☛ *AdminSPA* > `NARZĘDZIA` > `Akcje` > `Wyświetlanie komunikatów`



Dodawanie/Edycja akcji komunikatu

Oprócz właściwości wspólnych dla wszystkich akcji, w akcji komunikatów należy określić następujące właściwości:

- `Tytuł komunikatu`: tekst wyświetlany na pasku tytułu wyskakującego okienka komunikatu.
- `Możliwość kontynuowania dalszych akcji`: (domyślnie wyłączone) po włączeniu wykonywane są kolejne akcje (komunikat informacyjny/ostrzegawczy z możliwością kontynuacji).
- `Treść komunikatu (SQL)`: zapytanie SQL zwracające treść komunikatu.
- `Warunek wyświetlenia (SQL)`: zapytanie SQL, które musi zwrócić wartość *1* (typu *bit*), aby komunikat został wyświetlony; w przypadku wartości *0* komunikat nie zostanie wyświetlony.

PRZYKŁAD

Treść komunikatu:

```
SELECT
Concat('Nieprawidłowy adres e-mail:', {@EmailKlienta},
'. Prawidłowy adres musi należeć do domeny acme.com.')
```

Warunek wyświetlenia:

```
SELECT CASE
WHEN {@EmailKlienta} LIKE '%acme.com'
THEN Cast(0 as bit)
ELSE Cast(1 as bit)
END;
```

Po skonfigurowaniu akcji komunikatu jak w przytoczonym przykładzie, próba zapisania rekordu z nieprawidłowym adresem e-mail nie powiedzie się i zostanie wyświetlony komunikat o błędzie.

Komunikat o błędzie wyświetlony przez akcję komunikatu

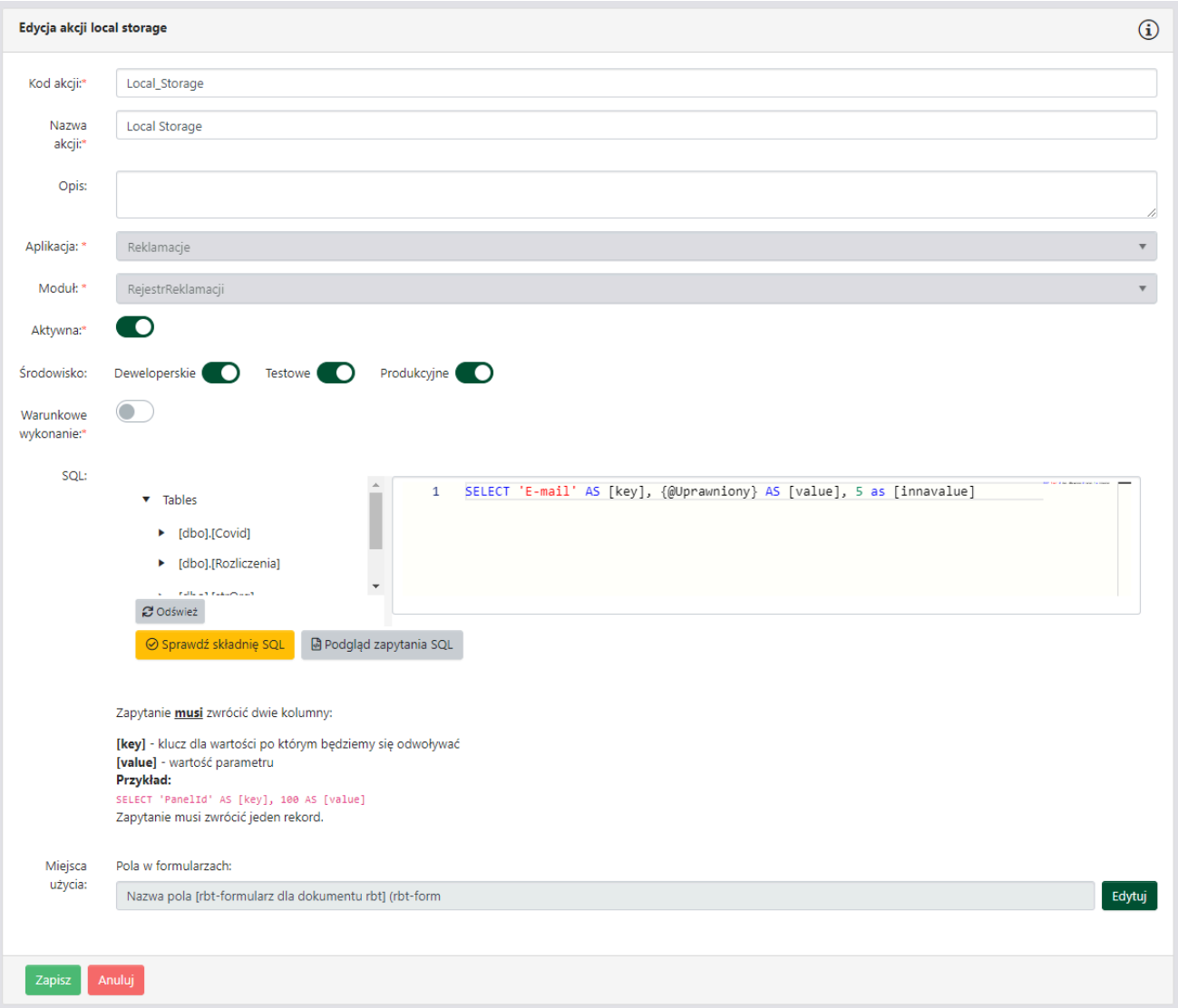
Powiązane tematy:

- Wspólne parametry akcji

6.1.34. Magazyn Local Storage

Podsumowanie:
Akcje local storage służą do zapisywania pary *key-value* w pamięci *local storage* przeglądarki. Para *key-value* musi zostać zwrócona przez zapytanie SQL. Do wartości zapisanej w ten sposób można się odwołać, używając zmiennej *{@sys_key}*, gdzie *key* to wartość w kolumnie *key* zwracana przez zapytanie, np. *{@sys_E-mail}*.

☛ *AdminSPA* > `NARZĘDZIA` > `Akcje` > `Magazyn Local Storage`



Akcja local storage

Oprócz właściwości wspólnych dla wszystkich akcji, w akcji local storage należy określić następujące właściwości:

- `SQL`: zapytanie zwracające kolumny *key* i *value*, których wartości zostaną zapisane w local storage.

> **UWAGA**
> Zapytanie musi zwrócić obie kolumny i tylko jeden rekord danych.

6.1.35. Szyfrowanie plików

> **Podsumowanie:**
> Akcja szyfrowania służy do szyfrowania załączników danego dokumentu biznesowego. Akcje obsługują tylko załączniki w formacie PDF.

☛ *AdminSPA* > `NARZĘDZIA` > `Akcje` > `Szyfrowanie plików`

Edycja akcji szyfrowania plików

Kod akcji:* Szyfruj_zalacznik

Nazwa akcji:* Szyfruj załącznik

Opis:

Aplikacja: * BaseApp

Moduł: * BaseModule

Aktywna:*

Środowisko: Deweloperskie Testowe Produkcyjne

Warunkowe wykonanie:*

Akcje szyfrowania plików: PDF (.pdf)

Konwertuj wszystkie pliki w dokumencie:

SQL dla hasła właściciela:

Tables
[dbo].[Covid]
[dbo].[Rozliczenia]
Odśwież
Sprawdź składnię SQL

```
1 SELECT 'owner'
```

Dostępne parametry w SQL:

SQL dla hasła użytkownika:

Tables
[dbo].[Covid]
[dbo].[Rozliczenia]
Odśwież
Sprawdź składnię SQL

```
1 SELECT 'user'
```

Dostępne parametry w SQL:

Kopiowanie treści:

Składanie dokumentów:

Zarządzanie adnotacjami:

Modyfikacja treści:

Wypełnianie pól:

Drukowanie:

Miejsca użycia: Pola w formularzach:

Szyfruj załącznik [dttest-formularz dla dokumentu dttest] (dttest-form Edytuj

Zapisz Anuluj

Akcja szyfrowania

Indywidualne właściwości akcji szyfrowania to:

`Akcje szyfrowania plików`: format pliku do szyfrowania; obecnie jedyny obsługiwany format plików to `PDF (.pdf)`.

`Konwertuj wszystkie pliki w dokumencie`: po włączeniu szyfruje wszystkie załączniki w dokumencie; jeśli niektóre załączniki zostały już zaszyfrowane, nie będą szyfrowane ponownie.

`Wybór plików do szyfrowania`: (przełącznik dostępny tylko, gdy przełącznik `Konwertuj wszystkie pliki w dokumencie` jest wyłączony) zapytanie SQL zwracające rekordy załączników z tabeli *core.Attachments*, na przykład:

```
SELECT * FROM core.Attachments WHERE RecordId = {@Id}
```

`SQL dla hasła właściciela`: instrukcja SQL zwracająca ciąg znaków, który zostanie użyty jako hasło dla właściciela dokumentu.

`SQL dla hasła użytkownika`: instrukcja SQL zwracająca ciąg znaków, który zostanie użyty jako hasło dla użytkownika dokumentu.

`Kopiowanie treści`: po włączeniu blokuje możliwość kopiowania tekstu z zaszyfrowanego pliku.

`Składanie dokumentów`: po włączeniu blokuje możliwość użycia bieżącego pliku PDF jako części składowej innego pliku PDF.

`Zarządzanie adnotacjami`: po włączeniu blokuje możliwość zarządzania adnotacjami w pliku PDF.

`Modyfikacja treści`: po włączeniu blokuje możliwość modyfikowania treści w zaszyfrowanym pliku PDF.

`Wypełnianie pól`: po włączeniu blokuje możliwość wypełniania pól w zaszyfrowanym pliku PDF.

`Drukowanie`: po włączeniu blokuje możliwość wydrukowania zaszyfrowanego pliku PDF.

Powiązane tematy:

- Wspólne parametry akcji

6.1.36. Drukowanie ZPL

Podsumowanie:
Akcje drukowania służą do drukowania etykiet z kodem kreskowym na drukarkach termicznych obsługujących język skryptowy ZPL. Akcja drukowania wymaga określenia zapytania SQL, które zwróci dane do wydrukowania na etykiecie (nie tylko w formie kodu kreskowego, ale także w formie tekstowej) oraz skrypt w języku ZPL, który zostanie wysłany na drukarkę. Aby w skrypcie osadzić dane zwrócone przez SQL, należy użyć notacji ***{@NazwaKolumny}***, gdzie *NazwaKolumny* to nazwa kolumny zwróconej przez zapytanie SQL.

☛ *AdminSPA* > `NARZĘDZIA` > `Akcje` > `Drukowanie ZPL`

INFO
Jeśli zapytanie zwróci zestaw rekordów, akcja wyśle je wszystkie na drukarkę, co pozwala drukować etykiety w partiach.

Edycja akcji wydruku

Kod akcji:* zpl_label_print

Nazwa akcji:* Wydruk etykiet

Opis:

Aplikacja: * BaseApp

Moduł: * BaseModule

Aktywna:*

Środowisko: Deweloperskie Testowe Produkcyjne

Warunkowe wykonanie:*

SQL pobieranych danych

- Tables
  - [dbo].[Covid]
  - [dbo].[DataTypeTest]
  - [dbo].[TabelaProcesGlowny]

Odśwież

```
SELECT UserId, Code, EmailKlienta, FirstName, LastName
FROM [dbo].[RejestrReklamacji] rr
INNER JOIN core.UserProfiles up ON rr.UserId = up.Id
WHERE rr.UserId = '18EF4E5E-11EE-41CD-5A9D-08D8DA52C48B'
```

Sprawdź składnię SQL | Sprawdź parametry

Dostępne parametry w SQL:

{@_Id},{@_FormId},{@_GridId},{@_BusinessDocDefId},{@_UserId},{@_LangId},{@_MobileMode},{@Transition},{@Status},{@Type1},{@Type2},{@Type3},{@Id},{@Code},{@Organization},{@RowVersion},{@UserId},{@CreateDate},{@ACLId},{@EmailKlienta},{@TekstReklamacji},{@UpdatedBy},{@UpdatedDate},{@ArchiveStatus},{@Blocked},{@BlockUser},{@LastEntryDate},{@IsAnonimized},{@FormId},{@FormCode},{@BusinessDocId},{@AppId}

Skrypt drukowania

```
^XA

^FX Top section with logo, name and address.
^CF0,60
^FO50,50^GB100,100,100^FS
^FO75,75^FR^GB100,100,100^FS
^FO93,93^GB40,40,40^FS
^FO220,50^FDClaim no. {@Code}^FS
^CF0,30
^FO220,115^FDReporter: {@EmailKlienta}^FS
^FO220,155^FDResponsible: {@FirstName} {@LastName}^FS

^FX Third section with bar code.
```

Miejsca użycia:

Przypisany formularz:

rkl2-formularz dla dokumentu rkl2 (rkl2-form) Edytuj

Pola w formularzach:

Kod kreskowy [rkl2-formularz dla dokumentu rkl2] (rkl2-form Edytuj

Zapisz Anuluj

Edycja akcji wydruku

Wydruk etykiety

W celu zdefiniowania akcji określ następujące właściwości:

`SQL pobieranych danych`: zapytanie SQL zwracające dane do skryptu ZPL.

`Skrypt drukowania`: skrypt ZPL definiujący wydruk etykiety; każdy wiersz skryptu zaczyna się od znaku ^.

> **WAŻNE**
> Drukowanie powiedzie się tylko wtedy, kiedy użytkownik uruchamiający akcję będzie miał przypisaną w swoim profilu drukarkę ZPL jako drukarkę domyślną.

Do testów można wykorzystać aplikację *Virtual ZPL Printer*, do pobrania: https://github.com/porrey/Virtual-ZPL-Printer). Aplikacja wymaga środowiska .NET Runtime 7.0, do pobrania: https://dotnet.microsoft.com/en-us/download/dotnet/thank-you/runtime-desktop-7.0.1-windows-x64-installer. Po zainstalowaniu należy skonfigurować adres IP i port, i zainstalować jako drukarkę. Opis procedury jest dostępny na stronie: https://intercom.help/asgard-alliance-inc/en/articles/6461275-setting-up-a-virtual-printer.

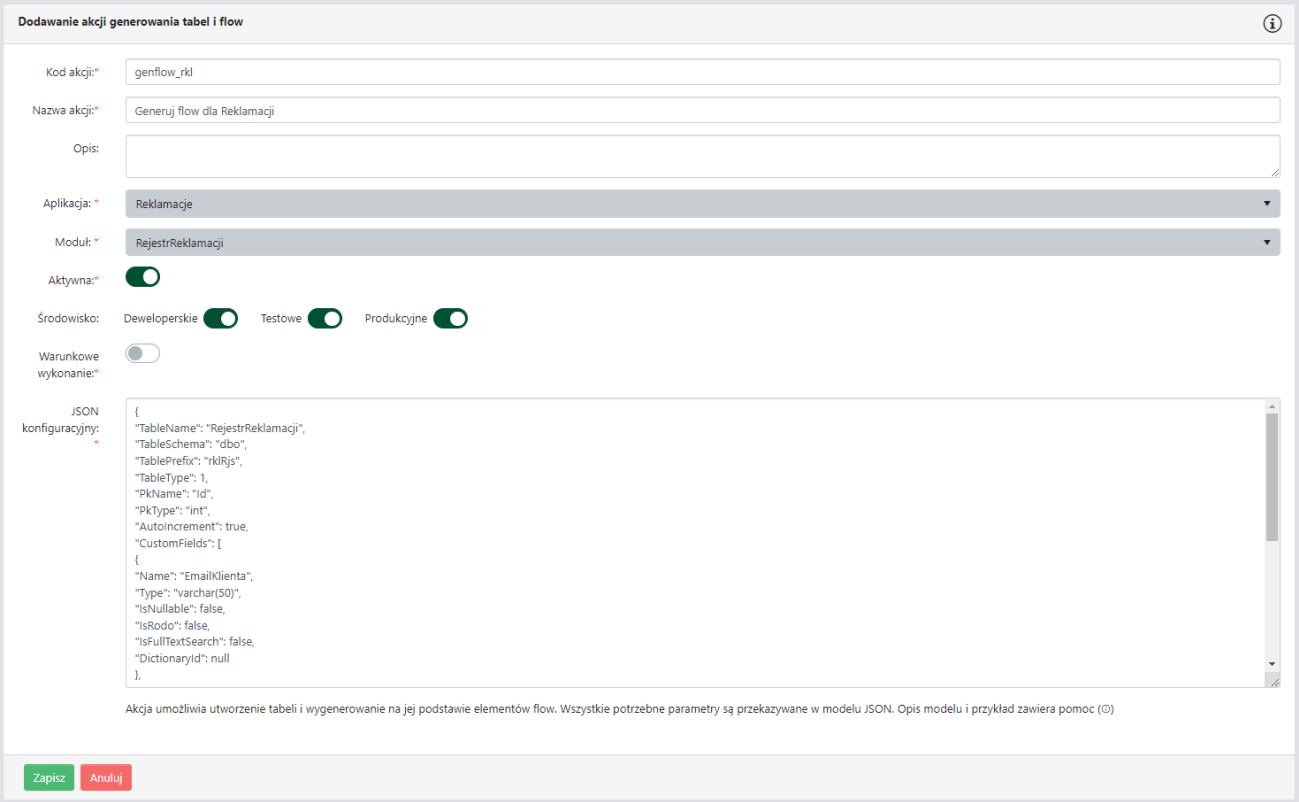
Powiązane tematy:

- Ustawienia profilu użytkownika - Drukarka

6.1.37. Generowanie elementów aplikacji

Podsumowanie:
Akcja generowania tabeli i elementów aplikacji umożliwia zautomatyzowane utworzenie tabeli i wygenerowanie na jej podstawie elementów aplikacji, czyli modelu procesu biznesowego. Generowanie elementów aplikacji jest opcjonalne, a generowane obiekty zależą od wybranego rodzaju tabeli. Wszystkie potrzebne parametry są przekazywane w formacie JSON.

☛ *AdminSPA* > `NARZĘDZIA` > `Akcje` > `Generowanie elementów aplikacji`



Akcja generowania elementów aplikacji

Oprócz właściwości wspólnych dla wszystkich akcji, w akcji generowania elementów aplikacji należy określić następujące właściwości:

`JSON konfiguracyjny`: parametry potrzebne do utworzenia tabeli oraz opcjonalnie wygenerowania flow dla tej tabeli; poszczególne parametry opisano poniżej.

Model JSON dla akcji tworzenia tabeli i flow

Klucz	Opis	Dozwolone wartości
TableName	Nazwa tworzonej tabeli	Dowolna nazwa tabeli
TableSchema	Schemat tabeli w bazie danych	Zwykle dla tabel użytkownika jest używany schemat *dbo*
TablePrefix	Przedrostek dla elementów generowanych na podstawie tabeli	Dowolna wartość (tylko znaki alfanumeryczne)
TableType	Typ tworzonej tabeli	0: tabela pomocnicza (SimpleTable) 1: tabela modelu standardowego (BusinessTable) 2: tabela modelu uproszczonego (CatalogTable)
PkName	Nazwa kolumny klucza głównego	Dowolna nazwa; tylko znaki alfanumeryczne i podkreślenie; pierwsza litera musi być wielka
PkType	Typ kolumny klucza głównego	Typ danych kolumny klucza głównego; zwykle numeryczny *int* lub znakowy *uniqueidentifier*
AutoIncrement	Automatyczna inkrementacja wartości	*true* lub *false*
CustomFields	Literał JSON definiujący poszczególne kolumny tabeli	
Name	Nazwa kolumny	Dowolna nazwa; tylko znaki alfanumeryczne i podkreślenie; pierwsza litera musi być wielka
Type	Typ danych kolumny	Dowolny typ danych przez SQL Server
IsNullable	Dozwolone wartości puste	*true* lub *false*
IsRodo	Czy zawiera dane	*true* lub *false*

Model JSON dla akcji tworzenia tabeli i flow

Klucz	Opis	Dozwolone wartości
	RODO	
IsFullTextSearch	Czy dane z tej kolumny są indeksowane do wyszukiwania pełnotekstowego	*true* lub *false*
DictionaryId	Identyfikator słownika	Identyfikator słownika zdefiniowanego w nAxiom, z którego wartości będą dostępne w tej kolumnie (dotyczy tylko kolumn typu *int*); jeśli to nie jest kolumna słownikowa, wartość *null*
RunFlow	Określa czy akcja ma generować flow	*true* lub *false*
GenerateType	Włącza lub wyłącza generowanie typu dokumentu biznesowego	*true* lub *false*
GenerateAutonumeration	Włącza lub wyłącza generowanie licznika do numeracji dokumentów biznesowych	*true* lub *false*
GenerateWorkflow	Włącza lub wyłącza generowanie przejścia statusu	*true* lub *false*
GenerateForm	Włącza lub wyłącza generowanie formularza dla definicji dokumentu biznesowego	*true* lub *false*
GenerateRecordList	Włącza lub wyłącza generowanie widoku listowego dla definicji dokumentu	*true* lub *false*

Model JSON dla akcji tworzenia tabeli i flow

Klucz	Opis	Dozwolone wartości
	biznesowego	

PRZYKŁAD

Przykład akcji tworzącej tabelę *Rejestr reklamacji* (z instrukcji **Pierwsze kroki**) oraz generującej flow na podstawie tej tabeli.

```
{
"TableName": "RejestrReklamacji",
"TableSchema": "dbo",
"TablePrefix": "rklRjs",
"TableType": 1,
"PkName": "Id",
"PkType": "int",
"AutoIncrement": true,
"CustomFields": [
{
"Name": "EmailKlienta",
"Type": "varchar(50)",
"IsNullable": false,
"IsRodo": false,
"IsFullTextSearch": false,
"DictionaryId": null
},
{
"Name": "TekstReklamacji",
"Type": "varchar(500)",
"IsNullable": false,
"IsRodo": false,
"IsFullTextSearch": false,
"DictionaryId": null
}
],
"RunFlow": true,
"GenerateType": true,
"GenerateAutonumeration": true,
"GenerateFlow": true,
```

```
"GenerateForm": true,
"GenerateRecordList": true
}
```

Powiązane tematy:

- Kreator modeli danych

6.1.38. Tworzenie pliku tekstowego

Podsumowanie:
Akcja tworzenia pliku tekstowego umożliwia wygenerowanie pliku tekstowego o dowolnej strukturze na podstawie danych zwróconych przez zapytanie SQL oraz wartości zmiennych typu *{@nazwa}*. Akcja wymaga określenia szablonu generującego dane tekstowe z wykorzystaniem składni języka *Razor* (patrz opis w www.w3schools.com) oraz zapytania zwracającego dane źródłowe. Wygenerowany plik jest automatycznie zapisywany w folderze pobierania przeglądarki.

☛ *AdminSPA* > `NARZĘDZIA` > `Akcje` > `Tworzenie pliku tekstowego`

Edycja akcji konwersji do formatu

Kod akcji:* helpexample

Nazwa akcji:* helpexample

Opis:

Aplikacja: * BaseApp

Moduł: * BaseModule

Aktywna:*

Środowisko: Deweloperskie Testowe Produkcyjne

Warunkowe wykonanie:*

Nazwa pliku:* TXT helpexample

Podgląd zapytania SQL

Przykład: `Faktura nr {@Code} z dnia {@CreateDate}`

Niedozwolone znaki (", <, >, |, :, *, ?, \, /) w nazwie pliku zostaną zastąpione '-'.

Format pliku:* xml

Szablon razor:*

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  var kod = @Model.FormContext.Code
3  <document code="@kod"
4  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"<
5  xsi:noNamespaceSchemaLocation="usersroles.xsd">
6      @if(Model.SqlContext != null)
7      {
8      <Users>
9          @foreach(var user in Model.SqlContext)
10         {
11         <User>
12             <Name>@user.UserName</Name>
13             @if(user.Roles != null)
14             {
15             <Roles>
16                 @foreach(var role in user.Roles)
17                 {
18                 <Role>
19                     <Id>@role.Id</Id>
20                     <Name>@role.Name</Name>
21                 </Role>
22                 }
23             </Roles>
24             }
25         </User>
26         }
```

Sql modelu danych:*

- Tables
  - [dbo].[Covid]
  - [dbo].[CustomAcceptanceListProc
  - [dbo].[DataTypeTest]
  - [dbo].[DocumentToUsersAcceptar
  - [dbo].[Rozliczenia]

Odśwież

Sprawdź składnię SQL

```
1  SELECT UP.UserName,
2  JSON_QUERY(
3      (SELECT Roles.Id, Roles.Name FROM [core].[Roles]
4      JOIN [core].[UserProfileRoles] UR ON Roles.Id = UR.RoleId
5      WHERE UR.UserProfileId = UP.Id
6      FOR JSON AUTO)) AS Roles
7  FROM [core].[UserProfiles] UP
8  FOR JSON AUTO, INCLUDE_NULL_VALUES
```

Zapytanie musi zwrócić dane w formacie json.

Przykład:
SELECT UserName FROM core.UserProfiles FOR JSON AUTO

Dostępne parametry w SQL:

Zapisz Anuluj

Tworzenie pliku tekstowego

Oprócz właściwości wspólnych dla wszystkich akcji, w akcji tworzenia pliku tekstowego należy określić następujące właściwości:

- `Format pliku`: wpisz rozszerzenie nazwy pliku lub wybierz je z listy.
- `Szablon razor`: okno edycji szablonu razor; w szablonie można używać danych zwracanych przez SQl poprzez odwołania do tablicy *Model.SqlContext* oraz wartosci z kontekstu formularza poprzez odwołania do tablicy *Model.FormContext*; w oknie szablonu jest dostępna asysta wprowadzania wyświetlana po wprowadzeniu znaku < z listą znaczników HTML.
- `SQL modelu danych`: zapytanie SQL zwracające dane, które zostaną wyrenderowane w szablonie Razor jak plik tekstowy; zapytanie musi zwrócić dane w formacie *JSON*.

PRZYKŁAD

Przykład akcji generującej plik *XML* na podstawie danych zwróconych przez zapytanie.

Zapytanie SQL:

```
SELECT UP.UserName,
JSON_QUERY(
    (SELECT Roles.Id, Roles.Name FROM [core].[Roles]
    JOIN [core].[UserProfileRoles] UR ON Roles.Id = UR.RoleId
    WHERE UR.UserProfileId = UP.Id
    FOR JSON AUTO)) AS Roles
FROM [core].[UserProfiles] UP
FOR JSON AUTO, INCLUDE_NULL_VALUES
```

Wynik zapytania:

```
[
   {
      "UserName":"system",
      "Roles":null
   },
   {
      "UserName":"admin",
      "Roles":[
         {
```

```
            "Id":3,
            "Name":"Klient"
         }
      ]
   },
   {
      "UserName":"consultant",
      "Roles":[
         {
            "Id":2,
            "Name":"Konsultant"
         }
      ]
   },
   {
      "UserName":"jtester",
      "Roles":[
         {
            "Id":3,
            "Name":"Klient"
         },
         {
            "Id":6,
            "Name":"Kontrola jakości"
         }
      ]
   },
   {
      "UserName":"akajot",
      "Roles":[
         {
            "Id":3,
            "Name":"Klient"
         },
         {
            "Id":7,
            "Name":"Kierownik KJ"
         }
      ]
   },
   {
      "UserName":"stechniczny",
```

```
      "Roles":[
         {
            "Id":3,
            "Name":"Klient"
         },
         {
            "Id":8,
            "Name":"Pracownik DT"
         }
      ]
   },
   {
      "UserName":"rfinan",
      "Roles":[
         {
            "Id":3,
            "Name":"Klient"
         },
         {
            "Id":10,
            "Name":"Finanse"
         }
      ]
   },
   {
      "UserName":"msciprzy",
      "Roles":[
         {
            "Id":3,
            "Name":"Klient"
         },
         {
            "Id":11,
            "Name":"Pracownik"
         }
      ]
   },
   {
      "UserName":"teopol",
      "Roles":[
         {
            "Id":3,
```

```
            "Name":"Klient"
         },
         {
            "Id":11,
            "Name":"Pracownik"
         }
      ]
   },
   {
      "UserName":"euneu",
      "Roles":[
         {
            "Id":1,
            "Name":"Administrator"
         },
         {
            "Id":5,
            "Name":"Administrator użytkowników PublicAPI"
         },
         {
            "Id":11,
            "Name":"Pracownik"
         },
         {
            "Id":12,
            "Name":"Kierownik"
         }
      ]
   },
   {
      "UserName":"jenglish",
      "Roles":[
         {
            "Id":3,
            "Name":"Klient"
         },
         {
            "Id":6,
            "Name":"Kontrola jakości"
         }
      ]
   }
```

```
]
```

Szablon Razor:

```
<?xml version="1.0" encoding="UTF-8"?>

<document code="@Model.FormContext.Code"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"<
xsi:noNamespaceSchemaLocation="usersroles.xsd">
    @if(Model.SqlContext != null)
    {
    <Users>
        @foreach(var user in Model.SqlContext)
        {
        <User>
            <Name>@user.UserName</Name>
            @if(user.Roles != null)
            {
            <Roles>
                @foreach(var role in user.Roles)
                {
                <Role>
                    <Id>@role.Id</Id>
                    <Name>@role.Name</Name>
                </Role>
                }
            </Roles>
            }
        </User>
        }
    </Users>
    }
</document>
```

Wynikowy plik *XML*:

```
<?xml version="1.0" encoding="UTF-8"?>
<document xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:noNamespaceSchemaLocation="usersroles.xsd">
   <Users>
      <User>
```

```
      <Name>system</Name>
   </User>
   <User>
      <Name>admin</Name>
      <Roles>
         <Role>
            <Id>1</Id>
            <Name>Administrator</Name>
         </Role>
      </Roles>
   </User>
   <User>
      <Name>consultant</Name>
      <Roles>
         <Role>
            <Id>2</Id>
            <Name>Konsultant</Name>
         </Role>
      </Roles>
   </User>
   <User>
      <Name>jtester</Name>
      <Roles>
         <Role>
            <Id>3</Id>
            <Name>Klient</Name>
         </Role>
         <Role>
            <Id>6</Id>
            <Name>Kontrola jakości</Name>
         </Role>
      </Roles>
   </User>
   <User>
      <Name>akajot</Name>
      <Roles>
         <Role>
            <Id>3</Id>
            <Name>Klient</Name>
         </Role>
         <Role>
            <Id>7</Id>
```

```
            <Name>Kierownik KJ</Name>
         </Role>
      </Roles>
   </User>
   <User>
      <Name>stechniczny</Name>
      <Roles>
         <Role>
            <Id>3</Id>
            <Name>Klient</Name>
         </Role>
         <Role>
            <Id>8</Id>
            <Name>Pracownik DT</Name>
         </Role>
      </Roles>
   </User>
   <User>
      <Name>rfinan</Name>
      <Roles>
         <Role>
            <Id>3</Id>
            <Name>Klient</Name>
         </Role>
         <Role>
            <Id>10</Id>
            <Name>Finanse</Name>
         </Role>
      </Roles>
   </User>
   <User>
      <Name>msciprzy</Name>
      <Roles>
         <Role>
            <Id>3</Id>
            <Name>Klient</Name>
         </Role>
         <Role>
            <Id>11</Id>
            <Name>Pracownik</Name>
         </Role>
      </Roles>
```

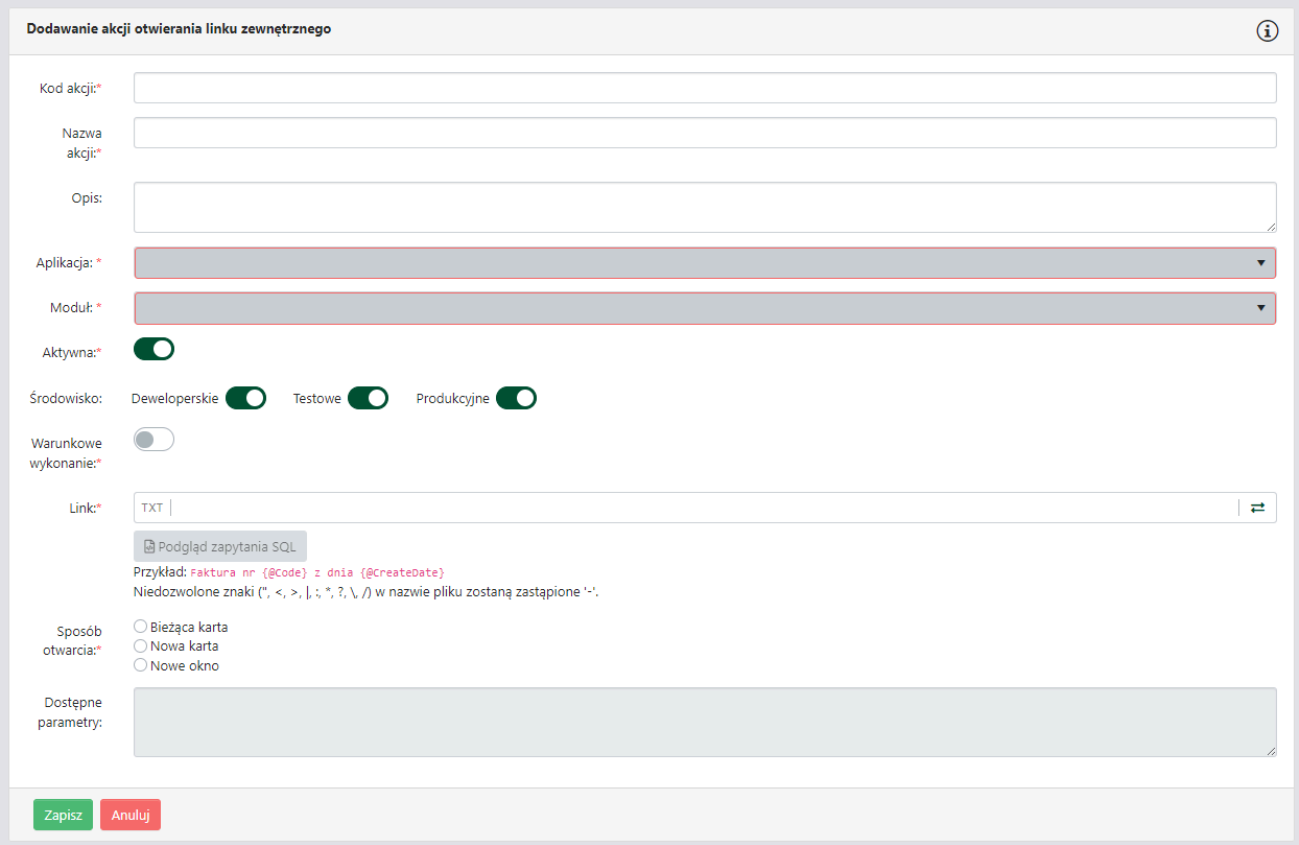
```
    </User>
    <User>
       <Name>teopol</Name>
       <Roles>
          <Role>
             <Id>3</Id>
             <Name>Klient</Name>
          </Role>
          <Role>
             <Id>11</Id>
             <Name>Pracownik</Name>
          </Role>
       </Roles>
    </User>
    <User>
       <Name>euneu</Name>
       <Roles>
          <Role>
             <Id>1</Id>
             <Name>Administrator</Name>
          </Role>
          <Role>
             <Id>5</Id>
             <Name>Administrator użytkowników PublicAPI</Name>
          </Role>
          <Role>
             <Id>11</Id>
             <Name>Pracownik</Name>
          </Role>
          <Role>
             <Id>12</Id>
             <Name>Kierownik</Name>
          </Role>
       </Roles>
    </User>
    <User>
       <Name>jenglish</Name>
       <Roles>
          <Role>
             <Id>3</Id>
             <Name>Klient</Name>
          </Role>
```

```
            <Role>
               <Id>6</Id>
               <Name>Kontrola jakości</Name>
            </Role>
         </Roles>
      </User>
   </Users>
</document>
```

6.1.39. Otwieranie linku zewnętrznego

Podsumowanie:
Akcja otwierania linku zewnętrznego wyświetla stronę docelową podanego linku zewnętrznego.

☛ *AdminSPA* > `Narzędzia` > `Akcje` > `Otwieranie linku zewnętrznego`



Akcja otwierania linku zewnętrznego

Oprócz właściwości wspólnych dla wszystkich akcji, w akcji otwierania linku zewnętrznego należy określić następujące właściwości:

`Link`: wpisz adres URL strony docelowej; w linku można używać zmiennych typu *{@Nazwa}*; po przełączeniu typu pola wejściowego można zdefiniować zapytanie SQL, które zwróci link.

`Sposób otwarcia`:

- `Bieżąca karta`
- `Nowa karta`
- `Nowe okno`

Powiązane tematy:

- Wspólne parametry akcji

6.1.40. Akcja przełączania aplikacji

Podsumowanie:
Akcja zmienia aktywną aplikację w trybie pojedynczej aplikacji, to jest wyświetla menu użytkownika zdefiniowane dla tej aplikacji oraz jej domyślny pulpit.

☛ *AdminSPA* > `NARZĘDZIA` > `Akcje` > `Przełącz aplikację`

Oprócz właściwości wspólnych dla wszystkich akcji, w akcji przełączania aplikacji należy wskazać aplikację docelową. Można to zrobić na dwa sposoby:

- `Wybierz z listy`: wybierz z listy rozwijanej aplikację, którą akcja ustawi jako aktywną.
- `Oblicz dynamicznie`: w trybie tekstowym można wpisać kod aplikacji lub jej identyfikator, a także referencję smart numbers w formie *{&KodAplikacji}* lub odwołanie do zmiennej aplikacji, która zwróci identyfikator lub kod aplikacji docelowej; w trybie SQL zdefiniuj zapytanie, które zwróci identyfikator (*Id* lub *RowGuid*) albo kod (*Code*) aplikacji docelowej, na przykład:

```
SELECT CAST({@AppGuid} AS uniqueidentifier)
```

WSKAZÓWKA
Jeśli tryb pojedynczej aplikacji jest wyłączony, uruchomienie akcji nie powoduje żadnego efektu.

Powiązane tematy:

- Wspólne parametry akcji

6.1.41. Algorytm C#

Podsumowanie:
Akcja pozwala użytkownikowi przygotować algorytm w języku C#. W kodzie można odwoływać się do bibliotek DLL dostępnych w folderze instalacyjnym witryny w podfolderze *API* lub folderach innych serwisów (oprócz wyboru z listy, można wpisać nazwę pliku) oraz do wartości z kontekstu formularza za pośrednictwem zmiennej *Model*.

☛ `Narzędzia` > `Akcje` > `Algorytm C#`

Akcja oferuje kilka klas, które opisano osobno. Są to:

- *SqlAction*, *EmailAction*, *CSharpScriptAction* oraz *WebServiceAction* (przestrzeń nazw *CSharpScript.ActionRunner*): umożliwiają wywołanie akcji zdefiniowanych w nAxiom (akcje SQL, wysyłka e-mail, zapytania do webserwisów oraz algorytmy C#).
- *RestManager* i *SoapManager* (przestrzeń nazw *CSharpScript.WebServiceRunner*): umożliwiają wysyłanie zapytań do serwisów SOAP i REST API.
- *SqlHelper* (przestrzeń nazw *CSharpScript.SqlRunner*): umożliwia wykonywanie typowych operacji na bazie danych SQL Server; patrz Akcja Algorytm C#, klasa SQLHelper .
- *DocumentAttachmentsHelper*: umożliwia zarządzanie załącznikami dokumentu; patrz Algorytm C#, klasa DocumentAttachmentsHelper .

Ponadto w akcji można używać wbudowanych elementów, takich jak:

- funkcja *ClearCache()*: czyści pamięć podręczną witryny nAxiom, tak samo jak kliknięcie przycisku `Wyczyść cały cache` w sekcji `Cache` w ustawieniach systemu (`ADMINISTRACJA` > `Ustawienia systemu` > `Cache`);
- interfejs *ILogger*: służy do zapisywania komunikatów w logach systemowych; patrz Logowanie komunikatów w akcji C# ;
- referencje do sekretów plikowych przechowywanych w sejfie **OpenBao**:
  - środowiska:

    ```
    byte[] file = await TenantsAdmin.Environment.Secrets.GetSecretFile("KodSekretu");
    ```

  - bieżącego tenanta:

```
byte[] file = await TenantsAdmin.CurrentTenant.Secrets.GetSecretFile("KodSekretu");
```

Więcej informacji o sekretach zawiera artykuł Integracja nAxiom z OpenBao .

W celu zdefiniowania akcji algorytmu C# należy określić następujące właściwości:

`Emituj informacje debugowania`: włącza prezentowanie dodatkowych informacji przez system podczas kompilacji i wykonywania algorytmu

`Paczki bibliotek...`: umożliwia wybranie z listy paczki bibliotek DLL zadeklarowanych w systemie w aplikacji TenantsAdmin ; po wybraniu paczek dostępny jest podgląd bibliotek znajdujących się w poszczególnych paczkach. Aby można było w kodzie akcji odwoływać się do typów zdefiniowanych w przestrzeniach nazw tych bibliotek, należy się do nich zreferować przy użyciu dyrektywy *using*.

UWAGI

W związku z tym, że funkcjonalność akcji C# w nAxiom wykorzystuje kompilowany kod C#, dostarczone rozwiązanie ma pewne ograniczenia:

- Brak możliwości korzystania z bibliotek systemowych w innej wersji niż instalowane z systemem — biblioteki zostaną załadowane, ale odwołania do nich mogą powodować błędy.
- Brak możliwości załadowania bibliotek w różnych wersjach: jeśli dwie paczki bibliotek będą zawierać tę samą bibliotekę DLL w różnych wersjach, próba załadowania drugi raz tej biblioteki zakończy się błędem. Poprawnie załadowana i wykonywana będzie tylko jedna wersja pliku DLL.
- Zmiana wersji pliku DLL wymaga ponownego uruchomienia serwisów wykonujących akcję C#, która korzysta z danej paczki bibliotek. W przeciwnym razie akcja będzie działać bez uwzględnienia zmian przez ok. 2 godziny, ponieważ tyle wynosi interwał odświeżania cache, w którym przechowywane są wyniki kompilacji.
- W przypadku usunięcia paczki z poziomu TenantsAdmin SPA akcja będzie się wykonywać dopóki jest przechowywana w cache. Błąd o braku paczki zostanie zgłoszony dopiero w momencie następnej kompilacji skryptu.

`Biblioteki systemowe`: referencje do zestawów (*asembly*), które mają być dostępne w kodzie akcji; **nazwę zestawu można wpisać ręcznie**; w przypadku akcji przypisanych do przycisku lub przejścia na diagramie procesów plik zestawu musi znajdować się w folderze serwisu *api*; w przypadku akcji wykonywanych w ramach zadań cyklicznych plik zestawu musi znajdować się w folderze serwisu *taskservice*; standardowo dostępne są biblioteki *System.Data.SqlClient.dll*, *Dapper.dll* i *Newtonsoft.Json.dll*. Niezależnie od wybrania zestawów

w tym polu, aby móc używać w kodzie akcji typów zdefiniowanych w przestrzeniach nazw tych zestawów, należy się do nich zreferować przy użyciu dyrektywy *using*.

`Algorytm C#`: kod akcji w języku C#; w kodzie można korzystać z predefiniowanych zmiennych:

- *Model (string)*: model formularza w formacie JSON
- *RecordId (string)*; identyfikator bieżącego rekordu
- *FormDefinitionId (int)*: identyfikator bieżącego formularza
- *BusinessDocumentId (int)*: identyfikator bieżącego dokumentu biznesowego
- *UserId (guid)*: identyfikator zalogowanego użytkownika
- *TenantId (string)*: identyfikator bieżącego tenanta
- *TenantBaseUrl (string)*: adres URL witryny bieżącego tenanta
- Przycisk `Sprawdź składnię/Skompiluj algorytm` weryfikuje poprawność składni i uruchamia kompilację kodu (tak działa również przycisk `Zapisz`). W przypadku błędów składni bądź braku referencji do zestawów, zostanie wyświetlony komunikat o błędzie, a lista błędów zostanie wyświetlona poniżej okna `Algorytm C#`. Akcję można zapisać tylko w przypadku pomyślnej weryfikacji/kompilacji.

INFO

W oknie edytora kodu można korzystać z asysty przy definiowaniu referencji *smart numbers*.

WSKAZÓWKA

W celu poprawy wydajności działania akcji C# wprowadzono zapisywanie skompilowanych skryptów akcji w pamięci cache. Ustawienia konfiguracyjne pamięci cache dla akcji C# znajdują się w sekcji *AppConfiguration:Modules:Scripting* w pliku *appsettings.json* serwisu *api*. Są to:

- `SlidingCacheExpirationInHours`: czas w godzinach (liczba zmiennoprzecinkowa) od ostatniego użycia, po upływie którego skompilowany skrypt jest usuwany z cache.
- `EnableCompiledScriptCache`: wyłączenie zapisywania skompilowanych skryptów w pamięci cache.

Powiązane tematy:

- Wspólne parametry akcji
- Zadania cykliczne
- Przestrzeń nazw SQLRunner
- Przestrzeń nazw WebServiceRunner
- Przestrzeń nazw ActionRunner

6.1.41.1. PRZESTRZEŃ NAZW SQLRUNNER

Podsumowanie:
Klasa *SqlHelper* to klasa statyczna dostępna w przestrzeni nazw *CSharpScript.SqlRunner*, która pozwala na wykonanie typowych operacji na bazach danych. Wspierane są połączenia z bazą systemową nAxiom oraz źródłami danych w bazach MS SQL i Oracle.

W klasie *SqlHelper* są dostępne następujące metody:

- *ExecuteSqlQueryRowsToDictionary*: zwraca wynik zapytania SQL jako *List<Dictionary<string, object>>*,
- *ExecuteSqlScalar*: zwraca wynik zapytania jako pojedynczy rezultat typu *object*,
- *ExecuteSqlNonQuery*: zwraca w wyniku *true* lub *false*, zależnie od rezultatu zapytania modyfikującego dane (np. czy instrukcja UPDATE dokonała aktualizacji rekordu); zwracany typ to *bool*,
- *ExecuteSqlNonQueryWithOutputParameters*: metoda wykorzystywana przede wszystkim do wykonywania procedur, gdzie wynik zwracany jest za pomocą parametrów, wynik zwracany jako *Dictionary<string, object>*.

PARAMETRY WEJŚCIOWE

Wszystkie metody przyjmują zestaw następujących parametrów wejściowych:

- *query* (*string*): zapytanie SQL do wykonania na źródle danych,
- *parameters* (*Dictionary<string, object>*): słownik parametrów wejściowych zapytania SQL,
- *dataSourceId* (*int*): identyfikator źródła danych.

W metodzie *ExecuteSqlNonQueryWithOutputParameters* parametr *parameters* nie jest używany, zamiast niego używane są parametry:

- *inputParameters* (*Dictionary<string, object>*): słownik parametrów wejściowych zapytania SQL,

- *outputParameters* (*Dictionary<string, object>*): słownik parametrów wyjściowych zapytania SQL,

METODA *EXECUTESQLQUERYROWSTODICTIONARY*

Asynchroniczna metoda statyczna służąca do pobierania wyniku zapytania SQL w formie *List<Dictionary<string, object>>*. Zwracana lista to lista rekordów, a słownik opisuje wartości kolumn w rekordzie: klucz to nazwa kolumny, a wartość to wartość z tej kolumny zwrócona przez SQL.

Wynik:

List<Dictionary<string, object>>

lista reprezentująca rekordy zwrócone przez zapytanie SQL, słownik reprezentuje wartości w poszczególnych kolumnach rekordu: klucz to nazwa kolumny, wartość klucza to wartość z kolumny.

Przykład:

```
using System;
using CSharpScript.SqlRunner;
using Newtonsoft.Json;
using System.IO;

var result =
await SqlHelper.ExecuteSqlQueryRowsToDictionary("SELECT UserName, Email, FirstName, LastName FROM [core].[Us
File.WriteAllText(@"C:\nAxiom\Akcje\Profile.txt",
    JsonConvert.SerializeObject(result, Formatting.Indented));
```

Plik, w którym mają zostać zapisane dane musi istnieć. Akcja zapisze dane w pliku w formacie JSON:

```
[
  {
    "UserName": "jtester",
    "Email": "jtester@naxiom.com",
    "FirstName": "Jerzy",
    "LastName": "Tester"
  },

...
```

```
]
```

METODA *EXECUTESQLSCALAR*

Asynchroniczna metoda statyczna służąca do pobierania wyniku zapytania SQL w formie pojedynczej wartości typu *object*.

Wynik:

object
: pojedyncza wartość zwracana przez zapytanie SQL

Przykład:

```
using System;
using CSharpScript.SqlRunner;
using System.IO;

File.WriteAllText(@"C:\nAxiom\Akcje\ScalarLocal.txt",
(await SqlHelper.ExecuteSqlScalar(
    "SELECT MIN(ExpireDate) FROM [core].[UserProfiles]")).ToString());
```

Plik tekstowy musi istnieć.

METODA *EXECUTESQLNONQUERY*

Asynchroniczna metoda statyczna służąca do wywołania polecenia SQL do manipulowania danymi (*NonQuery*, czyli nie zwracającego żadnego wyniku) na przykład *INSERT*, *UPDATE*, *DELETE*. Rezultat to wartość *true* lub *false*, zależnie od tego, czy zapytanie dokonało modyfikacji danych.

Wynik:

***true/false* (*bool*)**
: wartość zależna od tego, czy polecenie zmieniło dane w źródle

Przykład:

```
var result =
await SqlHelper.ExecuteSqlNonQuery(
    "INSERT INTO [dbo].[Names] ([Name]) VALUES ('Michał')");
```

METODA *EXECUTESQLNONQUERYWITHOUTPUTPARAMETERS*

Asynchroniczna metoda statyczna służąca do wywołania polecenia SQL, którego wynik jest zwracany za pomocą parametrów wyjściowych, na przykład procedury. Wynik jest zwracany w formie słownika, w którym klucz to nazwa parametru wyjściowego, a wartość to wynik dla danego parametru zwrócony przez zapytanie.

Wynik:

Dictionary<string, object>

słownik wartości wyjściowych zapytania SQL, gdzie klucz to nazwa parametru przekazana w danych wyjściowych (*outputParameters*), a wartość to dane zwrócone przez zapytanie

Przykład:

```
var result =
await SqlHelper.ExecuteSqlNonQueryWithOutputParameters(
"EXEC [dbo].[InsertNewNameWithOutputParam] @Name = {@Name}, @Id = {@Id} OUTPUT",
 new Dictionary<string, object>() { { "Name", "Protazy"} },
 new Dictionary<string, object>() { { "Id", 0} });
```

INNE ŹRÓDŁA DANYCH

Metod klasy *SqlHelper* można używać do wykonywania zapytań na bazach innych niż systemowa, dzięki użyciu parametru *dataSourceId*. Podczas definiowania akcji wartość parametru można określić przy użyciu mechanizmu *SmartNumbers*.

Przykład:

```
var resultDevLocal =
await SqlHelper.ExecuteSqlQueryRowsToDictionary("SELECT * FROM [core].[UserProfiles]",
null, {&BaseApp.DataSources.BaseModule.My_DS});
File.WriteAllText(@"C:\nAxiom\Akcje C#\2 ProfileDevLocal.txt",
    JsonConvert.SerializeObject(resultDevLocal, Formatting.Indented));
```

OBSŁUGA BŁĘDÓW

Mechanizm wspiera standardową obsługę błędów. Błędy specyficzne dla klasy *SqlHelper* to:

- *Api.ActionModule.CSharpScriptDataSourceNotExist*: brak źródła danych dla wskazanego ID,
- *Api.ActionModule.CSharpScriptNonSqlDataSource*: wskazane źródło danych nie jest źródłem SQL lub źródło nie zostało zaimplementowane.

PRZEKAZYWANIE PARAMETRÓW

W poleceniu SQL można używać zmiennych z platformy nAxiom, to znaczy *{@Parametr}*. Wartości tych zmiennych należy przekazać na przykład za pomocą słownika *Dictionary<string, object>*.

Taki słownik można zbudować ręcznie lub przekazać zdeserializowany model formularza ze zmiennej globalnej *Model*.

Przykład:

```
JsonConvert.DeserializeObject<Dictionary<string, object>>(Model)
```

Pamiętać należy o dołączeniu biblioteki *Newtonsoft.Json.dll* oraz dodaniu przestrzeni nazw na początku skryptu:

```
using Newtonsoft.Json;
```

PRZYKŁADY KODU

1. Pobranie profili użytkowników z bazy systemowej w formacie JSON i zapisanie do pliku

```
var result =
await SqlHelper.ExecuteSqlQueryRowsToDictionary(
 "SELECT * FROM [core].[UserProfiles]");
File.WriteAllText(@"C:\nAxiom\Akcje C#\1 Profile.txt",
 JsonConvert.SerializeObject(result, Formatting.Indented));
```

2. Pobranie profili użytkowników z innego źródła w formacie JSON i zapisanie do pliku

```
var resultDevLocal =
await SqlHelper.ExecuteSqlQueryRowsToDictionary(
 "SELECT * FROM [core].[UserProfiles]",
null, {&BaseApp.DataSources.BaseModule.My_DS});
File.WriteAllText(@"C:\nAxiom\Akcje C#\2 ProfileDevLocal.txt",
```

```
 JsonConvert.SerializeObject(resultDevLocal, Formatting.Indented));
```

3. Pobranie pierwszego ID profilu użytkownika

```
File.WriteAllText(@"C:\nAxiom\Akcje C#\4 ScalarLocal.txt",
(await SqlHelper.ExecuteSqlScalar(
 "SELECT TOP 1 [Id] FROM [core].[UserProfiles]")).ToString());
```

4. Wykonanie procedury jako polecenie skalarne, gdzie wynik to *ID* wstawionego wpisu *(SCOPE_IDENTITY())*, gdzie przekazano dane z formularza

```
File.WriteAllText(@"C:\nAxiom\Akcje C#\9 ScalarMSInsertProcedureFromModel.txt",
(await SqlHelper.ExecuteSqlScalar(
 "EXEC [dbo].[InsertNewName] {@Imie}",
 JsonConvert.DeserializeObject<Dictionary<string, object>>(Model))).ToString());
```

5. Wykonanie procedury na bazie Oracle z danymi wyjściowymi w formie parametrów, procedura zwraca imię dla wskazanego *ID* oraz datę wykonania

```
var resultParams =
await SqlHelper.ExecuteSqlNonQueryWithOutputParameters(
 "BEGIN GETNAMEBYID({@Id}, {@Imie}, {@Date}); END;",
new Dictionary<string, object>() { { "Id", 10} },
new Dictionary<string, object>() { { "Imie", ""}, { "Date", ""} }, 3);
```

Powiązane tematy:

- Akcja algorytmu C#
- Przestrzeń nazw WebServiceRunner
- Przestrzeń nazw ActionRunner
- Zadania cykliczne

Podsumowanie:
W akcji algorytm C# można korzystać z dwóch klas z przestrzeni nazw *CSharpScript.WebServiceRunner*, które umożliwiają wykonywanie zapytań do webserwisów:

- *CSharpScript.WebServiceRunner.RestManager*: służy dla wykonania zapytania REST
- *CSharpScript.WebServiceRunner.SoapManager*: służy dla wykonania zapytania SOAP

Aby wykonać żądanie do webserwisu, należy użyć metody *ExecuteRequestAsync* dla instancji odpowiedniej klasy. Na przykład:

```
await CSharpScript.WebServiceRunner.RestManager.Instance.ExecuteRequestAsync(model, cancellationToken);
await CSharpScript.WebServiceRunner.SoapManager.Instance.ExecuteRequestAsync(model, cancellationToken);
```

METODA EXECUTEREQUESTASYNC

```
public Task<IResponse> ExecuteRequestAsync(BaseRequestModel model, CancellationToken cancellationToken = def
```

Parametry:

- *Model*: model żądania; w zależności od typu żądania — *RestRequestModel* albo *SoapRequestModel*
- *CancellationToken*: parametr nieobowiązkowy; propaguje powiadomienie, że operacja powinna zostać anulowana.

Wynik:

Asynchroniczna operacja o typie *IResponse*:

```
public interface IResponse
{
        /// <summary>
        /// Contains the values of status codes returned from HTTP request
        /// Type: System.Net.HttpStatusCode
        /// </summary>
        HttpStatusCode ResponseStatusCode { get; }
```

```
        /// <summary>
        /// Contains information whether the request was successful
        /// </summary>
        bool IsSuccess { get; }

        /// <summary>
        /// Result of HTTP request
        /// In case of WebException will contains exception message
        /// </summary>
        string RawResult { get; }
}
```

KLASA *RESTMANAGER*

Sposób wywołania:

```
var model = new CSharpScript.WebServiceRunner.Models.RestRequestModel();
var result = await CSharpScript.WebServiceRunner.RestManager.Instance.ExecuteRequestAsync(model, cancellatic
```

Dla klasy *RestManager* metoda *ExecuteRequestAsync* przyjmuje jako model klasę o nazwie *RestRequestModel*.

```
public class RestRequestModel
{
        /// <summary>
        /// HTTP method to call (System.Net.Http.HttpMethod)
        /// Allowed values are Get | Post | Delete | Put | Patch
        /// </summary>
        public HttpMethod Method { get; set; }

        /// <summary>
        /// Request headers
        /// </summary>
        public Dictionary<string, string> Headers { get; set; }

        /// <summary>
        /// Web service URL
        /// </summary>
        public string Url { get; set; }
```

```
/// <summary>
/// Request body to send
/// REST - used when HttpMethod is set to POST | PUT | Patch
/// </summary>
public string Body { get; set; }

      /// <summary>
/// Authentication type
/// Default value is AuthType.NoAuth
/// ApiConnector.Enums.AuthType (NoAuth | Base64 | JWT | OAuth2Password | OAuth2ClientCredentials)
/// In REST Manager could be used with Base64 | JWT | OAuth2Password | OAuth2ClientCredentials
/// </summary>
public AuthType AuthType { get; set; } = AuthType.NoAuth;

/// <summary>
/// URL to authentication service
/// Could be null or empty when AuthType is set to Base64
/// Is required when AuthType is set to JWT | OAuth2Password | OAuth2ClientCredentials
/// </summary>
public string AuthUrl { get; set; }

/// <summary>
/// Username for authentication purpose
/// REST - is used when AuthType is set to Base64 | JWT | OAuth2Password
/// </summary>
public string UserName { get; set; }

/// <summary>
/// Password for authentication purpose
/// REST - is used when AuthType is set to Base64 | JWT | OAuth2Password
/// </summary>
public string Password { get; set; }

/// <summary>
/// Authentication scope
/// Used when AuthType is set to OAuth2Password | OAuth2ClientCredentials
/// </summary>
public string Scope { get; set; }

/// <summary>
/// Authentication client id
```

```
        /// Used when AuthType is set to OAuth2Password | OAuth2ClientCredentials
        /// </summary>
        public string ClientId { get; set; }

        /// <summary>
        /// Authentication client secret
        /// Used when AuthType is set to OAuth2Password | OAuth2ClientCredentials
        /// </summary>
        public string ClientSecret { get; set; }
}
```

KLASA *SOAPMANAGER*

Sposób wywołania:

```
var model = new CSharpScript.WebServiceRunner.Models.SoapRequestModel();
var result = await CSharpScript.WebServiceRunner.SoapManager.Instance.ExecuteRequestAsync(model);
```

Dla klasy *SoapManager* metoda *ExecuteRequestAsync* przyjmuje jako model klasę o nazwie *SoapRequestModel*.

```
public class SoapRequestModel
{
            /// <summary>
        /// Request headers
        /// </summary>
        public Dictionary<string, string> Headers { get; set; }

        /// <summary>
        /// Web service URL
        /// </summary>
        public string Url { get; set; }

        /// <summary>
            /// Content type - not required
        /// Default value is 'text/xml;charset="utf-8"'
        /// </summary>
        public string ContentType { get; set; }

        /// <summary>
```

```
        /// SOAP action from WSDL
        /// </summary>
        public string Action { get; set; }

        /// <summary>
        /// Request body to send
        /// SOAP - xml to send
        /// </summary>
        public string Body { get; set; }

        /// <summary>
        /// Authentication type
        /// Default value is AuthType.NoAuth
        /// ApiConnector.Enums.AuthType (NoAuth | Base64 | JWT | OAuth2Password | OAuth2ClientCredentials)
        /// In Soap Manager could be use only with Base64
        /// </summary>
        public AuthType AuthType { get; set; } = AuthType.NoAuth;

        /// <summary>
        /// Username for authentication purpose
        /// SOAP - is used when AuthType is set to Base64
        /// </summary>
        public string UserName { get; set; }

        /// <summary>
        /// Password for authentication purpose
        /// SOAP - is used when AuthType is set to Base64
        /// </summary>
        public string Password { get; set; }
}
```

Powiązane tematy:

- Akcja algorytmu C#
- Przestrzeń nazw WebServiceRunner
- Przestrzeń nazw SQLRunner
- Zadania cykliczne

Podsumowanie:
W akcji C# można używać klas z przestrzeni nazw *CSharpScript.ActionRunner*, które umożliwiają wykonywanie niektórych typów akcji.

W TYM TEMACIE

- Wprowadzenie
- Obsługa wyjątków
- Akcja zapytania SQL
- Akcja wysyłki e-mail
- Akcja zapytania do webserwisu
- Akcja C#
- Akcja raportów
- Akcja zapisu rekordu
- Akcja zmiany statusu

WPROWADZENIE

Do wykonania akcji służy funkcja *RunActionAsync*, która działa asynchronicznie. Aby akcje wykonywały się synchronicznie, wywołanie tej funkcji należy poprzedzić słowem kluczowym *await*. Funkcja przyjmuje dwa parametry: identyfikator wykonywanej akcji oraz model w formacie JSON. Identyfikator akcji można podać, używając składni smart numbers (*{&BaseApp.Actions.BaseModule.KodAkcji}*).

Modele danych dla akcji przedstawiono poniżej, jeśli nie trzeba przekazywać parametrów, można wstawić wartość *NULL* lub pusty ciąg znaków.

Funkcja *RunActionAsync* zwraca obiekt typu *ActionExecutionResult*, który jest zbudowany następująco:

```
public class ActionExecutionResult
{
public string Result { get; set; }
public bool IsValid { get; set; }
public ValidationResult ValidationResult { get; set; }
public List<ResponseMessage> ResponseMessages { get; set; }
}
```

gdzie:

- *Result*: zmienna z obiektem JSON odpowiedzi z wykonywanej akcji, w większości wypadków będzie to typ *bool*:

```
{
"ExecutedSuccessfully": true,
"ReturnValue": "",
}
```

- *IsValid*: flaga poprawności wykonania akcji.
- *ValidationResult*: lista błędów, które wystąpiły w czasie wykonania akcji. Jest to klasa pochodząca z biblioteki *FluentValidation*, a dokładnie z przestrzeni nazw *FluentValidation.Results.ValidationResult*. *ResponseMessages*; jest to klasa która może przekazywać komunikaty z serwera SQL Server. Struktura:

```
public class ResponseMessage
{
public string Message { get; set; }
public ResponseMessageType Type { get; set; }
public IList<object> Arguments { get; set; }
public bool ShowImmediately { get; set; }
}
```

OBSŁUGA WYJĄTKÓW

Błędy, które wystąpią w akcjach, zostaną zalogowane i zwrócone w akcji Algorytm C#, gdzie można będzie je obsłużyć (w bloku `try catch`). Błędy są zwracane jako *Exception* typu *ActionExecutionException*. Wyjątek zawiera właściwość *ExecutionResult* typu *ActionExecutionResult*, w której jest przechowywany wynik wykonanej akcji:

```
public ActionExecutionResult ExecutionResult { get; set; }
```

Typ *ActionExecutionException* jest dostępny w przestrzeni nazw *CSharpScript.ActionRunner*.

Przykład:

```
using Shared.Exceptions;
try
{
```

```
    var actionResult = await CSharpScript.ActionRunner.SqlAction.Instance.RunActionAsync(1, null);
}
catch (CSharpScript.ActionRunner.ActionExecutionException exc)
{
    throw new TranslatedException("Błąd wywołania akcji SQL.", exc.ExecutionResult.ValidationResult);
}
```

WAŻNE

W wersji nAxiom 1.15.1.0 powyższy sposób obsługi błędów nie ma zastosowania do akcji *SaveRecord* i *ChangeStatusAction* (patrz przykłady poniżej). Planowane jest ujednolicenie obsługi błędów w jednej z kolejnych wersji.

AKCJA ZAPYTANIA SQL

```
await CSharpScript.ActionRunner.SqlAction.Instance.RunActionAsync(id, model);
```

Struktura modelu:

```
{
"RecordId": "string",
"Model": "model JSON dla wykonywanej akcji",
"UserId": "00000000-0000-0000-0000-000000000000",
"GridId": null,
"FormDefinitionId": null,
"BusinessDocumentId": null
}
```

AKCJA WYSYŁKI E-MAIL

```
await CSharpScript.ActionRunner.EmailAction.Instance.RunActionAsync(id, model);
```

Struktura modelu:

```
{
"RecordId": "string",
"Model": "model JSON dla wykonywanej akcji",
"UserId": "00000000-0000-0000-0000-000000000000",
```

```
"GridId": null,
"FormDefinitionId": null,
"BusinessDocumentId": null,
"ExtraEmailsToSend": [
"string",
"string"
]
}
```

AKCJA ZAPYTANIA DO WEBSERWISU

```
await CSharpScript.ActionRunner.WebServiceAction.Instance.RunActionAsync(id, model);
```

Struktura modelu:

```
{
"RecordId": "string",
"Model": "model JSON dla wykonywanej akcji",
"UserId": "00000000-0000-0000-0000-000000000000",
"GridId": null,
"FormDefinitionId": null,
"BusinessDocumentId": null
}
```

AKCJA C#

```
await CSharpScript.ActionRunner.CSharpScriptAction.Instance.RunActionAsync(id, model);
```

Struktura modelu:

```
{
"RecordId": "string",
"Model": "model JSON dla wykonywanej akcji",
"UserId": "00000000-0000-0000-0000-000000000000",
"GridId": null,
"FormDefinitionId": null,
"BusinessDocumentId": null
}
```

AKCJA RAPORTÓW

Akcja wywołana w ten sposób musi mieć włączony przełącznik `Zapisz wygenerowany plik do załączników`, pobranie wygenerowanego raportu przez przeglądarkę nie jest możliwe.

Wywołanie:

```
var result =  await CSharpScript.ActionRunner.ReportsAction.Instance.RunActionAsync({&BaseApp.Actions.BaseMo
```

Parametr *model* jest przekazywany jako ciąg znaków. Aby wygenerowany raport został dodany do załącznika, w modelu należy wskazać *RecordId* (instancję dokumentu) oraz *BusinessDocumentId* (definicję dokumentu). Dla poprawnego działania uprawnień konieczne jest przekazanie parametru *UserId*. Parametry używane w zapytaniu raportu należy przekazać w polu *Model* parametru *model*. To pole nie może mieć wartości *NULL*, jeśli parametry nie są przekazywane, musi ono być pustym obiektem JSON ({}):

```
using System;
using System.Linq;
using System.Text;
using Shared.Exceptions;
using Newtonsoft.Json;
using System.Collections.Generic;
using Shared.Exceptions;

var recordId = 122;
var model = new Dictionary<string,object>()
{
  { "Model", @"{""Test"" : 1}" },
  { "RecordId", recordId},
  { "BusinessDocumentId", BusinessDocumentId},
  { "UserId", UserId }
};

var serializedModel = JsonConvert.SerializeObject(model);

try
{
    var result =  await CSharpScript.ActionRunner.ReportsAction.Instance.RunActionAsync({&BaseApp.Actions.Ba
}
catch (CSharpScript.ActionRunner.ActionExecutionException exc)
```

```
{
    throw new TranslatedException("Błąd akcji raportów:", exc.ExecutionResult.ValidationResult);
}
```

W parametrze model można przekazać do akcji raportów wartość kontekstu (pole *Model*) samej akcji Algorytm C#:

```
var model = new Dictionary<string,object>()
{
  { "Model", Model,
  { "RecordId", recordId},
  { "BusinessDocumentId", BusinessDocumentId},
  { "UserId", UserId }
};
```

Przy tworzeniu modelu można również pominąć krok serializacji słownika i ręcznie stworzyć odpowiedni ciąg znaków:

```
var model = $@"{{""Model"": ""{{ }}"", ""RecordId"": ""{recordId}"", ""BusinessDocumentId"": {BusinessDocume
```

AKCJA ZAPISU REKORDU

Wywołanie:

```
var result = await CSharpScript.ActionRunner.SaveRecordAction.Instance.RunActionAsync(request);
```

Definicja typu parametru *request*:

```
public class SaveDocumentRequest
{
  public int? BusinessDocumentId { get; set; }
  public Dictionary<string, object> Model { get; set; }
  public string RecordId { get; set; }
}
```

- *BusinessDocumentId* (int): id definicji dokumentu
- *RecordId* (string): id dokumentu (aktualizacja); w przypadku nowego rekordu należy przekazać wartość *0*
- *Model* (Dictionary<string, object>): słownik par *klucz: wartość*, wartości pól

dokumentu biznesowego

Przykład zapisu nowego dokumentu z przekazaniem wartości *Number* (nieprawidłowej, aby wyzwolić obsługę błędu). Id dokumentu jest przypisane do zmiennej *newId*:

```
using Shared.Exceptions;
using System.Collections.Generic;
using CSharpScript.ActionRunner;

var request = new SaveDocumentRequest()
{
    RecordId = "0",
    BusinessDocumentId = BusinessDocumentId,
    Model = new Dictionary<string, object>()
    {
        { "Number", "ABC" },
    }
};

var respone = await CSharpScript.ActionRunner.SaveRecordAction.Instance.RunActionAsync(request);

if (!respone.IsValid)
{
    throw new TranslatedException("Błąd zapisu instancji dokumentu biznesowego.", respone.ValidationResult);
}
var newId = response.Result;
```

AKCJA ZMIANY STATUSU

Wywołanie:

```
var result = await CSharpScript.ActionRunner.ChangeStatusAction.Instance.RunActionAsync(request);
```

Definicja typu parametru request:

```
public class ChangeStatusRequest
{
  public int? BusinessDocumentId { get; set; }
  public string TransitionInternalId { get; set; }
  public IDictionary<string, object> Model { get; set; }
  public string RecordId { get; set; }
```

```
}
```

- *TransitionInternalId* (string): wartość pola *InternalId* przejścia, które ma zostać wykonane (*core.BusinessTransitions.InternalId*)
- *BusinessDocumentId* (int): id definicji dokumentu
- *RecordId* (string): id dokumentu (aktualizacja); w przypadku nowego rekordu należy przekazać wartość *0*
- *Model* (Dictionary<string, object>): słownik par *klucz: wartość*, umożliwia przekazanie kontekstu danych dla akcji *przed* i *po* wykonywanych podczas zmiany statusu

Przykład zmiany statusu dla bieżącego rekordu (otworzonego w formularzu, na którym znajduje się przycisk wywołujący akcję):

```
using Shared.Exceptions;
using System.Collections.Generic;
using Newtonsoft.Json;

var changeStatusRequest = new CSharpScript.ActionRunner.ChangeStatusRequest
{
  RecordId = "0",
  TransitionInternalId = Guid.NewGuid().ToString(), // Celowa nieistniejąca wartość
  BusinessDocumentId = 1
};
// Akcja może zwrócić wyjątek TranslatedException jeżeli RecordId nie jest prawidłową wartością typu INTEGER
// lub TransitionInternalId nie zostanie odnaleziony w bazie danych
var respone = await CSharpScript.ActionRunner.ChangeStatusAction.Instance.RunActionAsync(changeStatusRequest
if (!respone.IsValid)
{
  throw new TranslatedException("Błąd zmiany statusu.", respone.ValidationResult);
}
```

Powiązane tematy:

- Zadania cykliczne

6.1.41.4. LOGOWANIE KOMUNIKATÓW W AKCJI C#

Podsumowanie:
W kodzie akcji można korzystać z właściwości *Logger*, aby zapisywać komunikaty w logach systemowych.

☛ `Narzędzia` > `Akcje` > `Algorytm C#`

Logowanie komunikatów może odbywać się za pomocą metod opisanych w dokumentacji interfejsu *ILogger* z przestrzeni nazw Microsoft.Extensions.Logging.

Najniższy poziom logowania komunikatów określa ustawienie poziomu logowania w serwisie TenantsAdmin. Na przykład, jeśli ustawiony w TenantsAdmin poziom logowania to *Error*, komunikaty z akcji C# z niższym poziomem nie będą logowane.

Dostępne są następujące poziomy logowania:

- *FATAL (CRITICAL)*
- *ERROR*
- *WARNING*
- *INFO*
- *DEBUG*
- *TRACE*

UWAGA

Aby skorzystać z udostępnionego mechanizmu należy zaimportować przestrzeń nazw *Microsoft.Extensions.Logging* za pomocą dyrektywy *using* na początku kodu:

```
using Microsoft.Extensions.Logging;
```

W nowo tworzonych skryptach będzie ona dodawana automatycznie.

Przykład skryptu z wykorzystaniem mechanizmu logowania zgodny z dokumentacją przywołaną powyżej:

```
using System;
using System.Linq;
using System.Text;
using CSharpScript.ActionRunner;
using CSharpScript.SqlRunner;
using CSharpScript.WebServiceRunner;
using CSharpScript.WebServiceRunner.Models;
using Microsoft.Extensions.Logging;

Logger.LogTrace("TRACE");
Logger.LogDebug("DEBUG");
```

```
Logger.LogInformation("INFO");
Logger.LogWarning("WARNING");
Logger.LogError("ERROR");
Logger.LogCritical("CRITICAL");

Logger.LogTrace("TRACE: {recordId}", RecordId);
Logger.LogDebug("DEBUG: {recordId}", RecordId);
Logger.LogInformation("INFO: {recordId}", RecordId);
Logger.LogWarning("WARNING: {recordId}", RecordId);
Logger.LogError("ERROR: {recordId}", RecordId);
Logger.LogCritical("CRITICAL: {recordId}", RecordId);

var value = "TEXT";

try
{
    var number = int.Parse(value);
}
catch (Exception exc)
{
    Logger.LogError(exc, "Error occurred while parsing {value} to integer for record with ID {recordId}", va
}
```

6.1.41.5. ALGORYTM C# - PRZYKŁAD 1: ZMIANA STATUSU DOKUMENTÓW I WYSYŁKA E-MAIL

Podsumowanie:
W tym przykładzie używane są trzy dodatkowe klasy dostępne w nAxiom (*ActionRunner*, *WebServiceRunner*, *SqlHelper*). Akcja wyszuka wszystkie zgłoszenia, które nie są w statusie *Completed* a ich termin wykonania przypada w okresie najbliższych 7 dni. Wyszukane zgłoszenia zostaną przeniesione do odpowiedniego statusu. Jeśli akcja zmiany statusu zostanie wykonana poprawnie, zostanie wykonana akcja wysyłki e-mail w celu poinformowania odpowiednich osób.

Zamiast wywoływania akcji zmiany statusu przez żądanie REST API, można zdefiniować akcję zmiany statusu i wykonać ją w kodzie akcji C#.

```
using System;
using System.Net;
using System.Net.Http;
using System.Linq;
using System.Collections.Generic;
using Newtonsoft.Json;
```

```
using CSharpScript.ActionRunner;
using CSharpScript.SqlRunner;
using CSharpScript.WebServiceRunner;
using CSharpScript.WebServiceRunner.Models;
//gets the JSON data from the model.
var json = JsonConvert.DeserializeObject<Dictionary<string, object>>(Model);
//defines the URL and authentication credentials for the API call.
string _UrlChangeStatus = $"{TenantBaseUrl}/publicapi/Document/ChangeStatus";
string _AuthUrl = $"{TenantBaseUrl}/connect/token";
string _UserName = "jan.nowak";
string _Password = "sTr0nGp@ssw0rd";
string _Scope = "public-api";
string _ClientSecret = "sTr0nGs3cR3t";
string _ClientId = "demo-client";
// gets the ID of the business document, form, and transition for the tickets.
int ticketBusinessDocumentId = (int) await SqlHelper.ExecuteSqlScalar("SELECT Id FROM core.BusinessDocuments
int ticketFormId = (int) await SqlHelper.ExecuteSqlScalar("SELECT Id FROM core.Forms WHERE Code='Ticket-form
int ticketTransitionId = (int) await SqlHelper.ExecuteSqlScalar("SELECT Id FROM core.BusinessTransitions WHE
// gets a list of tickets that need to be changed to a different status.
var ticketsToChangeStatus = await SqlHelper.ExecuteSqlQueryRowsToDictionary($"SELECT Id FROM dbo.Ticket WHER
// iterates through the list of tickets and changes their status.
foreach (var ticket in ticketsToChangeStatus)
{
    // creates a new RestRequestModel object.
        var modelChangeTicketStatus = new RestRequestModel(){
        Method = HttpMethod.Post,
        AuthType = ApiConnector.Enums.AuthType.OAuth2Password,
        AuthUrl = _AuthUrl,
        Url = _UrlChangeStatus + $"?documentDefinitionId={ticketBusinessDocumentId}&recordId={ticket["Id"]}&
        UserName = _UserName,
        Password = _Password,
        ClientId = _ClientId,
        ClientSecret = _ClientSecret,
        Body = "{}"
    };
    // makes the API call to change the status of the ticket.
    var changeTicketStatusResponse = await RestManager.Instance.ExecuteRequestAsync(modelChangeTicketStatus)
    // checks if the API call was successful.
    if(changeTicketStatusResponse.IsSuccess == false)
    {
        // throws an exception if the API call was not successful.
        throw new Exception($"Something went wrong: {JsonConvert.SerializeObject(changeTicketStatusResponse.
```

```
    }
    // executes a SQL query to get addtional parameters for the email action from the database.
    var sqlResult = await SqlHelper.ExecuteSqlQueryRowsToDictionary($"SELECT {ticket["Id"]} AS [TicketId], '
    // creates a new dictionary to pass to the EmailAction.
    var mailActionModel = new Dictionary<string, object>
    {
        { "RecordId", null },
        { "Model", JsonConvert.SerializeObject(sqlResult.FirstOrDefault()) },
        { "UserId", UserId },
        { "BusinessDocumentId", null }
    };
    // runs the EmailAction and gets the result.
    var mailSendingResult = await EmailAction.Instance.RunActionAsync({&BaseApp.Actions.BaseModule.Example_E
    // checks if the EmailAction was successful.
    if (mailSendingResult.IsValid == false)
    {
        // This code throws an exception if the EmailAction was not successful.
        throw new Exception($"Something went wrong: {JsonConvert.SerializeObject(mailSendingResult.ResponseM
    }
}
```

6.1.41.6. ALGORYTM C# - PRZYKŁAD 2: DODANIE ZAŁĄCZNIKA

Podsumowanie:
Poniższy kod akcji C# pozwala na dodanie załącznika do dokumentu. Należy określić identyfikator dokumentu oraz identyfikator definicji dokumentu.

```
using System;
using System.Linq;
using System.Net.Http;
using System.Threading.Tasks;
using CSharpScript.WebServiceRunner.Models;

var baseUri = new System.Uri(TenantBaseUrl);
var _endpointUri = new System.Uri(baseUri, "publicapi/attachment/uploadAttachment");
var _authUri = new System.Uri(baseUri, "connect/token");
var _userName = "tutaj nazwa użytkownika, który ma dostęp do public-api";
var _userPassword = "tutaj hasło użytkownika";
var _scope = "public-api";
var _clientSecret = "secret";
var _clientId = "openjs";
```

```
var recordId = 0; //tutaj podajemy id instancji dokumentu biznesowego, id rekordu
var businessDocumentId = 0; //tutaj podajemy id definicji dokumentu biznesowego
var overwrite = false; //czy załącznik ma zostać nadpisany? należy ustawić na true jeżeli instancja dokument
var pathToFile = @"C:\temp\plik_tekstowy.txt"; //ścieżka do pliku załącznika
var attachmentFileName = System.IO.Path.GetFileName(pathToFile); //nazwa załącznika w dokumencie biznesowym,

var request = new RestRequestModel()
{
    Method = HttpMethod.Post,
    Url = $"{_endpointUri.AbsoluteUri}?recordId={recordId}&businessDocumentId={businessDocumentId}&overwrite
    AuthType = ApiConnector.Enums.AuthType.OAuth2Password,
    AuthUrl = _authUri.AbsoluteUri,
    UserName = _userName,
    Password = _userPassword,
    ClientId = _clientId,
    ClientSecret = _clientSecret,
    Scope = _scope,
};

var boundary = System.Guid.NewGuid().ToString();
using (var client = new HttpClient())
using (var formData = new MultipartFormDataContent(boundary))
{
    client.DefaultRequestHeaders.Accept.Clear();
    if (request.Headers?.Any() == true)
    {
        foreach (var item in request.Headers)
        {
            client.DefaultRequestHeaders.Add(item.Key, item.Value);
        }
    }
    client.Timeout = TimeSpan.FromMinutes(10);

    formData.Headers.Remove("Content-Type");
    formData.Headers.TryAddWithoutValidation("Content-Type", "multipart/form-data; boundary=" + boundary);

    await PrepareAuth(client, request.AuthUrl, request.UserName, request.Password, request.ClientId, request

    var fileContent = await ReadFile(pathToFile);
    HttpContent bytesContent = new ByteArrayContent(fileContent);
    formData.Add(bytesContent, "file", attachmentFileName);
```

```
    var response = await client.PostAsync(request.Url, formData);
    response.EnsureSuccessStatusCode();
    string responseBody = await response.Content.ReadAsStringAsync();
    NLog.LogManager.GetCurrentClassLogger().Trace(responseBody);
}

private async Task<byte[]> ReadFile(string pathToFile)
{
    using (var stream = System.IO.File.Open(pathToFile, System.IO.FileMode.Open))
    {
        byte[] result = new byte[stream.Length];
        await stream.ReadAsync(result, 0, (int)stream.Length);
        return result;
    }
}

private async Task PrepareAuth(HttpClient client, string authServer, string userName, string password, strin
{
    var token = await GetBearerToken(authServer, userName, password, clientId, clientSecret, scope);
    client.DefaultRequestHeaders.Authorization = new System.Net.Http.Headers.AuthenticationHeaderValue("Bear
}

private async Task<string> GetBearerToken(string authServer, string userName, string password, string client
{
    var client = new HttpClient();
    client.DefaultRequestHeaders.Accept.Clear();
    client.Timeout = TimeSpan.FromMinutes(10);

    var contentValues = new System.Collections.Generic.Dictionary<string, string>();
    contentValues.Add("grant_type", "password");

    if (!string.IsNullOrWhiteSpace(clientId))
    {
        contentValues.Add("client_id", clientId);
    }

    if (!string.IsNullOrWhiteSpace(clientSecret))
    {
        contentValues.Add("client_secret", clientSecret);
    }

    if (!string.IsNullOrWhiteSpace(scope))
```

```
    {
        contentValues.Add("scope", scope);
    }

    if (!string.IsNullOrWhiteSpace(password))
    {
        contentValues.Add("password", password);
    }

    if (!string.IsNullOrWhiteSpace(userName))
    {
        contentValues.Add("username", userName);
    }

    var content = new FormUrlEncodedContent(contentValues);

    var httpResponse = await client.PostAsync(authServer, content, default);
    if (httpResponse == null)
    {
        throw new Shared.Exceptions.TranslatedException("Api.ActionModule.ConnectionFailed");
    }

    var response = await httpResponse.Content.ReadAsStringAsync();
    dynamic objectResponse = Newtonsoft.Json.JsonConvert.DeserializeObject(response);

    if (objectResponse == null || objectResponse.access_token == null)
    {
        throw new Exception(response);
    }

    return objectResponse.access_token;
}
```

6.1.41.7. ALGORYTM C# - PRZYKŁAD 3: POBRANIE ZAŁĄCZNIKA

Podsumowanie:
Poniższy kod akcji C# pobiera załącznik o podanym identyfikatorze i zapisuje do pliku w określonej lokalizacji.

```
using System;
using System.Linq;
using System.Net.Http;
```

```
using System.Threading.Tasks;
using CSharpScript.WebServiceRunner.Models;

var baseUri = new System.Uri(TenantBaseUrl);
var _endpointUri = new System.Uri(baseUri, "publicapi/attachment/downloadAttachment");
var _authUri = new System.Uri(baseUri, "connect/token");
var _userName = "tutaj nazwa użytkownika, który ma dostęp do public-api";
var _userPassword = "tutaj hasło użytkownika";
var _scope = "public-api";
var _clientSecret = "secret";
var _clientId = "openjs";


var attachmentId = 0; //tutaj podajemy id załącznika, którego plik chcemy pobrać
var pathToFile = @"C:\temp\jakis plik.ext"; //ścieżka zapisu załącznika wraz z nazwą pliku

var request = new CSharpScript.WebServiceRunner.Models.RestRequestModel()
{
    Method = System.Net.Http.HttpMethod.Get,
    Url = $"{_endpointUri.AbsoluteUri}?id={attachmentId}",
    AuthType = ApiConnector.Enums.AuthType.OAuth2Password,
    AuthUrl = _authUri.AbsoluteUri,
    UserName = _userName,
    Password = _userPassword,
    ClientId = _clientId,
    ClientSecret = _clientSecret,
    Scope = _scope,
};

using (var client = new HttpClient())
{
    await PrepareAuth(client, request.AuthUrl, request.UserName, request.Password, request.ClientId, request

    var response = await client.GetAsync(request.Url);
    response.EnsureSuccessStatusCode();
    var fileContent = await response.Content.ReadAsByteArrayAsync();
    await SaveFile(pathToFile, fileContent);
}

private Task SaveFile(string pathToFile, byte[] fileContent)
{
    return System.IO.File.WriteAllBytesAsync(pathToFile, fileContent);
```

```
}

private async Task PrepareAuth(HttpClient client, string authServer, string userName, string password, strin
{
    var token = await GetBearerToken(authServer, userName, password, clientId, clientSecret, scope);
    client.DefaultRequestHeaders.Authorization = new System.Net.Http.Headers.AuthenticationHeaderValue("Bear
}

private async Task<string> GetBearerToken(string authServer, string userName, string password, string client
{
    var client = new HttpClient();
    client.DefaultRequestHeaders.Accept.Clear();
    client.Timeout = TimeSpan.FromMinutes(10);

    var contentValues = new System.Collections.Generic.Dictionary<string, string>();
    contentValues.Add("grant_type", "password");

    if (!string.IsNullOrWhiteSpace(clientId))
    {
        contentValues.Add("client_id", clientId);
    }

    if (!string.IsNullOrWhiteSpace(clientSecret))
    {
        contentValues.Add("client_secret", clientSecret);
    }

    if (!string.IsNullOrWhiteSpace(scope))
    {
        contentValues.Add("scope", scope);
    }

    if (!string.IsNullOrWhiteSpace(password))
    {
        contentValues.Add("password", password);
    }

    if (!string.IsNullOrWhiteSpace(userName))
    {
        contentValues.Add("username", userName);
    }
```

```
    var content = new FormUrlEncodedContent(contentValues);

    var httpResponse = await client.PostAsync(authServer, content, default);
    if (httpResponse == null)
    {
        throw new Shared.Exceptions.TranslatedException("Api.ActionModule.ConnectionFailed");
    }

    var response = await httpResponse.Content.ReadAsStringAsync();
    dynamic objectResponse = Newtonsoft.Json.JsonConvert.DeserializeObject(response);

    if (objectResponse == null || objectResponse.access_token == null)
    {
        throw new Exception(response);
    }

    return objectResponse.access_token;
}
```

6.1.41.8. ALGORYTM C# - PRZYKŁAD 4: USUNIĘCIE ZAŁĄCZNIKA

Podsumowanie:
Poniższy kod akcji C# usuwa wszystkie wersje załącznika o podanym identyfikatorze.

```
var baseUri = new System.Uri(TenantBaseUrl);
var _endpointUri = new System.Uri(baseUri, "publicapi/attachment/removeAttachment");
var _authUri = new System.Uri(baseUri, "connect/token");
var _userName = "tutaj nazwa użytkownika, który ma dostęp do public-api";
var _userPassword = "tutaj hasło użytkownika";
var _scope = "public-api";
var _clientSecret = "secret";
var _clientId = "openjs";

var attachmentId = 0; //<-- tutaj podajemy id załącznika, który chcemy usunąć
                      //poniższy kod usuwa załącznik o podanym id wraz ze wszystkimi wersjami

var request = new CSharpScript.WebServiceRunner.Models.RestRequestModel()
{
    Method = System.Net.Http.HttpMethod.Delete,
    Url = $"{_endpointUri.AbsoluteUri}?id={attachmentId}",
    AuthType = ApiConnector.Enums.AuthType.OAuth2Password,
    AuthUrl = _authUri.AbsoluteUri,
```

```
    UserName = _userName,
    Password = _userPassword,
    ClientId = _clientId,
    ClientSecret = _clientSecret,
    Scope = _scope,
};

var response = await CSharpScript.WebServiceRunner.RestManager.Instance.ExecuteRequestAsync(request);

if(!response.IsSuccess)
{
    throw new System.Net.WebException(response.RawResult);
}
```

6.1.42. Importowanie dokumentów

Podsumowanie:
Akcja pozwala importować dane z tabeli pomocniczej do tabeli dokumentu biznesowego w modelu standardowym lub uproszczonym. W akcji należy zdefiniować zapytanie SQL zwracające dane oraz skonfigurować mapowanie kolumn. Dane zwracane przez zapytanie SQL mogą także zostać użyte do zasilenia kolumn systemowych. Możliwe jest także wypełnienie tych kolumn stałymi wartościami.

☛ *AdminSPA* > `NARZĘDZIA` > `Akcje` > `Importowanie dokumentów`

Oprócz właściwości wspólnych dla wszystkich akcji, w akcji importu dokumentów należy określić następujące właściwości:

`SQL`: zapytanie zwracające dane z tabeli pomocniczej, którymi zostaną wypełnione kolumny użytkownika i kolumny systemowe w tabeli powiązanej ze wskazaną definicją dokumentu. W zapytaniu można używać składni *smart numbers* oraz zmiennych odwołujących się do kontekstu formularza (*{@Nazwa}*).

WAŻNE
Zapytanie musi zwrócić kolumnę z identyfikatorem importowanych rekordów na potrzeby audytu/raportowania.

`Kolumny dostępne do mapowania na pola dokumentu`: nazwy kolumn, które należy powiązać z kolumnami w tabeli dokumentu biznesowego. Do nazw zwracanych przez zapytanie jest dodawany przedrostek *Import_*.

`Kolumna z identyfikatorem rekordu do zapisu informacji przetwarzania`: nazwa kolumny z identyfikatorem rekordu, który zostanie zapisany w tabeli audytu.

`Dokument biznesowy`: nazwa definicji dokumentu, do której tabeli zostaną zaimportowane dane z tabeli pomocniczej.

Tabela mapowania kolumn:

`Parametry użytkownika/systemowe`:

- `Nazwa parametru`: nazwa kolumny użytkownika/systemowej w docelowej tabeli dokumentów.
- `Wartość`: pole wprowadzania typy *txt/sql*, w którym można wpisać:
  - odwołanie do kolumny w tabeli źródłowej w formie *{@Import_nazwa}*
  - wartość stałą, odwołanie do kontekstu formularza, wyrażenie *smart numbers*
  - zapytanie SQL

WAŻNE

- Wartości statusu dla importowanych dokumentów muszą na diagramie procesu odpowiadać blokom *Zadanie ręczne* lub *Oczekiwanie*. Statusy innych bloków nie są obsługiwane.
- Jeśli kolumna *Status* w tabeli docelowej nie zostanie wypełniona, akcja wypełni ją identyfikatorem statusu dla pierwszego kroku wychodzącego z bloku *Start*.
- Ze względu na wydajność, podczas importu dokumentów nie są wykonywane akcje przypisane do przejść na diagramie procesu dla danej definicji dokumentu.
- Jeśli zapytanie może zwracać wartości *NULL*, do mapowania kolumn należy używać trybu SQL; w przeciwnym razie import nie powiedzie się (błąd konwersji), a w przypadku danych tekstowych w pola docelowe zostanie wpisany literał tekstowy *NULL*.

`Transakcyjność wykonywania akcji`:

- `wszystkie rekordy`: import wszystkich rekordów zwróconych przez zapytanie wykonuje się w transakcji; niepowodzenie w imporcie jednego rekordu oznacza niepowodzenie całej transakcji.
- `rekord po rekordzie`: niepowodzenie importu jednego rekordu nie ma wpływu na import pozostałych rekordów.

HISTORIA IMPORTU

Historia importu dokumentów biznesowych jest zapisywana w dwóch tabelach systemowych:

core.BatchImportHistories:

- *BatchProcessId*: GUID danego procesu importu
- *BatchId*: identyfikator zbioru akcji z akcją importu
- *ImportedBy*: GUID użytkownika wywołującego akcję
- *ImportDatetime*: czas wywołania akcji
- *ImportActionRowGuid*: GUID użytej akcji importu dokumentów
- *TargetBusinessDocumentGUID*: GUID docelowej definicji dokumentu biznesowego

core.BatchImportHistoryElements:

- *Id*: klucz główny
- *BatchImportHistoryId*: GUID danego procesu importu
- *SourceRecordId*: identyfikator importowanego rekordu; musi być zwracany przez zapytanie SQL w akcji\
- *BatchImportStatus*: status importu (wartości: *Success*, *Error*, *Skip*)
- *ErrorMessage*: komunikat o błędzie

Operacja importu dokumentów jest rejestrowana w danych audytowych dokumentów biznesowych w ramach zdarzenia *Dokument - zaimportowanie*. To zdarzenie jest domyślnie włączone dla nowych i już istniejących dokumentów.

ZALECENIA DOTYCZĄCE WYDAJNOŚCI

- Przygotować dane źródłowe w taki sposób, aby zawierały numer dokumentu, i zmapować te dane na kolumnę *Code*; funkcja autonumeracji znacząco obniża wydajność operacji wsadowego importu dokumentów.

  > **UWAGA**
  > Numer dokumentu musi być niepowtarzalny w całej tabeli modelu danych.

- Mapując wartości importowanych danych na kolumny tabeli dokumentu nie należy korzystać z trybu SQL, który generuje dodatkowe podzapytania obniżające wydajność; kolumny zapytania SQL zwracającego dane źródłowe (*{@Import_nazwa}*) są dostępne również w trybie TXT. Na przykład zamiast : `<SQL> SELECT 'Code: ' + {@Import_DocNo}` należy użyć metody `<TXT> Code: {@Import_DocNo}`.
- W przypadku, kiedy wymagane są bardziej zaawansowane operacje na importowanych danych, zaleca się wykonanie tych operacji na etapie przygotowania danych do importu.

- W razie problemów z wydajnością akcji importu przy włączonej obsłudze uprawnień ACL zaleca się, aby import wielu dokumentów wykonywać w paczkach jako zadanie cykliczne.

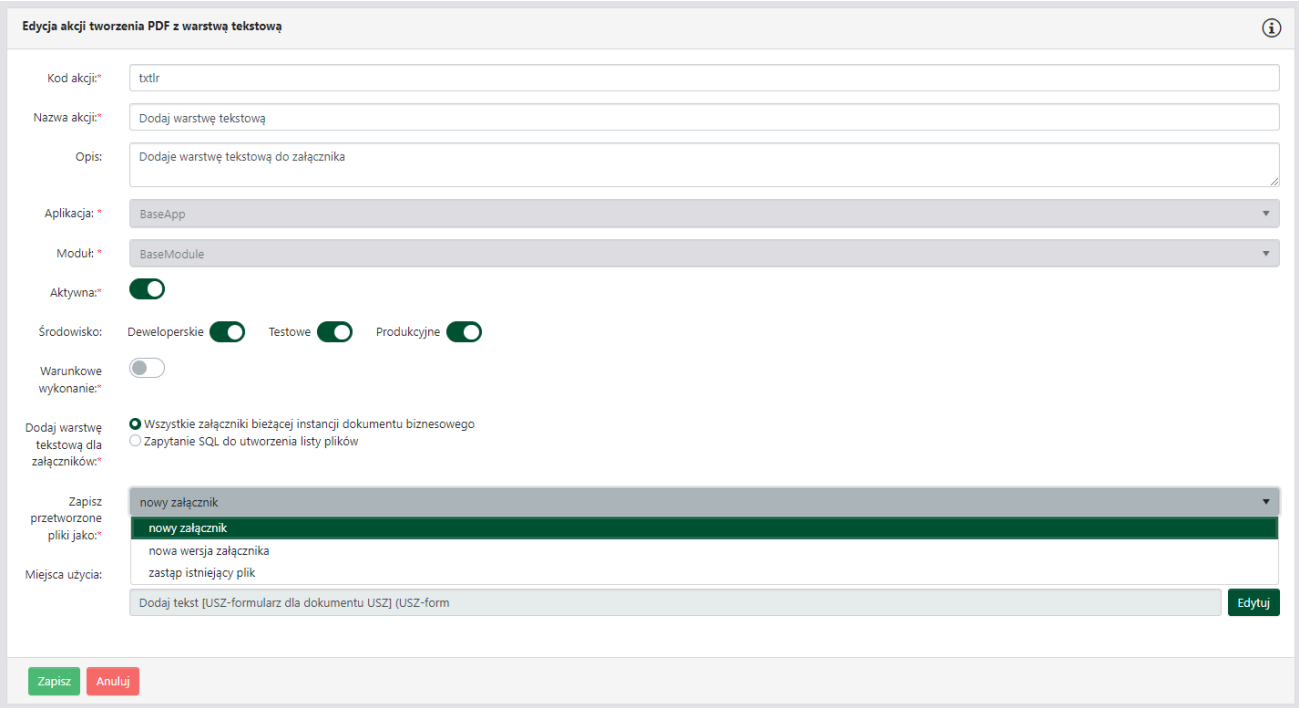
Powiązane tematy:

- Kreator modeli danych
- Smart numbers
- Zadania cykliczne
- Definicja dokumentu, Audyt

6.1.43. Utwórz PDF z warstwą tekstową

Podsumowanie:
Akcja Utwórz PDF z warstwą tekstową konwertuje załącznik (obraz lub PDF bez warstwy tekstowej) w plik PDF z warstwą tekstową, to jest taki, w którym działa wyszukiwanie tekstowe. Dodatkowo akcja dodaje rozpoznany tekst do indeksu wyszukiwarki Elastic Search.

☛ *AdminSPA* > `Narzędzia` > `Akcje` > `Utwórz PDF z warstwą tekstową`



Akcja Utwórz PDF z warstwą tekstową

Oprócz właściwości wspólnych dla wszystkich akcji, w akcji należy określić następujące właściwości:

- `Dodaj warstwę tekstową dla załączników`: dostępne opcje:
  - `Wszystkie załączniki bieżącej instancji dokumentu biznesowego`: akcja przetwarza wszystkie załączniki w dokumencie (z wszystkich kategorii),
  - `Zapytanie SQL do utworzenia listy plików`: należy napisać zapytanie SQL, które zwróci identyfikatory załączników do przetworzenia przez akcję.
- `Zapisz przetworzone pliki jako`: dostępne wartości:
  - `nowy załącznik`: zapisuje wynikowy PDF z warstwą tekstową jako nowy załącznik w tej samej kategorii,
  - `nowa wersja załącznika`: zapisuje wynikowy PDF z warstwą tekstową jako nową wersję załącznika,
  - `zastąp istniejący plik`: zapisuje plik wynikowy w miejsce oryginalnego załącznika.

Akcja jest wykonywana asynchronicznie. Po uruchomieniu dodaje wpisy do tabeli *core.OCRProcessesQueue*. Ta tabela jest odczytywana przez serwis *taskservice*, który dla nowych lub wymagających ponownienia rekordów wysyła żądania dodania warstwy tekstowej do serwisu *ocrapi*. W razie trzykrotnego niepowodzenia, w tabeli kolejki jest logowany komunikat o błędzie z *ocrapi*.

WAŻNE
Domyślnie sprawdzanie kolejki procesów OCR jest wyłączone. Aby je włączyć, należy ustawić wartość parametru `Częstotliwość sprawdzania kolejki procesów wykonywanych przez OCR (w sekundach)` (*adminSPA* > `Ustawienia systemu` > `OCR`) większą od 0.

UWAGA
W procesie dodawania warstwy tekstowej używany jest folder *ocr_others*, którego lokalizację określa klucz *PathToProcessingFiles* w pliku *appsettings.json* serwisu *ocrapi*. W razie niepowodzenia przetwarzania należy sprawdzić, czy folder istnieje i czy wpis w pliku *appsettings.json* jest prawidłowy.

Akcja wykorzystuje środowisko uruchomieniowe ABBY FineReader Engine 12.5.6 z odpowiednimi licencjami, patrz Licencje Abby FR Engine .

Powiązane tematy:
- Wspólne parametry akcji
- Częstotliwość sprawdzania kolejki przetwarzania OCR

6.1.44. Akcja odczytu kodów QR/BC z plików

Podsumowanie:
Akcja obsługuje załączniki w formatach .png, .jpeg, .jpg, .pdf i jest licencjonowana osobno (klucz licencji *QR_BC*).

☛ *AdminSPA* > `NARZĘDZIA` > `Akcje` > `Odczyt kodów QR/BC z plików`

Oprócz właściwości wspólnych dla wszystkich akcji, w akcji odczytu kodów QR/BC należy określić:

- `Rodzaj kodów...`: formaty kodów używane w przetwarzanych dokumentach, zarówno BC, jak i QR.
- `Zapytanie SQL do wyboru plików`: zapytanie, który zwróci identyfikatory plików załączników do przetworzenia, na przykład:

```
SELECT [Id]
FROM core.Attachments
WHERE [RecordId] = {@_Id}
AND [BusinessDocumentId] = {@BusinessDocId}
```

- `Procedura odpowiedzi wyników skanowania`: zapytanie które przetworzy wyniki zwracane przez akcję w postaci dwóch zmiennych JSON:
    - *@_ReadCodes*: tablica odczytanych wartości kodów QR/BC z poszczególnych plików:

```
[
 {
 "attachmentFileName": "some_attachment.pdf",
 "attachmentId": 123,
 "attachmentVersion": 1,
 "code": "Some code value",
 "type":"QR",
 "typeDetails": "QR_CODE",
 "pageNumber": 1
 },

]
```

Gdzie *code* to odczytana wartość kodu, *type* to typ kodu (QR lub BC), *typeDetails* to format kodu, a *pageNumber* określa numer strony tylko dla plików PDF.

- *@_ProcessingResult*: zawiera informacje o statusie przetwarzania poszczególnych załączników przeznaczone do własnej obsługi błędów:

```
[
 {
 "attachmentFileName": "Nowy Dokument tekstowy.txt",
 "attachmentId": 1205,
 "attachmentVersion": 1,
 "errorMessage": "Nieobsługiwany format pliku '.txt'. System obsługuje następujące formaty.
 }
]
```

Zwrócone zmienne można na przykład zapisać w tabeli pomocniczej:

```
INSERT INTO [dbo].[QrBcProcessingResults]([ProcessingResult], [ReadCodes])
VALUES({@_ProcessingResult}, {@_ReadCodes})
```

Można także je przetworzyć przy użyciu funkcji SQL *OPENJSON()*.

UWAGA

Na skuteczność działania akcji duży wpływ ma jakość źródłowych obrazów. Minimalna rozdzielczość nie może być mniejsza nić 200 DPI, obraz powinien być ostry, kontrastowy równomiernie oświetlony. Minimalna gęstość pikseli na moduł (np. pasek lub odstęp w kodzie BC) powinna wynosić 3 piksele dla kodów 1D (BC) i 5 pikseli dla kodów 2D (QR i podobne).

Dodatkowo na wyniki wpływa wybór obsługiwanych formatów — zaleca się ograniczenie liczby formatów do używanych najczęściej w przetwarzanych dokumentach.

Powiązane tematy:

- Wspólne parametry akcji

6.1.45. Uruchom model decyzyjny

Podsumowanie:
Ta akcja służy do uruchamiania modelu decyzyjnego . Domyślnie akcja uruchamia najnowszą opublikowaną wersję modelu. Akcja może przesłać dane wejściowe do modelu z kontekstu rekordu lub zwrócone przez zapytanie SQL. Wynik uruchomienia modelu jest zwracany jako obiekt JSON, który można przetworzyć zapytaniem SQL.

Oprócz właściwości wspólnych dla wszystkich akcji, w tej akcji należy określić następujące właściwości:

- `Model decyzyjny`: nazwa modelu, który ma zostać uruchomiony.
- `Wersja modelu decyzyjnego`: dostępne wartości:
  - `Najnowsza opublikowana`
  - `Wskazana wersja`: uruchamia wersję wybraną z listy rozwijanej.
- `Dane wejściowe w wybranej wersji modelu`: pole tylko do odczytu z listą nazw zmiennych wejściowych zdefiniowanych w modelu.
- `Przekazywany zestaw danych wyjściowych`: dostępne wartosci:
  - `Pobierz kontekst rekordu`: przekazuje do modelu dane z kontekstu formularza, z którego jest uruchamiana akcja.
  - `Zdefiniuj przez zapytanie SQL`: przekazuje do modelu dane zwrócone przez zapytanie SQL zdefiniowane w dostępnym oknie; zapytanie powinno zwrócić jeden rekord, nazwy zwróconych kolumn zostaną dopasowane do nazw zmiennych wejściowych dostępnych w modelu.
- `Dane wyjściowe wywoływanej decyzji`: pole tylko do odczytu z listą nazw zmiennych wyjściowych zwróconych przez model po jego wykonaniu; te nazwy muszą być takie same jak nazwy w kontekście formularza lub nazwy zwracane przez zapytanie SQL.
- `Obsługa danych wyjściowych`: zapytanie SQL przetwarzające dane wyjściowe z modelu decyzyjnego; dane są zwracane jako obiekt JSON o nazwie *@_DecisionResult*, do wyodrębnienia danych z tego obiektu można użyć funkcji SQL OPENJSON.

Powiązane tematy:

- Modele decyzyjne DMN

6.1.46. Akcje systemowe

Podsumowanie:
Akcja systemowe to grupa akcji konfigurowanych lokalnie w miejscach użycia: na przyciskach w formularzach i widokach listach, a także na przejściach diagramu procesów jako akcje przed zmianą statusu i po. Konfiguracje akcji systemowych są zapisywane w dedykowanych tabelach: *core.FormActions*, *core.FormSectionActions*, *core.FormFieldActions* i *core.GridButtonAction*. Parametry konfiguracyjne akcji systemowej są zapisywane w tych tabelach w formacie JSON w kolumnie *Parameters*.

Definicje akcji systemowych są zapisane wraz z pozostałymi akcjami w tabeli *core.Actions*, wyróżnikiem tych akcji jest wartość *5* w kolumnie *Type*. Akcje systemowe są domyślnie przypisane do systemowego modułu *BaseModule* (aplikacja *BaseApp*).

```
/****** Script for SelectTopNRows command from SSMS  ****
SELECT [Id]
      ,[Name]
      ,[Command]
      ,[IsActive]
      ,[Type]
FROM [naxiom155].[core].[Actions]
WHERE Type = 5
```

00 %

Results | Messages

	Id	Name	Command	IsActive	Type
1	1	Odśwież stronę	RefreshPage	1	5
2	2	Zamknij formularz	CloseForm	1	5
3	3	Zapisz rekord	SaveRecord	1	5
4	4	Usuń rekord	DeleteRecord	1	5
5	5	Odśwież model	RefreshModel	1	5
6	6	Otwórz formularz	OpenForm	1	5
7	7	Odśwież formularz	RefreshForm	1	5
8	9	Wyloguj	LogOut	1	5
9	19	Ustaw dane formularza	SetDataOnForm	1	5
10	28	Odśwież wskazany formularz	RefreshIndicatedForm	1	5
11	29	Zmień status archiwizacji	SetArchiveStatus	1	5
12	30	Blokuj dokument	BlockDocument	1	5
13	31	Odblokuj dokument	UnblockDocument	1	5
14	43	Anonimizuj dane	AnonimizeData	1	5
15	46	Zmiana statusu dokumentu	SetWorkflowStatus	1	5
16	63	Odśwież listę	RefreshGrid	1	5

Akcje systemowe w tabeli *core.Actions*

Dla przycisku formularza dostępne są wszystkie akcje systemowe. Podobnie dla przejść diagramu procesu. W przypadku przycisków w widoku listy (zarówno w nagłówku, jak i w

wierszu) dostępne są tylko akcje zamknij/otwórz/odśwież formularz, odśwież wskazany formularz i odśwież listę. Dla zdarzeń otwarcia i zamknięcia formularza są dostępne akcje, odpowiednio, zablokowania i odblokowania dokumentu.

Akcje systemowe są niedostępne dla zdarzeń systemowych, zdarzeń dokumentu biznesowego i zdarzeń audytu.

W kolejnych tematach opisano poszczególne akcje systemowe.

Powiązane tematy:

- Przypisywanie akcji do przycisku
- Przypisywanie akcji do przycisku w widoku listy
- Akcje zdarzeń systemowych
- Przypisywanie akcji do zdarzeń audytu
- Przypisywanie akcji do zdarzeń dokumentu

6.1.46.1. ODŚWIEŻ STRONĘ

Podsumowanie:
Akcja odświeża bieżący formularz.

Działanie jest analogiczne jak kliknięcie łącza otwierającego formularz, kliknięcie przycisku odświeżania formularza w prawym górnym rogu lub wykonanie akcji odśwież formularz.

Akcja odświeża dane wyświetlane w formularzu zgodnie ze stanem rekordu w bazie danych. W przypadku nowego rekordu, akcja powoduje wyczyszczenie niezapisanych danych i wyświetla pola puste lub z wartościami domyślnymi.

Akcję można przypisać tylko do przycisku na formularzu i przejścia na diagramie procesu.

Akcja nie ma żadnych parametrów.

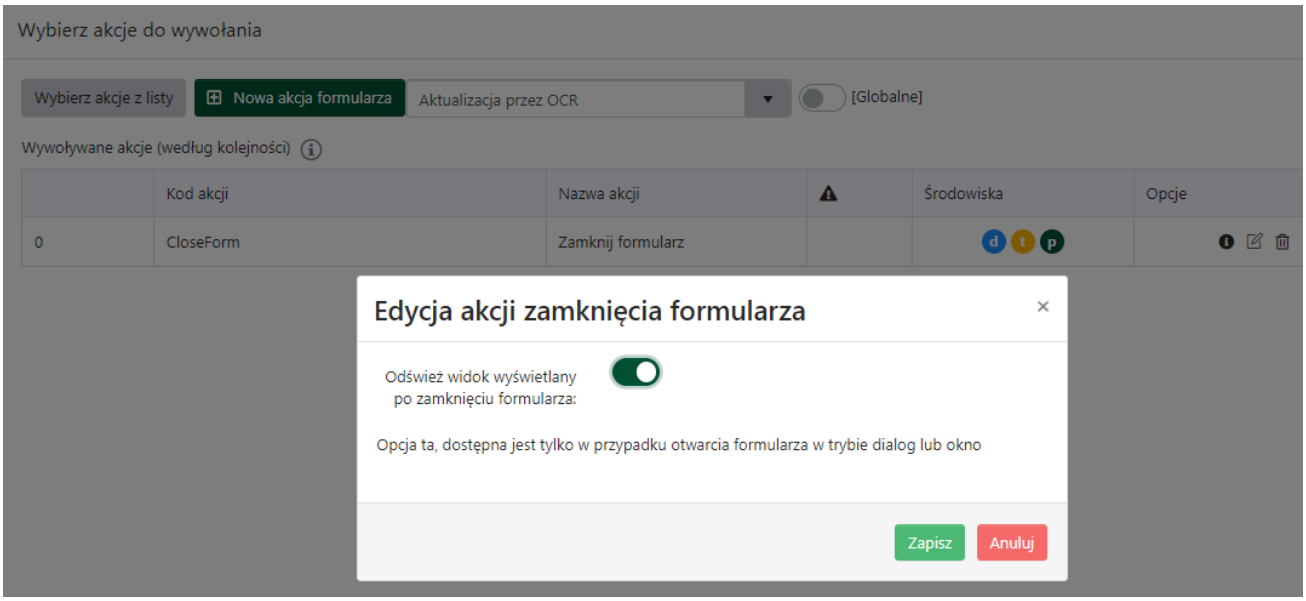
6.1.46.2. ZAMKNIJ FORMULARZ

Podsumowanie:
Akcja zamyka bieżący formularz.

Wywolanie akcji powoduje wyświetlenie widoku sprzed otwarcia formularza, np. widoku listy, pulpitu lub innego formularza.

Akcję można przypisać do przycisku na formularzu i widoku listy oraz do przejścia na diagramie procesu.

O ile zamykany formularz był otwarty w trybie *dialog* lub *okno*, w akcji można włączyć opcję odświeżania widoku wyświetlanego po zamknięciu formularza. Jest to przydatne, gdy w zamykanym formularzu zmieniono dane w formularzu wyświetlanym „pod spodem”.



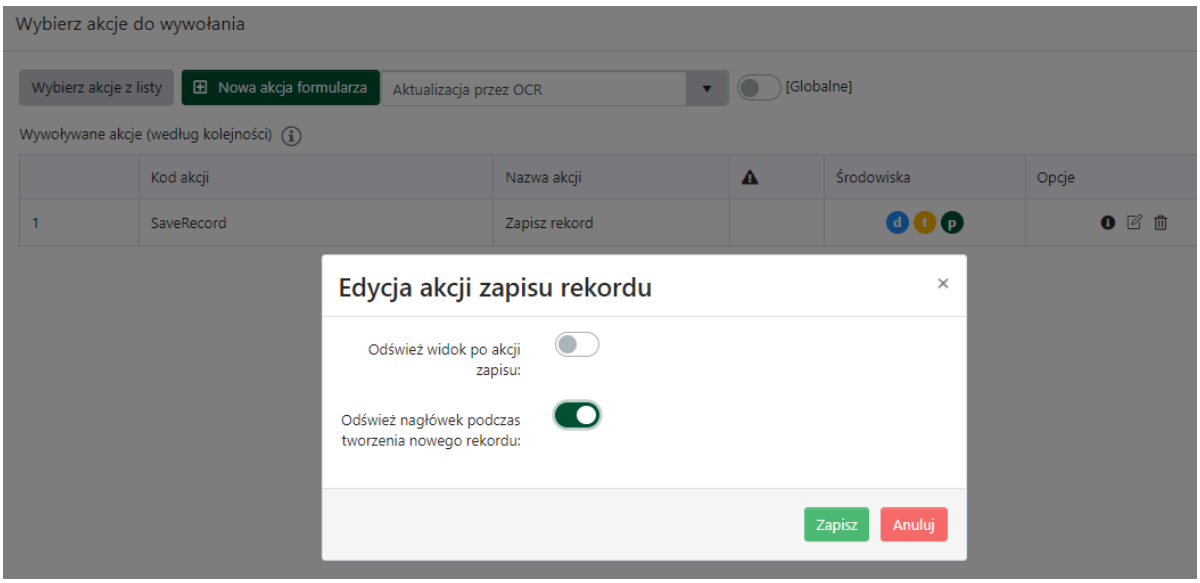
Opcje zamykania formularza

6.1.46.3. ZAPISZ REKORD

> **Podsumowanie:**
> Akcja zapisuje w bazie danych wartości z pól aktywnego formularza. W definicji akcji można używać przełączników umożliwiających sterowanie odświeżaniem formularza po akcji zapisu. Szczegółowe zachowanie nAxiom dla różnych konfiguracji zapisu rekordu przedstawiono na diagramach sekwencji w tym temacie.

Domyślnie wraz z zapisem wykonywane jest odświeżenie formularza. Obejmuje ono odświeżenie wartości wyliczanych, zależności oraz odświeżenie sekcji innych niż pola formularza, w tym nagłówka (sekcja toolbar). W niektórych przypadkach może to być niepożądane. Dlatego akcja zapisz rekord ma dodatkowe przełączniki:

- `Odśwież widok po akcji zapisu`: jeśli zostanie wyłączony, dane z formularza zostaną zapisane, ale widok formularza nie zostanie odświeżony.
- `Odśwież nagłówek podczas tworzenia nowego rekordu`: pozwala włączyć odświeżanie sekcji toolbar po zapisie nowego dokumentu, kiedy wyłączono odświeżanie formularza.



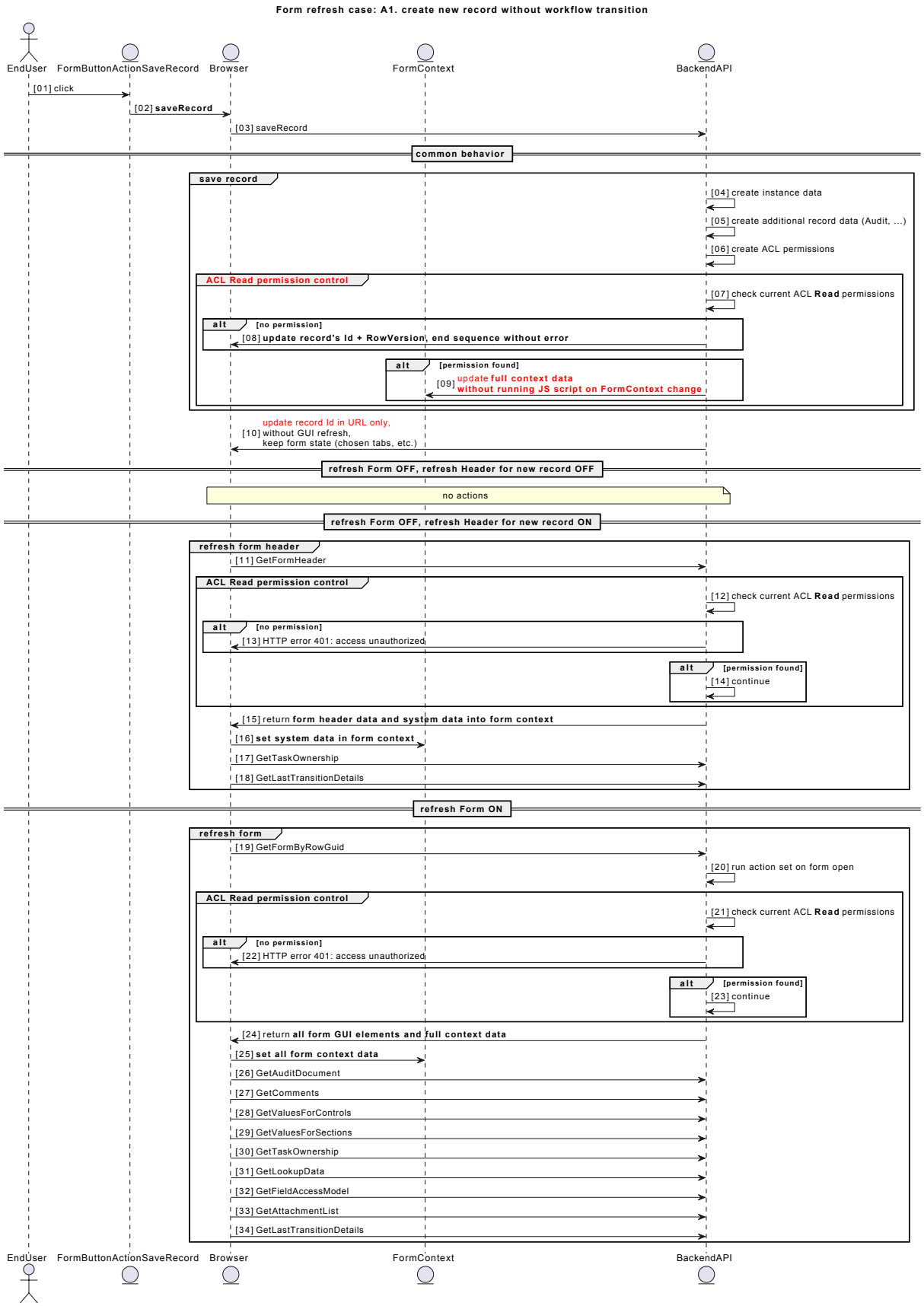
Edycja akcji zapisu rekordu

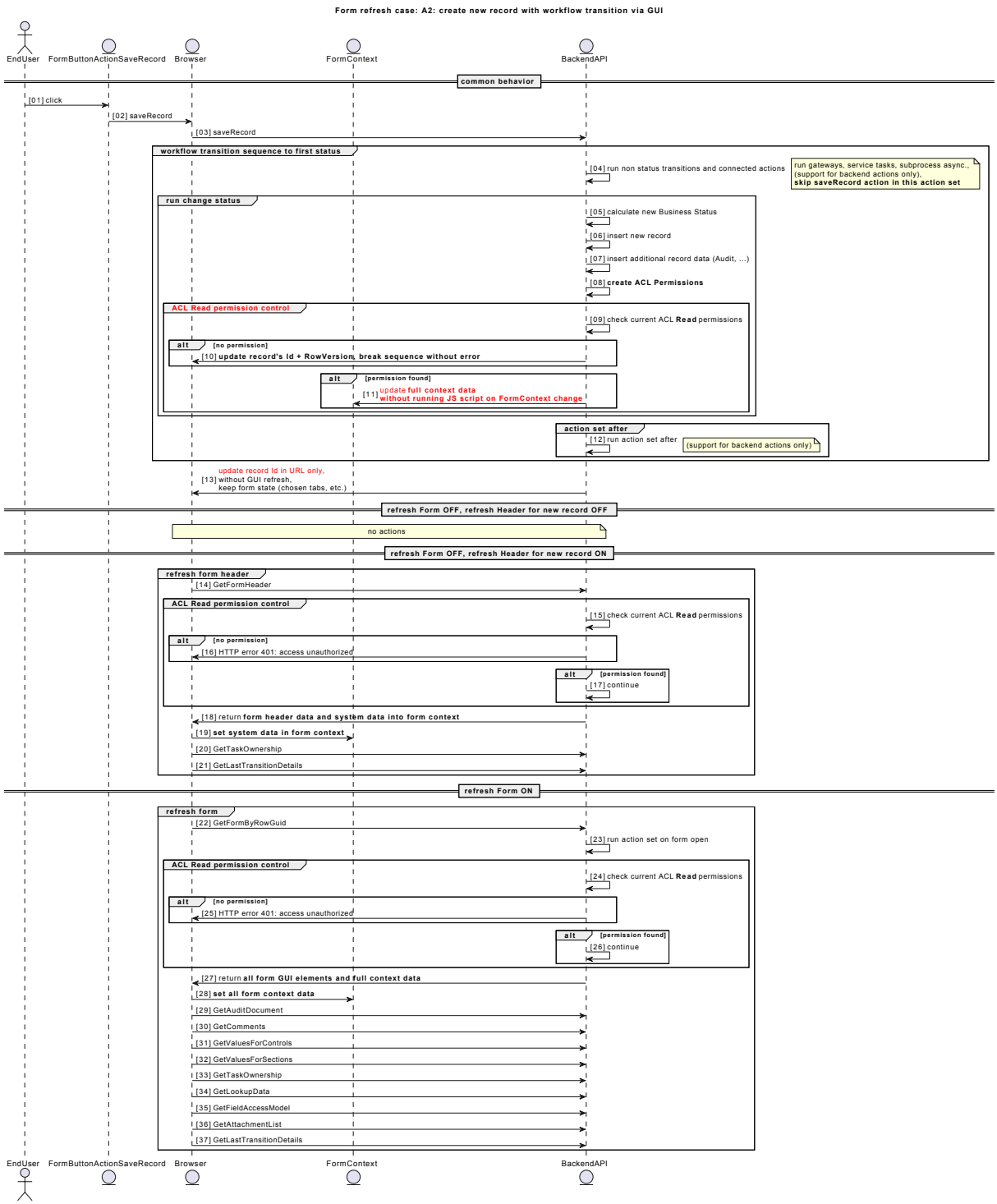
Akcję można przypisać do przycisku na formularzu i do przejścia na diagramie procesu.

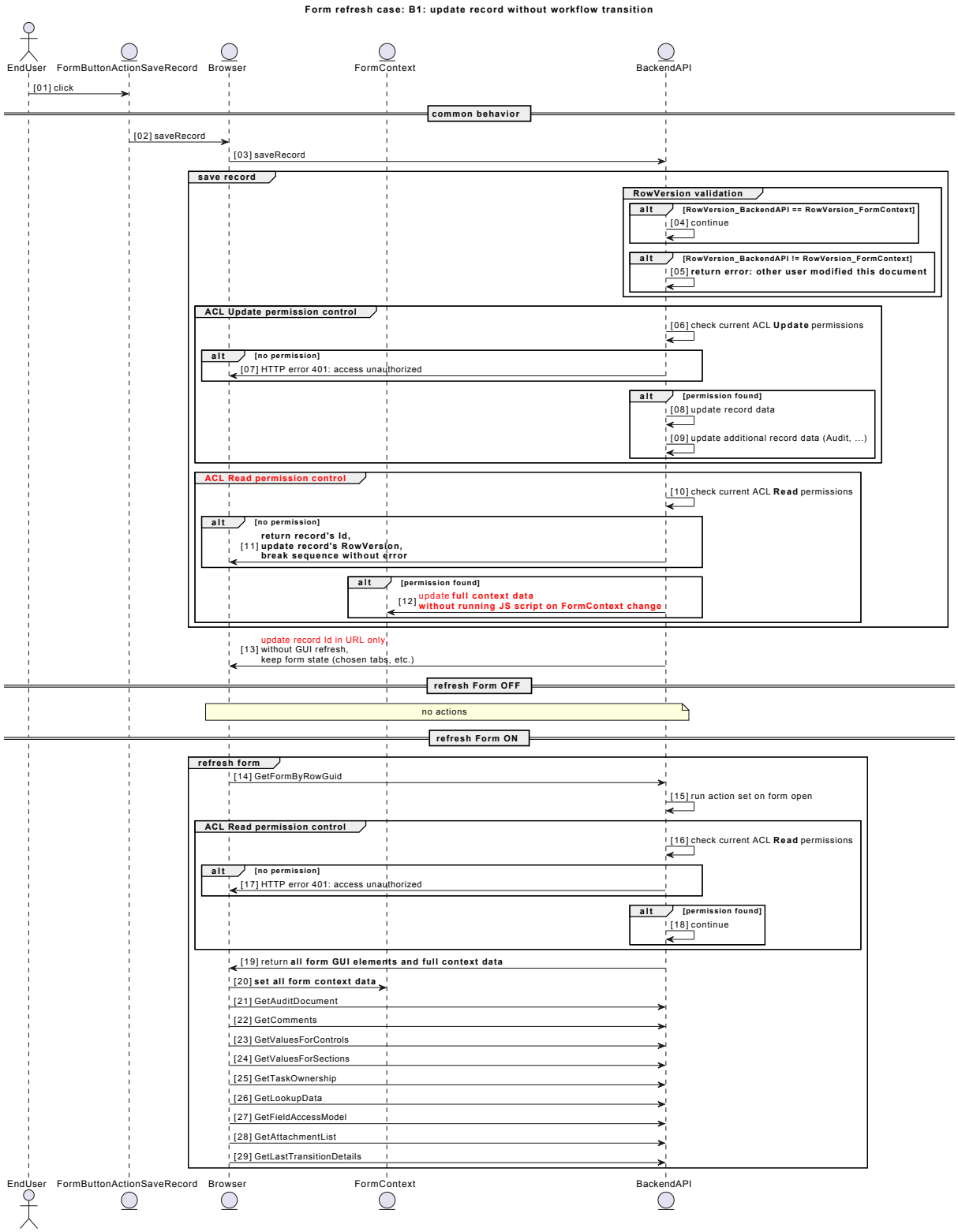
ODŚWIEŻANIE FORMULARZA PODCZAS ZAPISU REKORDU

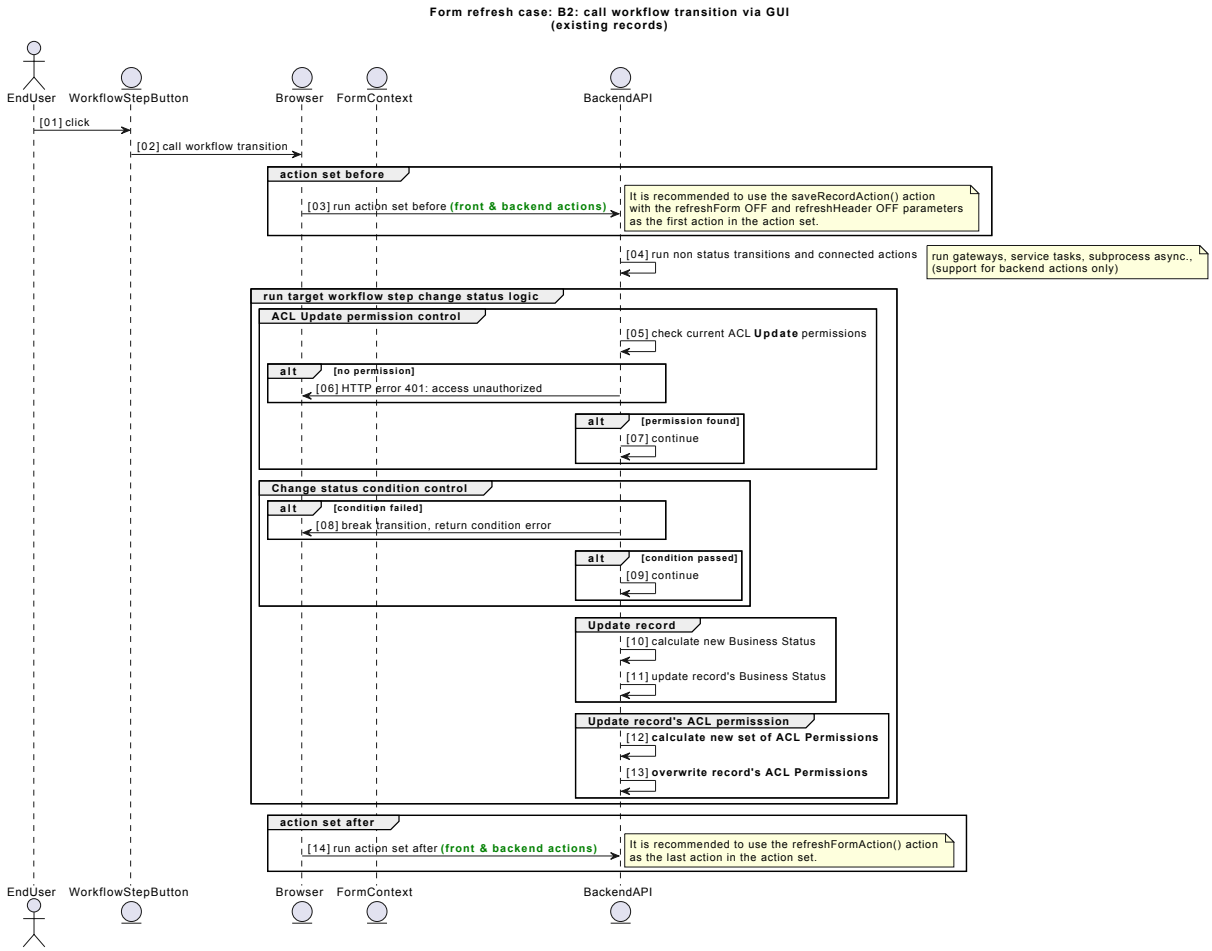
W tej sekcji zamieszczono diagramy sekwencji ilustrujące czynności wykonywane podczas operacji zapisu, zależnie od stanu przełączników dotyczących odświeżania. Diagramy dotyczą czterech przypadków: utworzenie i aktualizacja rekordu w modelu uproszczonym (bez zmiany statusu) oraz utworzenie i aktualizacja rekordu w modelu podstawowym (ze zmianą statusu).

W każdym diagramie na początku wymieniono operacje wykonywane zawsze podczas zapisu rekordu, a następnie dodatkowe czynności w przypadku, gdy odświeżany jest tylko nagłówek (dotyczy tylko zapisu nowego rekordu) oraz gdy odświeżany jest formularz.


Form refresh case: A1. create new record without workflow transition
EndUser
FormButtonActionSaveRecord
Browser
FormContext
BackendAPI
[01] click
[02] saveRecord
[03] saveRecord
common behavior
save record
[04] create instance data
[05] create additional record data (Audit, ...)
[06] create ACL permissions
ACL Read permission control
[07] check current ACL Read permissions
alt
[no permission]
[08] update record's Id + RowVersion, end sequence without error
alt
[permission found]
[09] update full context data without running JS script on FormContext change
update record Id in URL only,
[10] without GUI refresh,
keep form state (chosen tabs, etc.)
refresh Form OFF, refresh Header for new record OFF
no actions
refresh Form OFF, refresh Header for new record ON
refresh form header
[11] GetFormHeader
ACL Read permission control
[12] check current ACL Read permissions
alt
[no permission]
[13] HTTP error 401: access unauthorized
alt
[permission found]
[14] continue
[15] return form header data and system data into form context
[16] set system data in form context
[17] GetTaskOwnership
[18] GetLastTransitionDetails
refresh Form ON
refresh form
[19] GetFormByRowGuid
[20] run action set on form open
ACL Read permission control
[21] check current ACL Read permissions
alt
[no permission]
[22] HTTP error 401: access unauthorized
alt
[permission found]
[23] continue
[24] return all form GUI elements and full context data
[25] set all form context data
[26] GetAuditDocument
[27] GetComments
[28] GetValuesForControls
[29] GetValuesForSections
[30] GetTaskOwnership
[31] GetLookupData
[32] GetFieldAccessModel
[33] GetAttachmentList
[34] GetLastTransitionDetails
EndUser
FormButtonActionSaveRecord
Browser
FormContext
BackendAPI


Form refresh case: A2: create new record with workflow transition via GUI
EndUser
FormButtonActionSaveRecord
Browser
FormContext
BackendAPI
common behavior
[01] click
[02] saveRecord
[03] saveRecord
workflow transition sequence to first status
[04] run non status transitions and connected actions
run gateways, service tasks, subprocess async., (support for backend actions only), skip saveRecord action in this action set
run change status
[05] calculate new Business Status
[06] insert new record
[07] insert additional record data (Audit, ...)
[08] create ACL Permissions
ACL Read permission control
[09] check current ACL Read permissions
alt
[no permission]
[10] update record's Id + RowVersion, break sequence without error
alt
[permission found]
[11] update full context data without running JS script on FormContext change
action set after
[12] run action set after
(support for backend actions only)
update record Id in URL only,
[13] without GUI refresh,
keep form state (chosen tabs, etc.)
refresh Form OFF, refresh Header for new record OFF
no actions
refresh Form OFF, refresh Header for new record ON
refresh form header
[14] GetFormHeader
ACL Read permission control
[15] check current ACL Read permissions
alt
[no permission]
[16] HTTP error 401: access unauthorized
alt
[permission found]
[17] continue
[18] return form header data and system data into form context
[19] set system data in form context
[20] GetTaskOwnership
[21] GetLastTransitionDetails
refresh Form ON
refresh form
[22] GetFormByRowGuid
[23] run action set on form open
ACL Read permission control
[24] check current ACL Read permissions
alt
[no permission]
[25] HTTP error 401: access unauthorized
alt
[permission found]
[26] continue
[27] return all form GUI elements and full context data
[28] set all form context data
[29] GetAuditDocument
[30] GetComments
[31] GetValuesForControls
[32] GetValuesForSections
[33] GetTaskOwnership
[34] GetLookupData
[35] GetFieldAccessModel
[36] GetAttachmentList
[37] GetLastTransitionDetails
EndUser
FormButtonActionSaveRecord
Browser
FormContext
BackendAPI


Form refresh case: B1: update record without workflow transition
EndUser
FormButtonActionSaveRecord
Browser
FormContext
BackendAPI
[01] click
common behavior
[02] saveRecord
[03] saveRecord
save record
RowVersion validation
alt
[RowVersion_BackendAPI == RowVersion_FormContext]
[04] continue
alt
[RowVersion_BackendAPI != RowVersion_FormContext]
[05] return error: other user modified this document
ACL Update permission control
[06] check current ACL Update permissions
alt
[no permission]
[07] HTTP error 401: access unauthorized
alt
[permission found]
[08] update record data
[09] update additional record data (Audit, ...)
ACL Read permission control
[10] check current ACL Read permissions
alt
[no permission]
[11] return record's Id, update record's RowVersion, break sequence without error
alt
[permission found]
[12] update full context data without running JS script on FormContext change
[13] update record Id in URL only, without GUI refresh, keep form state (chosen tabs, etc.)
refresh Form OFF
no actions
refresh Form ON
refresh form
[14] GetFormByRowGuid
[15] run action set on form open
ACL Read permission control
[16] check current ACL Read permissions
alt
[no permission]
[17] HTTP error 401: access unauthorized
alt
[permission found]
[18] continue
[19] return all form GUI elements and full context data
[20] set all form context data
[21] GetAuditDocument
[22] GetComments
[23] GetValuesForControls
[24] GetValuesForSections
[25] GetTaskOwnership
[26] GetLookupData
[27] GetFieldAccessModel
[28] GetAttachmentList
[29] GetLastTransitionDetails
EndUser
FormButtonActionSaveRecord
Browser
FormContext
BackendAPI



6.1.46.4. USUŃ REKORD

> **Podsumowanie:**
> Akcja usuwa bieżący dokument. Wraz z dokumentem usuwane są także wpisy ACL wygenerowane dla tego dokumentu. Akcja nie usuwa komentarzy ani załączników dodanych do dokumentu.

Po wykonaniu akcji widok formularza ani kontekst nie zostaje odświeżony — nadal są widoczne dane z usuniętego dokumentu.

Akcję można przypisać tylko do przycisku na formularzu i do przejścia na diagramie procesu.

Akcja nie ma żadnych parametrów.

Akcja usuwa wpisy ACL z tabel *core.ACLInstanceEntries*, *core.ACLInstanceEntryRole* oraz *core.ACLStakeholderEntries*.

6.1.46.5. ODŚWIEŻ MODEL

> **Podsumowanie:**
> Akcja odświeża model danych (kontekst) bieżącego formularza.

Akcja wysyła jedno żądanie *GetFormModel* i odpowiednio aktualizuje wartości wyświetlane w formularzu.

Akcję można przypisać tylko do przycisku na formularzu i przejścia na diagramie procesu.

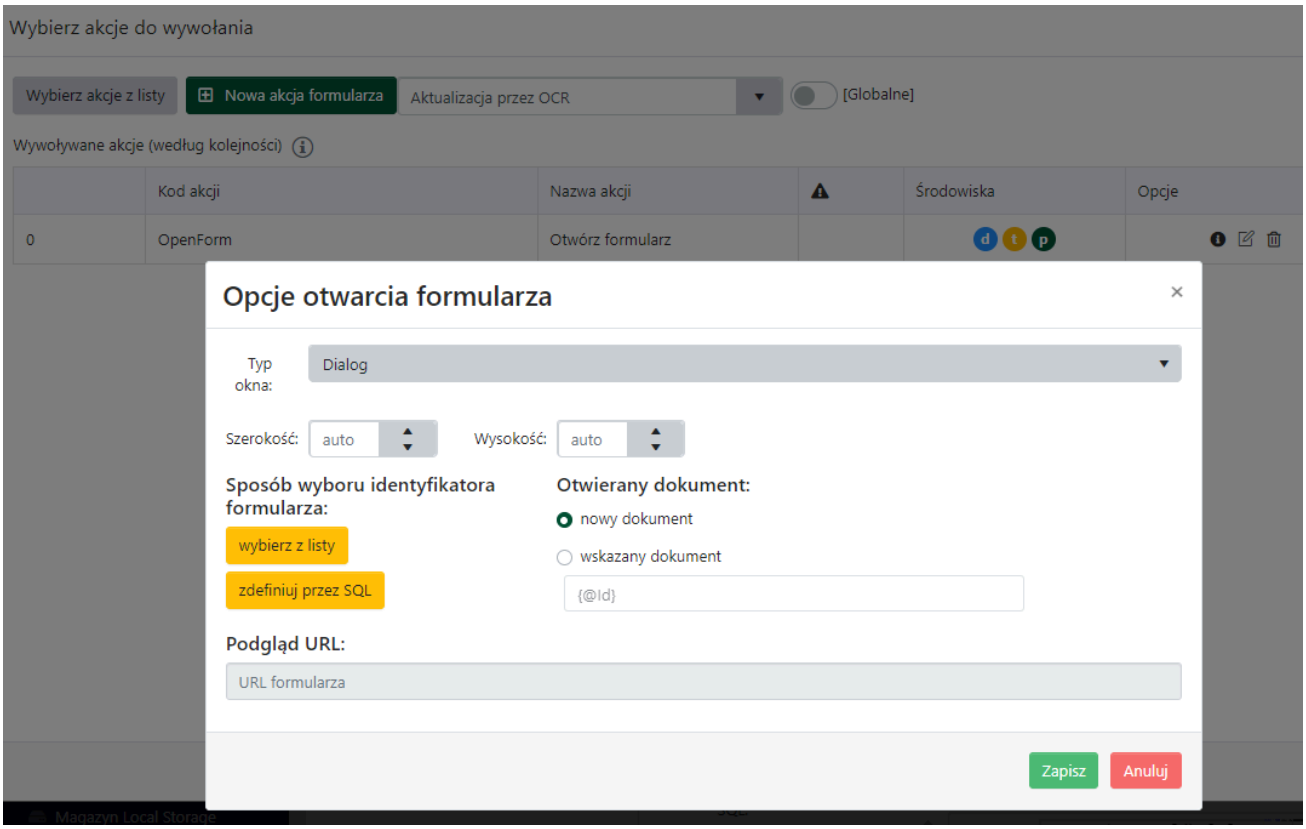
Akcja nie ma żadnych parametrów.

6.1.46.6. OTWÓRZ FORMULARZ

Podsumowanie:
Akcja otwiera wskazany formularz.

Akcja działa tak samo jak globalna akcja otwierania formularza: można skonfigurować tryb wyświetlania okna formularza, wskazać formularz i rekord do wyświetlenia oraz przekazać wartości do pól otwieranego formularza.

Akcję można przypisać do przycisku na formularzu i widoku listy oraz do przejścia na diagramie procesu.



Opcje otwierania formularza

Powiązane tematy:

- Akcja otwierania formularza (globalna)

6.1.46.7. ODŚWIEŻ FORMULARZ

> **Podsumowanie:**
> Akcja odświeża bieżący formularz.

Działanie jest analogiczne jak kliknięcie łącza otwierającego formularz, kliknięcie przycisku odświeżania formularza w prawym górnym rogu lub wykonanie akcji odśwież stronę.

Akcja odświeża dane wyświetlane w formularzu zgodnie ze stanem rekordu w bazie danych. W przypadku nowego rekordu, akcja powoduje wyczyszczenie niezapisanych danych i wyświetla pola puste lub z wartościami domyślnymi.

Akcję można przypisać do przycisku na formularzu i w widoku listy oraz do przejścia na diagramie procesu.

Akcja nie ma żadnych parametrów.

6.1.46.8. WYLOGUJ

> **Podsumowanie:**
> Akcja wylogowuje bieżącego użytkownika.

Po wykonaniu akcji wyświetlane jest okienko z komunikatem o pomyślnym wylogowaniu i linkiem do strony logowania. Działanie akcji jest analogiczne do kliknięcia ikony wylogowania w prawym górnym rogu okna aplikacji *FrontSPA*.

Akcję można przypisać tylko do przycisku na formularzu i do przejścia na diagramie procesu.

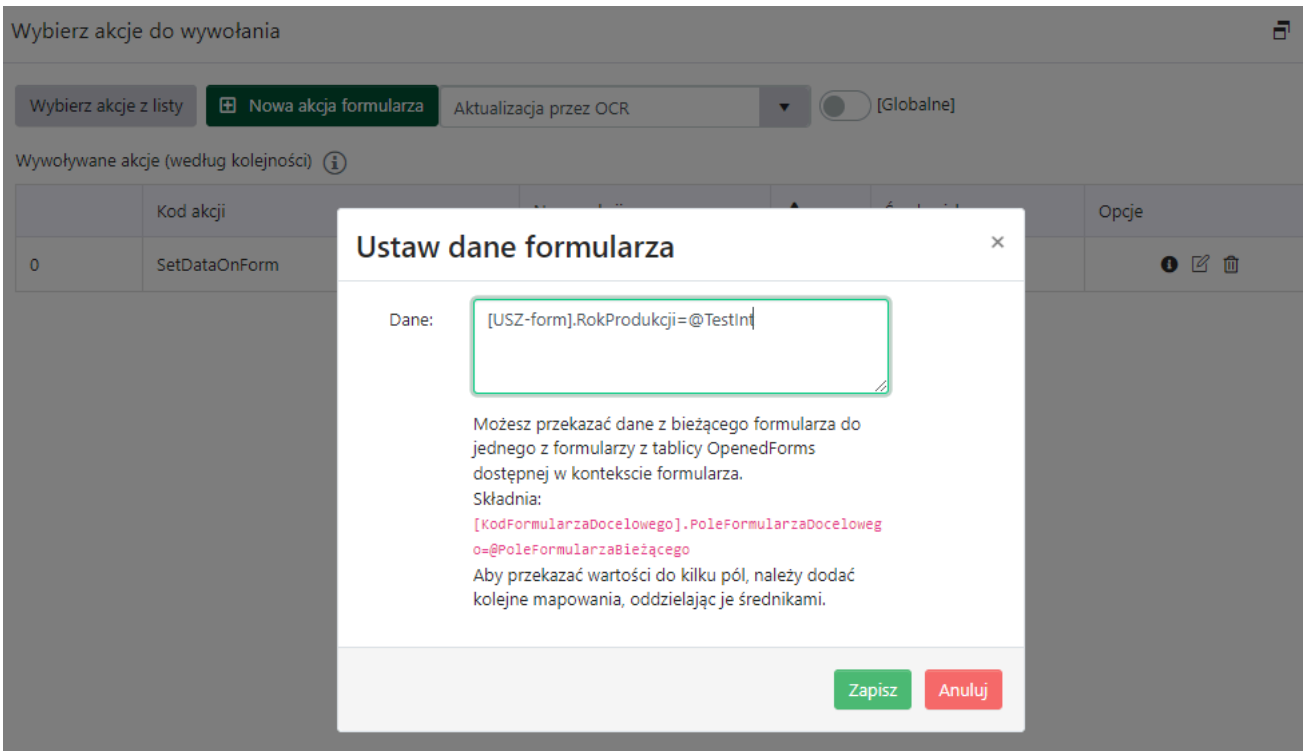
Akcja nie ma żadnych parametrów.

6.1.46.9. USTAW DANE FORMULARZA

> **Podsumowanie:**
> Akcja przekazuje dane z bieżącego formularza do jednego z otwartych formularzy. Akcję można utworzyć tylko w miejscu wywołania (akcja lokalna).

Wywołanie akcji powoduje ustawienie wartości pól kontekstu w innym (otwartym) formularzu. W oknie dialogowym konfiguracji akcji należy wskazać formularz docelowy i pole w tym formularzu oraz pole w bieżącym formularzu, z którego wartość ma zostać przekazana. Należy użyć następującej składni:

```
[KodFormularzaDocelowego].PoleFormularzaDocelowego=@PoleFormularzaBieżącego
```



Ustaw dane formularza

Kod i pole formularza docelowego powinny być wyświetlane w tablicy *OpenedForms* w kontekście bieżącego formularza.

Kontekst formularza nadrzędnego

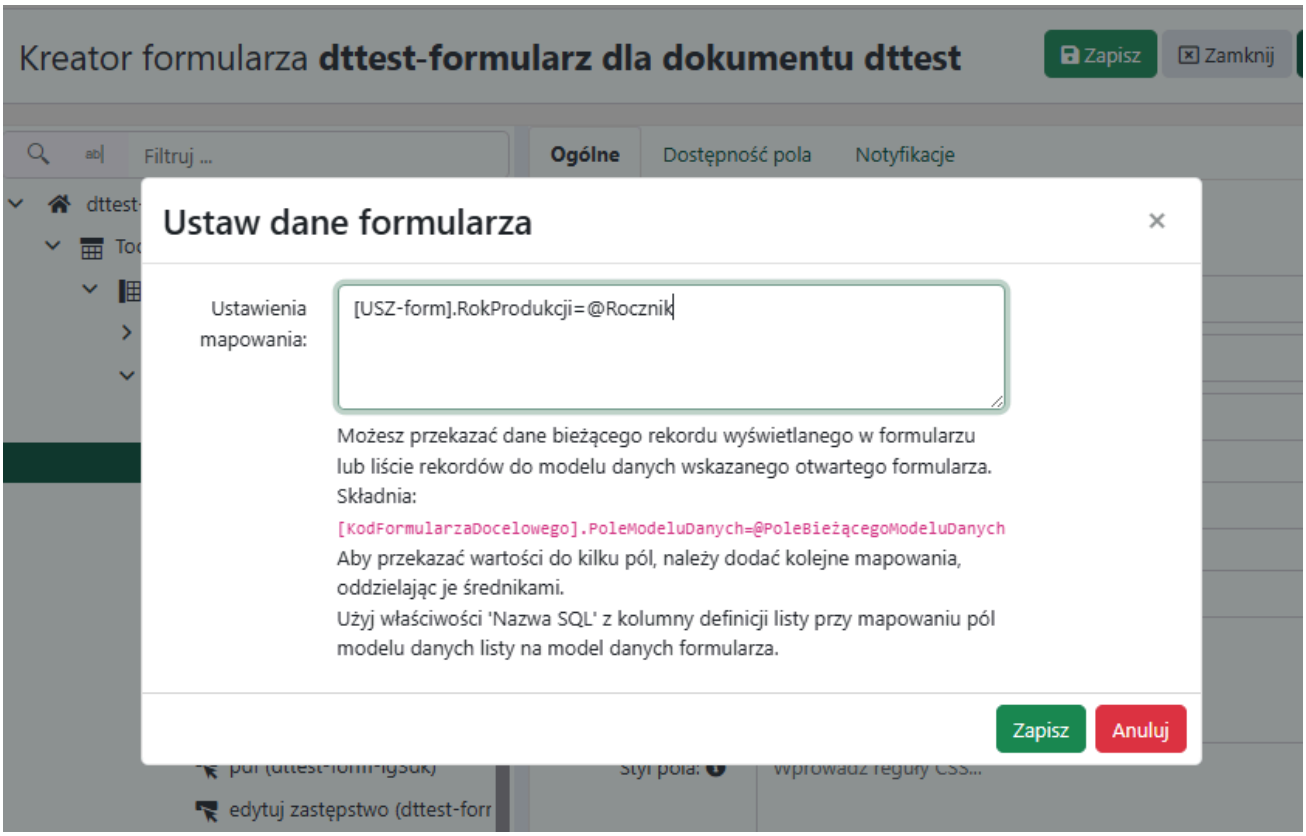
Akcję można przypisać do przycisku na formularzu i do przejścia na diagramie procesu.

Akcję można przypisać do przycisku na formularzu, do przejścia na diagramie procesu oraz do przycisku w wierszu listy osadzonej na formularzu.

Wywołanie akcji powoduje ustawienie wartości pól kontekstu w innym (otwartym) formularzu. W oknie dialogowym konfiguracji akcji należy wskazać formularz docelowy i

pole w tym formularzu oraz pole w modelu danych bieżącego obiektu (formularz lub rekord listy), z którego wartość ma zostać przekazana. Należy użyć następującej składni:

```
[KodFormularzaDocelowego].PoleModeluDanych=@PoleBieżącegoModeluDanych
```



Ustaw dane formularza

W przypadku, gdy akcja jest uruchamiana przyciskiem z formularza, w kontekście tego formularza, w tablicy *OpenedForms* powinny być wyświetlane kod i pole formularza docelowego.

Kontekst formularza nadrzędnego

Akcja może także służyć do ustawiania wartości w kontekście formularza nadrzędnego z poziomu rekordu listy osadzonej w tym formularzu w sekcji typu lista.

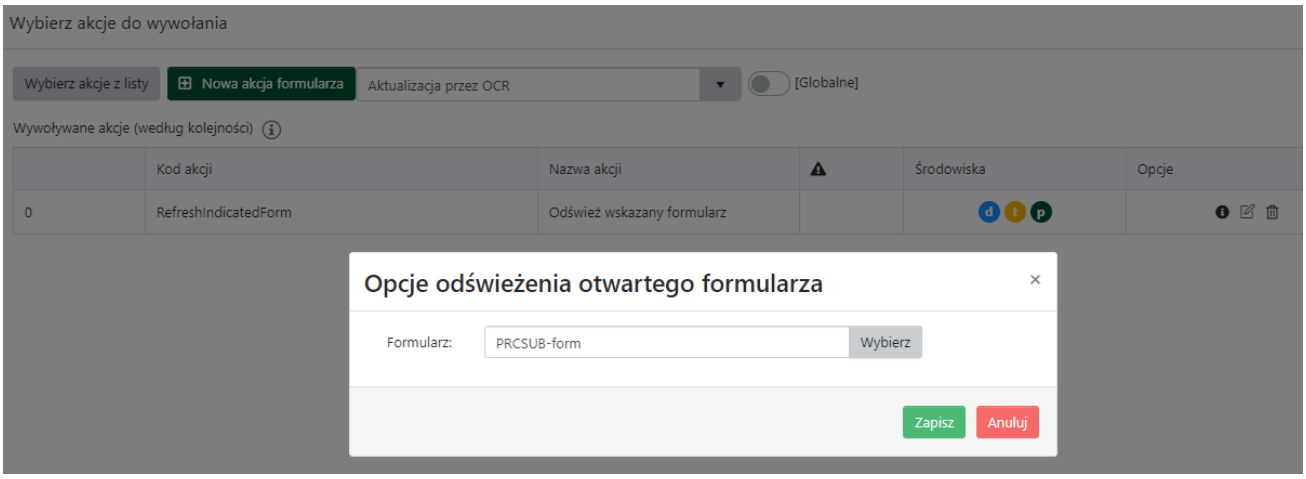
6.1.46.10. ODŚWIEŻ WSKAZANY FORMULARZ

> **Podsumowanie:**
> Akcja odświeża otwarty formularz o podanym kodzie.

Akcja wysyła żądanie ponownego załadowania z bazy danych definicji tego formularza oraz danych z bieżącego rekordu.

Akcję można przypisać do przycisku na formularzu i w widoku listy oraz do przejścia na diagramie procesu.

W celu skonfigurowania akcji należy podać kod formularza, który ma zostać odświeżony. Przycisk `Wybierz` w oknie `Opcje odświeżania otwartego formularza` wyświetla listę zdefiniowanych formularzy.



Opcje odświeżania otwartego formularza

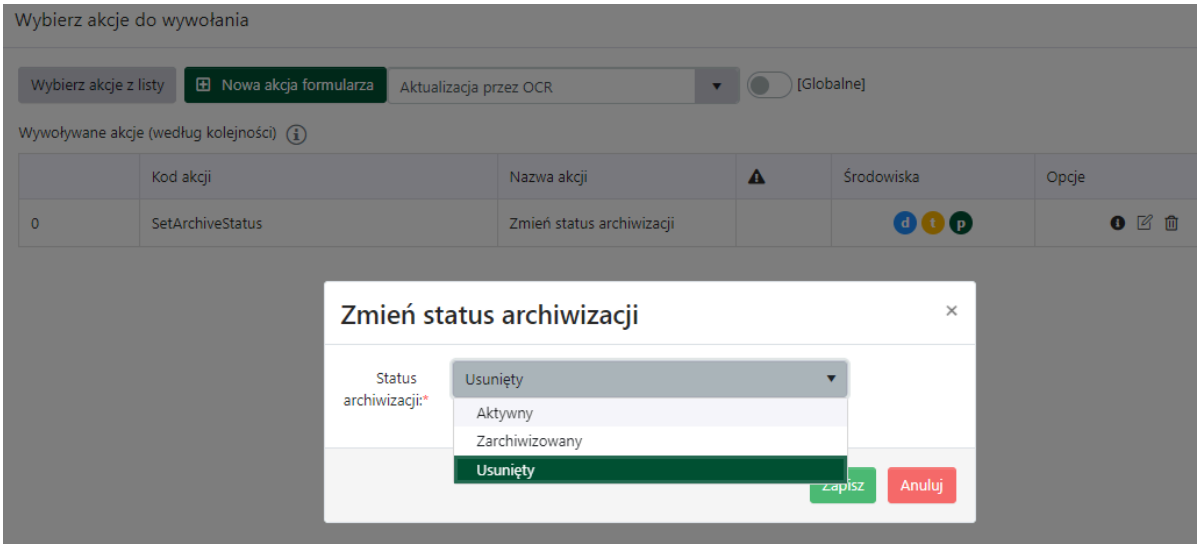
6.1.46.11. ZMIEŃ STATUS ARCHIWIZACJI

> **Podsumowanie:**
> Akcja zmienia status archiwizacji bieżącego dokumentu, to jest wartość w kolumnie systemowej *ArchiveStatus* tabeli modelu danych.

Wywołanie akcji powoduje ustawienie statusu archiwizacji zgodnie z konfiguracją akcji. W oknie dialogowym należy wybrać status archiwizacji. DOstępne wartości:

- `Aktywny`: domyślny status archiwizacji dla aktywnych dokumentów; wartość *1* w kolumnie *ArchiveStatus*.
- `Zarchiwizowany`: dokument w tym statusie jest wyświetlany w formularzu w trybie tylko do odczytu; jeśli przewidywana jest zmiana statusu archiwizacji, odpowiedni przycisk powinien mieć włączony przełącznik `Przycisk zawsze dostępny`; wartość *2* w kolumnie *ArchiveStatus*.

- `Usunięty`: działa taka samo jak status `Zarchiwizowany`, ale ma wartość *3* w kolumnie *ArchiveStatus*



Zmień status archiwizacji

Akcję można przypisać do przycisku na formularzu i do przejścia na diagramie procesu. Podczas wykonywania akcji sprawdzane są uprawnienia ACL użytkownika, który zmienił akcję. Do zmiany statusu na archiwalny wymagane są uprawnienia do odczytu i aktualizacji (RU), a do ustawienia statusu usuniętego uprawnienia do odczytu i usuwania (RD). W przypadku zmiany statusu archiwizacji dla dokumentu w statusie archiwalnym lub usuniętym wymagane są uprawnienia do odczytu i administracji (RA).

6.1.46.12. BLOKUJ DOKUMENT

> **Podsumowanie:**
> Akcja blokuje bieżący dokument, to jest ustawia wartość *1* (*true*) w kolumnie *Blocked* tabeli modelu danych.

Akcja ustawia flagę blokady bieżącego rekordu, co pozwala zapobiec próbom jednoczesnej zmiany rekordu przez kilku użytkowników. Opcje blokady rekordów można ustawić we właściwościach definicji dokumentu.

Akcja jest przeznaczona do użycia w zdarzeniu formularza (`Kreator formularza...` > `Akcje formularza` > `Akcje na otwarcie`).

Akcja nie ma żadnych parametrów.

> **Powiązane tematy:**
> - Definicja dokumentu, karta Blokady
> - Właściwości formularza, Akcje na otwarcie

6.1.46.13. ODBLOKUJ DOKUMENT

Podsumowanie:
Akcja odblokowuje bieżący dokument, to jest ustawia wartość *0* (*false*) w kolumnie *Blocked* tabeli modelu danych.

Akcja wyłącza flagę blokady bieżącego rekordu, co pozwala edytować go przez innych użytkowników. Opcje blokady rekordów ustawia się we właściwościach definicji dokumentu.

Akcja jest przeznaczona do użycia w zdarzeniu formularza (`Kreator formularza...` > `Akcje formularza` > `Akcje na zamknięcie`).

Akcja nie ma żadnych parametrów.

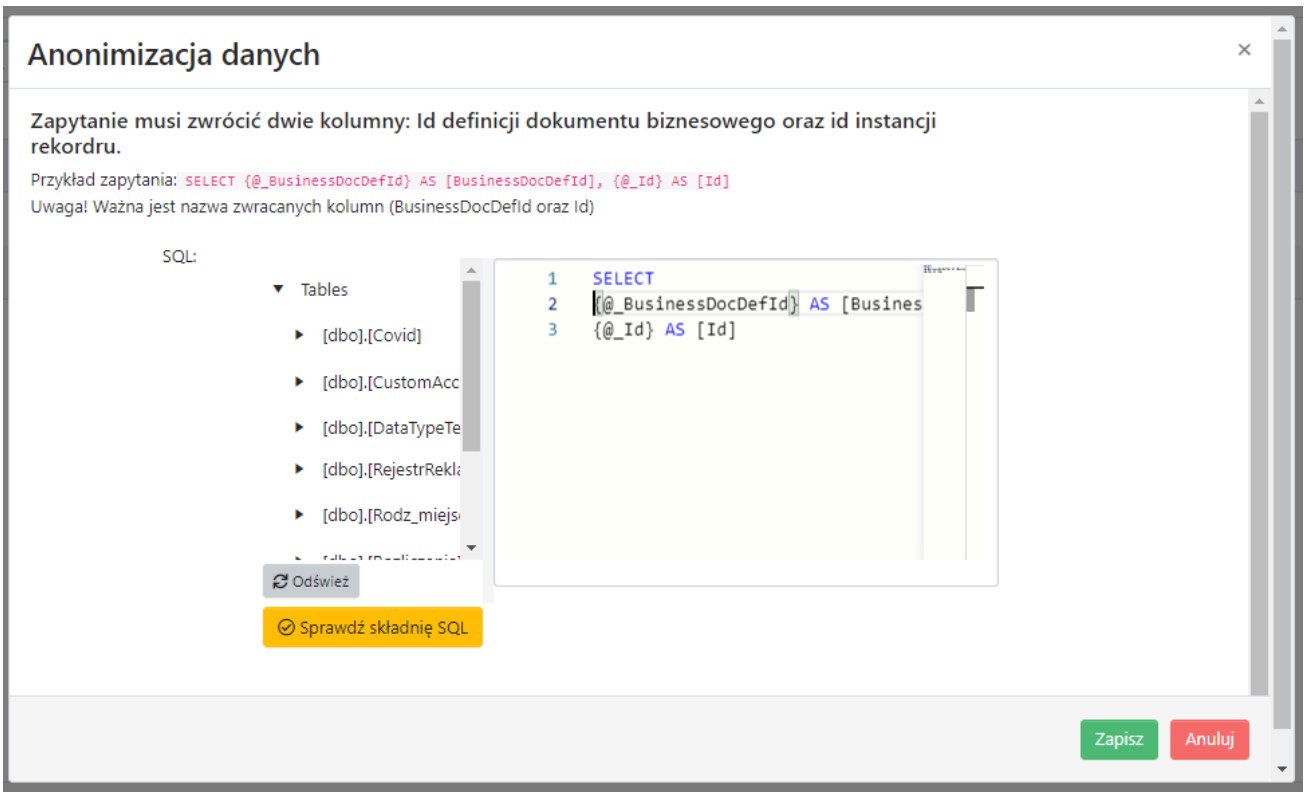
Powiązane tematy:

- Definicja dokumentu, karta Blokady
- Właściwości formularza, Akcje na otwarcie

6.1.46.14. ANONIMIZUJ DANE

Podsumowanie:
Akcja anonimizacji danych służy do zanonimizowania danych osobowych w dokumencie biznesowym wyświetlanym w bieżącym formularzu. Akcję przypisuje się do przycisku w nagłówku sekcji typu *Sekcja danych RODO*.

Akcja wymaga skonfigurowania zapytania SQL, które zwróci identyfikator definicji dokumentu biznesowego (pod nazwą *BusinessDocDefId*) oraz identyfikator samego dokumentu biznesowego (pod nazwą *Id*).



Akcja anonimizacji danych

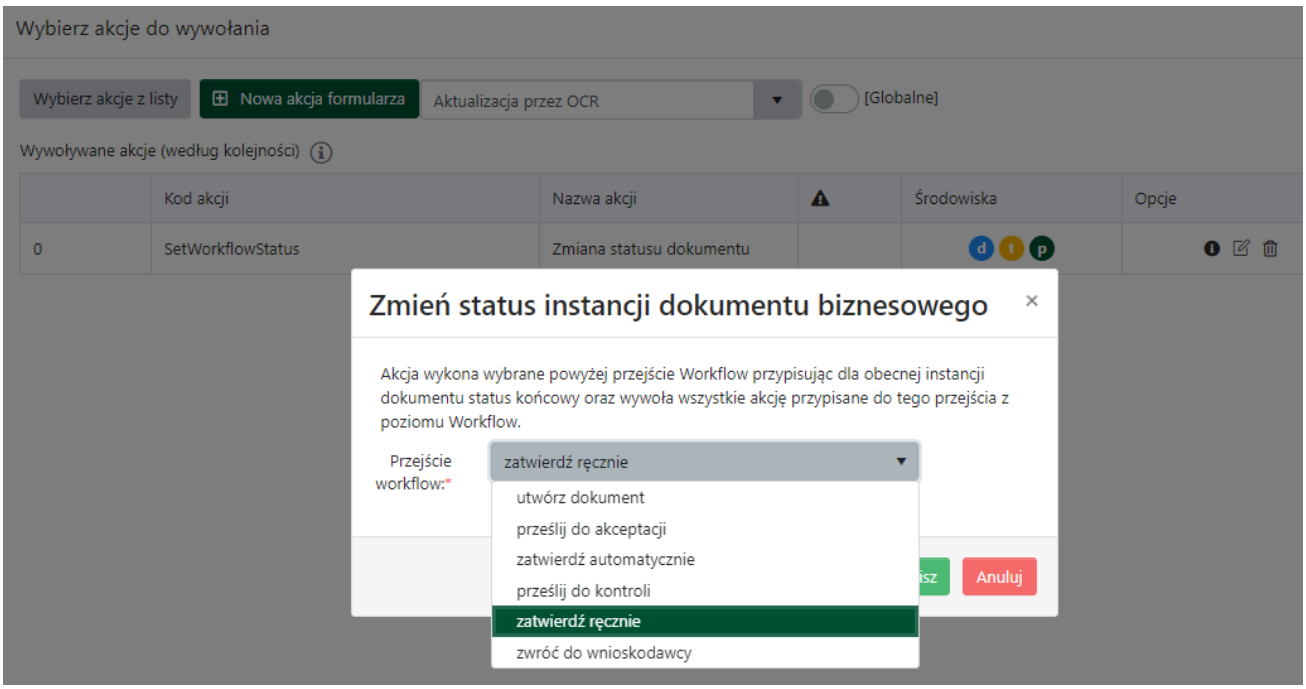
Powiązane tematy:

- Sekcja danych RODO

6.1.46.15. ZMIANA STATUSU DOKUMENTU

Podsumowanie:
Akcja zmienia status bieżącego dokumentu, wykonując wskazane przejście diagramu procesu wraz z akcjami przypisanymi do tego przejścia.

Działa tak samo, jak kliknięcie przycisku zmiany statusu na formularzu. Wywołanie akcji powoduje ustawienie statusu bieżącego dokumentu zgodnie z konfiguracją akcji. W oknie dialogowym należy wybrać jedno z przejść zdefiniowanych na diagramie procesu dla bieżącego dokumentu.



Zmiana statusu dokumentu

Akcję można przypisać do przycisku na formularzu i do przejścia na diagramie procesu.

Powiązane tematy:

- Akcja zmiany statusu dokumentów (globalna)

6.1.46.16. ODŚWIEŻ LISTĘ

Podsumowanie:
Akcja odświeża bieżący widok listy.

Działanie jest analogiczne jak kliknięcie łącza otwierającego widok listy lub kliknięcie przycisku odświeżania listy w prawym górnym rogu i polega na wysłaniu żądania *GetGridPartialContent* z odpowiednimi parametrami.

Akcja odświeża dane wyświetlane w widoku listy zgodnie z ostatnimi ustawieniami grupowania, filtrowania i sortowania.

Akcję można przypisać tylko do przycisku w widoku listy. Przypisanie akcji do przejścia na diagramie procesu spowoduje wystąpienie błędu w czasie wykonywania przejścia.

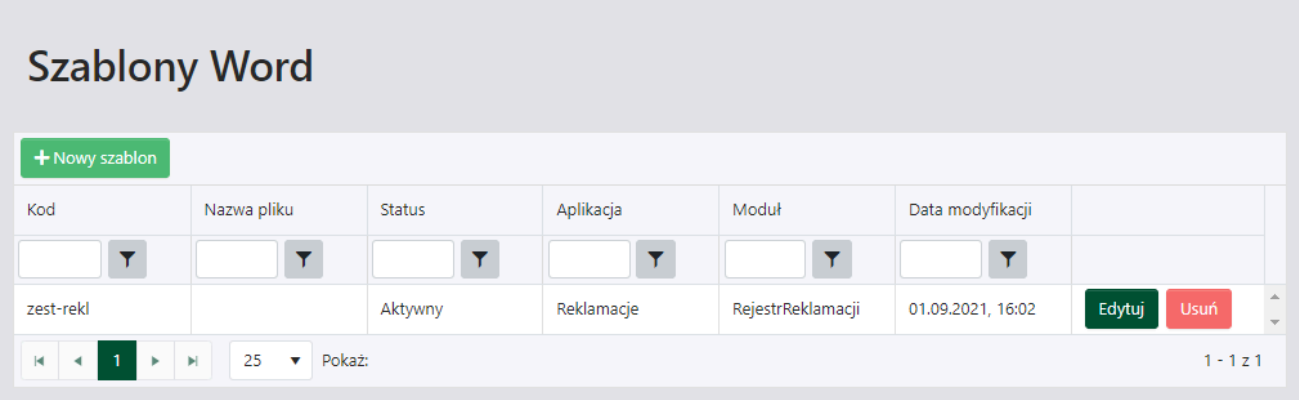
Akcja nie ma żadnych parametrów.

6.2. SZABLONY WORD

Podsumowanie:
Funkcja szablonów word pozwala definiować szablony dokumentów w formacie programu MS Word, które są wypełniane danymi z nAxiom i mogą być konwertowane do formatu PDF lub zapisywane w formacie programu Word np. w celu wydruku. W szablonach word obsługiwane są pola typu *text*, *date*, *barcode* oraz tabele (*table*), ponadto pliki szablonów mogą zawierać bloki warunkowe *if/else/endif*.

☛ *AdminSPA*: `NARZĘDZIA` > `Szablony` > `Szablony Word`

Po wybraniu polecenia `Szablony word` zostanie wyświetlona strona `Szablony word`.



Lista szablonów word

Przycisk `Nowy szablon` otwiera okno dialogowe `Tworzenie definicji szablonu word`, w którym należy określić ustawienia szablonu word na kartach `Ogolne` i `Źródło danych` (karta dostępna tylko wtedy, gdy w modelu danych dla szablonu zadeklarowano tabelę) zgodnie z opisem w poniższej sekcji. Przyciski `Edycja` i `Usuń` służą odpowiednio do edytowania ustawień szablonu oraz do usunięcia definicji szablonu z listy.

UWAGA
Korzystanie z szablonu word wymaga zdefiniowania akcji generowania pliku PDF; patrz Generowanie PDF z Word .

OKNO `TWORZENIE DEFINICJI SZABLONU WORD` — KARTA `OGÓLNE`

Tworzenie definicji szablonu Word

Ogólne | Źródło danych

Kod: zest-rekl

Nazwa: Zestawienie reklamacji

Status: Aktywny

Aplikacja: * Reklamacje

Moduł: * RejestrReklamacji

Opis: Generuje zestawienie reklamacji

Model danych JSON:

```
1  [
2    {
3      "value": "Imie",
4      "testValue": "Jan",
5      "type": "text"
6    },
7    {
8      "value": "Nazwisko",
9      "testValue": "Kowalski",
10     "type": "text"
11   },
12   {
13     "value": "IdUzytkownika",
14     "type": "barcode",
15     "width": 1000,
```

Sprawdź poprawność modelu

Edytor SQL Query: Aby skorzystać z danych formularza użyj składni: **{@NazwaPola}**

```
1  SELECT
2  MIN([core].[UserProfiles].[FirstName]) AS [Imie]
3  ,MIN([core].[UserProfiles].[LastName]) AS [Nazwisko]
4  ,MIN([core].[UserProfiles].[Id]) AS [IdUzytkownika]
5  ,MIN([dbo].[RejestrReklamacji].[CreateDate]) AS [Dat
6  ,MAX([dbo].[RejestrReklamacji].[CreateDate]) AS [Dat
7  FROM
8  [core].[UserProfiles]
9  RIGHT JOIN
10 [dbo].[RejestrReklamacji]
11 ON [core].[UserProfiles].[Id] = {@_UserId}
```

Prześlij szablon: Szablon nie został odnaleziony w bazie danych!

Zapisz | Anuluj | Pobierz przesłany szablon | Aktualizuj model JSON | Generuj przykładowy dokument

Definicja szablonu word, karta Ogólne

W wyświetlonym oknie dialogowym należy określić szereg ustawień szablonu:

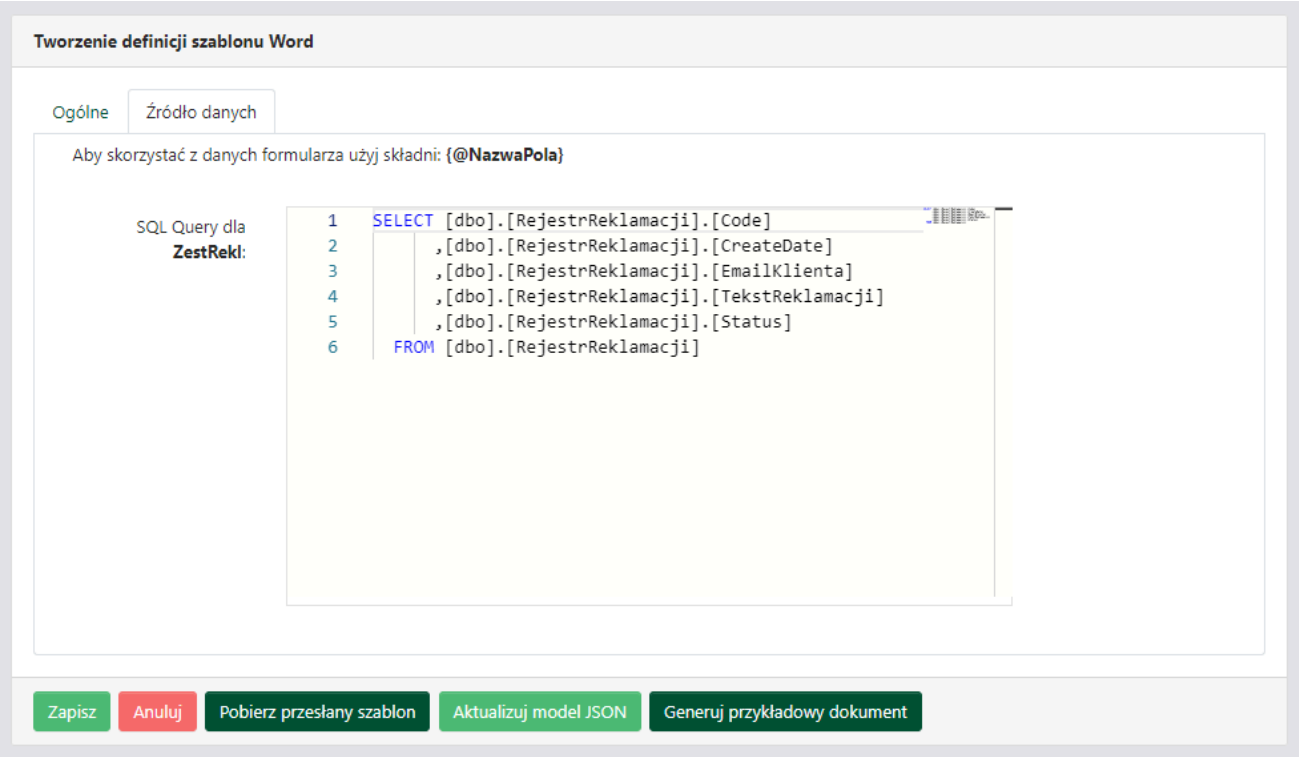
- `Kod`: kod szablonu word.
- `Nazwa`: nazwa szablonu word.
- `Status`: status szablonu word; dostępne wartości `Aktywny` (wartość domyślna), `Zarchiwizowany` i `Usunięty`. `Aplikacja`: nazwa aplikacji, do której zostanie przypisany tworzony obiekt; wszystkie obiekty przypisane do aplikacji można wraz z nią wyeksportować, używając polecenia `Migrator aplikacji` w sekcji `ADMINISTRACJA` menu głównego. `Moduł`: nazwa modułu aplikacji, do którego zostanie przypisany tworzony obiekt.
- `Opis`: opis szablonu word.
- `Model danych JSON`: deklaracja pól i ich typów używanych w szablonie w formacie JSON; kliknięcie ikony informacji obok etykiety wyświetla przykładowe deklaracje pól różnych typów, tabeli oraz całego modelu; klucz *testValue* pozwala zadeklarować przykładową wartość danego pola w celu sprawdzenia podglądu generowanego pliku za pomocą przycisku `Generuj przykładowy dokument`.

> **UWAGA**
> Model danych JSON zadeklarowany w tym polu musi być tablicą.

- `Edytor SQL Query`: standardowe okno edytora SQL, w którym należy wpisać zapytanie zwracające wartości dla pól w szablonie; w zapytaniu należy używać aliasów dla nazw kolumn zgodnych z podanymi w modelu JSON.
- `Prześlij szablon`: pole, na które można upuścić plik szablonu lub które można kliknąć w celu wyświetlenia pola wyboru pliku szablonu; przycisk `Pobierz przesłany szablon` umożliwia pobranie pliku szablonu, np. w celu modyfikacji — w takim przypadku należy ponownie upuścić zmodyfikowany plik na pole `Prześlij szablon`.

OKNO `TWORZENIE DEFINICJI SZABLONU WORD` — KARTA `ŹRÓDŁO DANYCH`

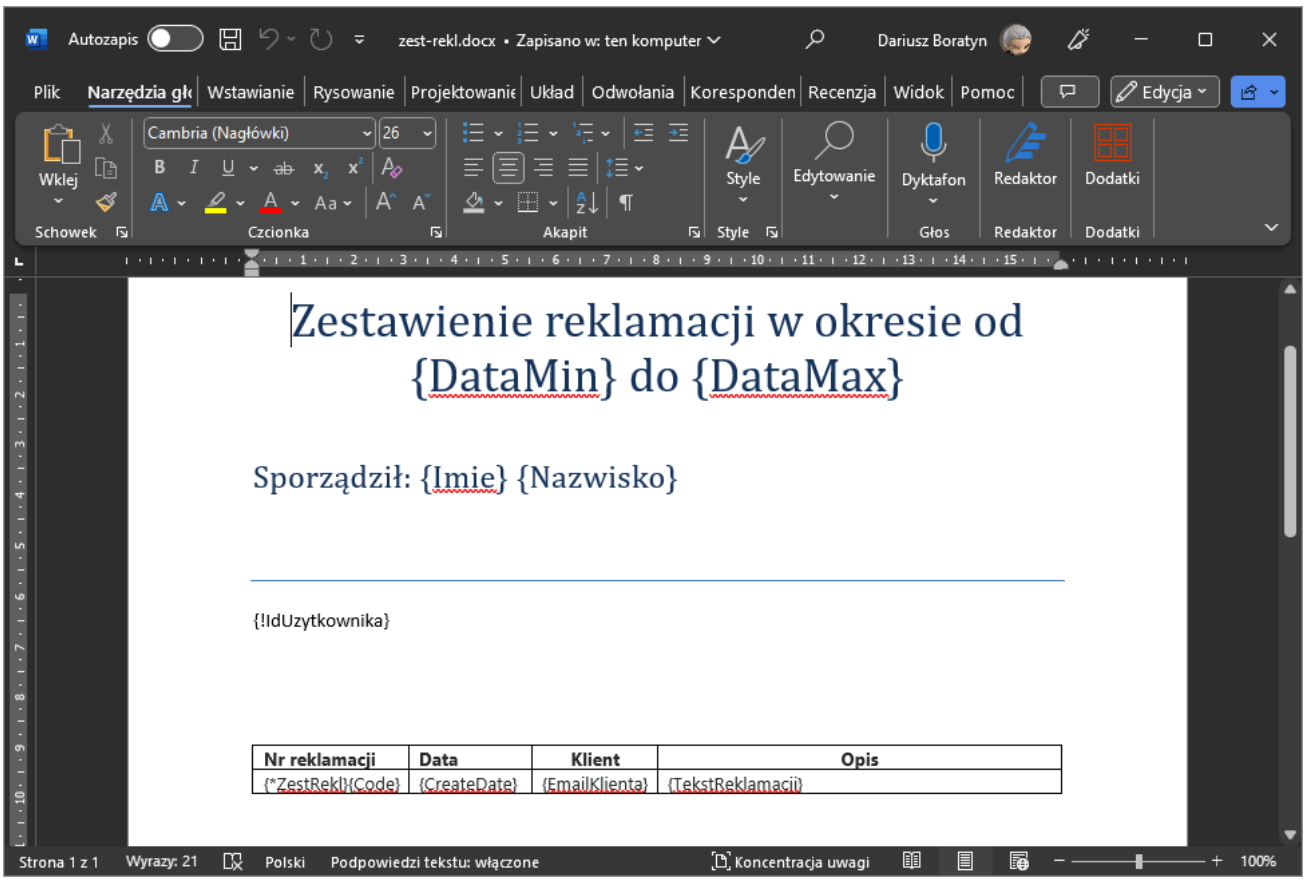
Jeśli w modelu danych JSON szablonu word zostanie zadeklarowana tabela, kliknij przycisk `Aktualizuj model JSON`, aby wyświetlić kartę `Źródło danych`, na której należy zadeklarować zapytanie SQL, które zwróci dane do tabeli. Na tej karcie jest dostępne standardowe okno edytora SQL z etykietą `SQL Query dla *NazwaTabeli*`, gdzie *NazwaTabeli* to nazwa tabeli zadeklarowana w modelu danych JSON. W zapytaniu można używać odwołań do danych z formularza, używając składni *{@PoleFormularza}*.



Definicja szablonu word, karta Źródło danych

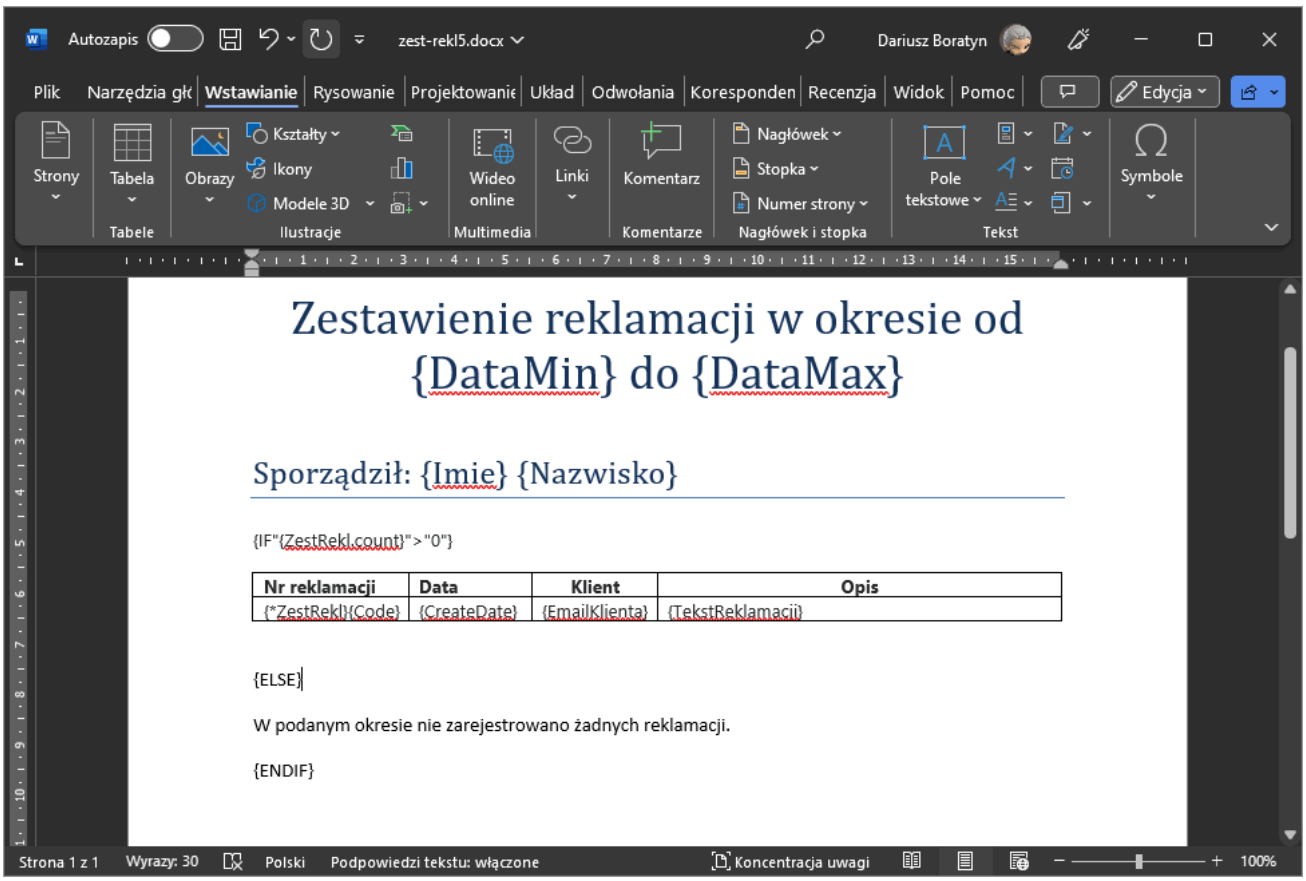
PLIK SZABLONU

Podstawowym elementem definicji szablonu word jest plik szablonu w formacie programu MS Word. Jest to makieta wynikowego dokumentu, która zostanie wypełniona danymi i zapisana w formacie *doc* lub przekonwertowana do formatu PDF. Szablon, oprócz stałego tekstu, może zawierać pola, które będą wypełniane danymi z zapytań SQL podanych w definicji szablonu. Pola w szablonie mają format *{Nazwa}*, gdzie *Nazwa* to wartość klucza *Value* z modelu danych JSON. W przypadku tabeli należy w pliku szablonu wstawić tabelę i wpisać w pierwszej kolumnie *{*Tabela}{Kolumna1}*, gdzie *Tabela* to nazwa tabeli zdefiniowana w modelu danych, a *Kolumna1* to nazwa pierwszej kolumny. W kolejnych kolumnach tabeli wpisujemy nazwy kolejnych kolumn z modelu w nawiasach klamrowych, np. *{Kolumna2}*, *{Kolumna3}* itd. Aby w generowanym dokumencie wyświetlić dowolną wartość przekonwertowaną na kod kreskowy, należy użyć składni *{!Nazwa}*, gdzie *Nazwa* to nazwa pola zadeklarowana w modelu danych (w modelu danych należy dla takiego pola zadeklarować także typ *barcode*). Poniższa ilustracja przedstawia przykładowy szablon w programie MS Word.



Szablon w programie MS Word

W pliku szablonu można także umieszczać sekcje warunkowe, na przykład przedstawione na ilustracji poniżej.



Blok warunkowy if/else/endif w pliku szablonu

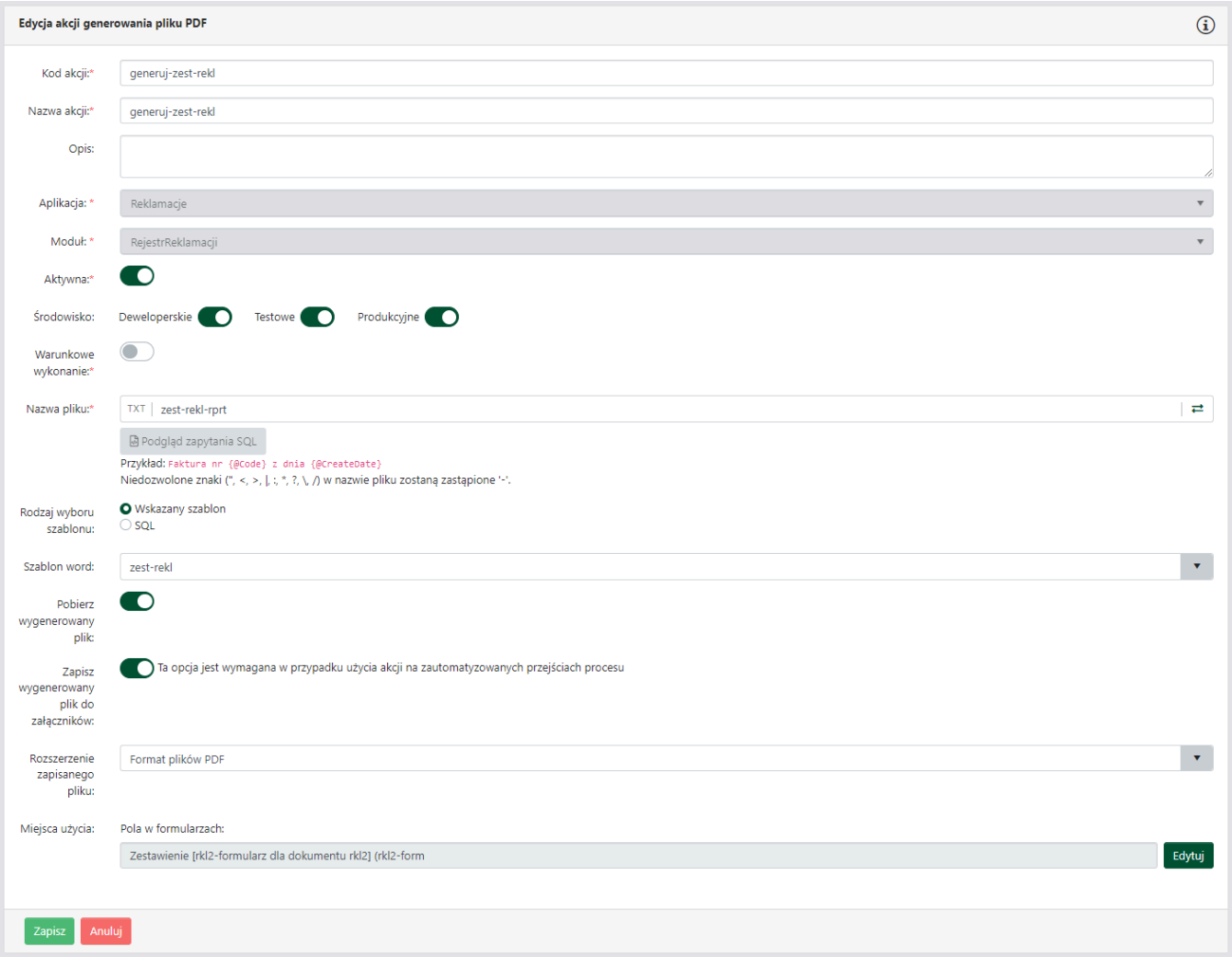
W warunkach można używać operatorów == (równe), <= (mniejsze lub równe), >= (większe lub równe), <> (różne), < (mniejsze od), > (większe od). Operandy należy ująć w cudzysłowy proste (", Alt + 0034).

> **WAŻNE**
> W programie MS Word należy wyłączyć opcję autoformatowania, która automatycznie zamienia cudzysłów prosty na „drukarski". Szablon z cudzysłowami drukarskimi nie będzie działał.

Aby w warunku odwołać się do wartości kolumny w tabeli, należy użyć składni *{Tabela[n].Kolumna}*, gdzie *Tabela* to nazwa tabeli, *n* to numer wiersza (pierwszy wiersz ma numer *0*), a *Kolumna* to nazwa kolumny tabeli. Ponadto bloki warunkowe można zagnieżdżać.

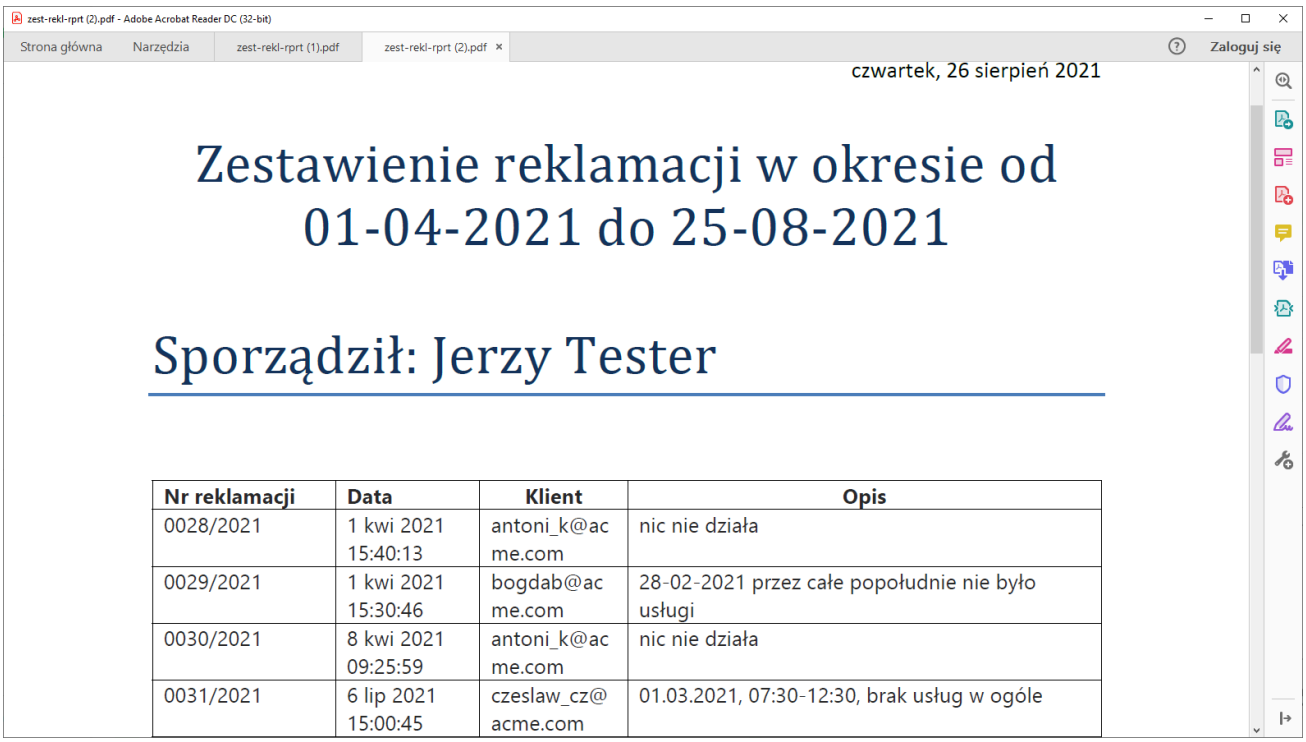
UŻYCIE SZABLONU W AKCJI GENEROWANIA PLIKU PDF

Aby użyć definicji szablonu wraz z plikiem szablonu do wygenerowania pliku w formacie PDF lub doc, należy skonfigurować akcję generowania pliku PDF, w której ten szablon zostanie użyty, na przykład taką, jak na ilustracji poniżej.



Przykładowa akcja generowania pliku PDF

Wykonanie akcji, na przykład przez kliknięcie na formularzu przycisku, do którego została przypisana, spowoduje wygenerowanie pliku i pobranie go przez przeglądarkę. Wygenerowany plik przedstawiono na ilustracji poniżej.



czwartek, 26 sierpień 2021

Zestawienie reklamacji w okresie od 01-04-2021 do 25-08-2021

Sporządził: Jerzy Tester

Nr reklamacji	Data	Klient	Opis
0028/2021	1 kwi 2021 15:40:13	antoni_k@acme.com	nic nie działa
0029/2021	1 kwi 2021 15:30:46	bogdab@acme.com	28-02-2021 przez całe popołudnie nie było usługi
0030/2021	8 kwi 2021 09:25:59	antoni_k@acme.com	nic nie działa
0031/2021	6 lip 2021 15:00:45	czeslaw_cz@acme.com	01.03.2021, 07:30-12:30, brak usług w ogóle

Plik w formacie PDF wygenerowany na podstawie szablonu

6.3. RAPORTY

Podsumowanie:
Raporty służą do wizualnej prezentacji danych przetwarzanych w aplikacjach nAxiom. Aby korzystać z raportów, należy zaprojektować szablon raportu, a następnie użyć akcji raportów Telerik do wygenerowania raportu z zestawem danych określonym za pomocą parametrów.

☛ *AdminSPA*: `NARZĘDZIA` > `Szablony` > `Szablony raportów`

Funkcjonalność projektowania szablonów raportów w nAxiom opiera się na module Progress® Telerik® Reporting firmy Telerik. Szczegółowe instrukcje projektowania raportów można znaleźć na stronie https://docs.telerik.com/reporting/creating-the-report.

Polecenie `Szablony raportów` wyświetla stronę z listą zdefiniowanych szablonów raportów. W ostatniej kolumnie w każdym wierszu są dostępne przyciski:

- `Designer`: otwiera w nowej stronie przeglądarki komponent `Web Report Designer`, w którym można edytować szablon raportu.
- `Edytuj`: wyświetla okno edycji definicji szablonu raportu, w którym można edytować podstawowe dane szablonu.
- `Usuń`: usuwa definicję szablonu.

TWORZENIE NOWEGO RAPORTU

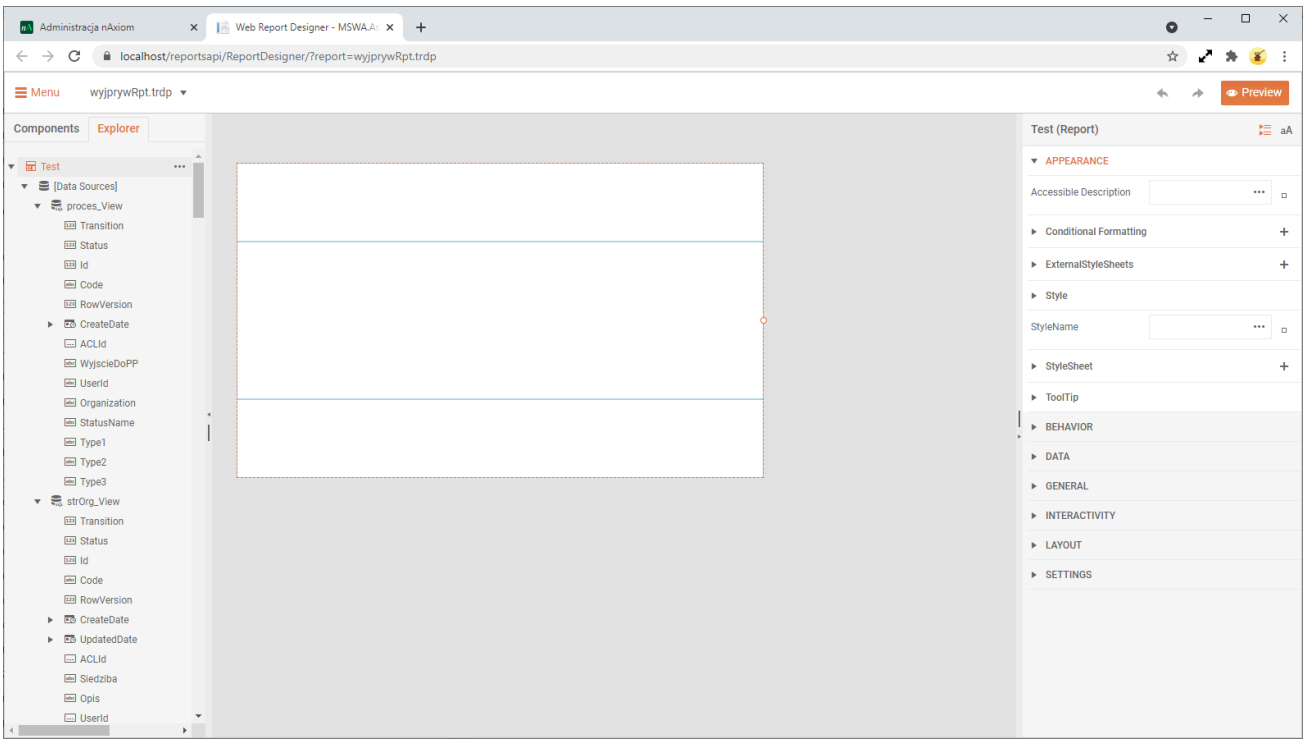
1. Na stronie `Szablony raportów` kliknij przycisk `Nowy raport` w lewym górnym rogu tej strony.
   Zostanie wyświetlone okno dialogowe `Nowy raport`.
2. Wypełnij pola `Kod`, `Nazwa`, `Aplikacja` i `Moduł`, a następnie kliknij przycisk `Zapisz`.
3. Kliknij przycisk `Designer` w lewym dolnym rogu okna dialogowego, aby uruchomić moduł Telerik Report Designer na nowej karcie.
4. Zaprojektuj szablon raportu zgodnie z opisem w sekcji Projektowanie raportów .
5. Aby zabaczyć podgląd raportu, kliknij przycisk `Preview` (Podgląd) w prawym górnym rogu. Przełączenie do trybu podglądu powoduje automatyczne zapisanie zmian. Aby wymusić zapisanie zmian, kliknij ikonę zwiniętego menu w lewym górnym rogu i kliknij polecenie `Save` (Zapisz).

WSKAZÓWKA
Definicje szablonów raportów są zapisywane w bazie danych w tabelach *core.ReportDefinitions*, *core.TemplateFiles* i *core.FileContent*

6.3.1. Web Report Designer

Podsumowanie:
Web Report Designer to niezależna aplikacja do projektowania szablonów raportów. W tym temacie i następnym opisano pokrótce najważniejsze elementy i sposób projektowania prostego raportu. Pełną dokumentację designero można znaleźć na stronach producenta pod adresem: https://docs.telerik.com/reporting/creating-the-report.

☛ *AdminSPA*: `NARZĘDZIA` > `Szablony` > `Szablony raportów` > `Designer`



Report Designer

Na stronie `Web Report Designer` można wyróżnić następujące elementy (od prawej strony u góry):

- rozwijane menu z poleceniami, między innymi:
  - `New report` (Nowy): tworzy nowy raport.
  - `Open` (Otwórz): otwiera raport.
  - `Save` (Zapisz): zapisuje zmiany w projekcie raportu.
  - `Save As` (Zapisz jako): zapisuje raport pod nową nazwą.
  - `Close` (Zamknij): zamyka projekt raportu.
  - `Workspace preferences` (Preferencje obszaru roboczego): pozwala określić preferencje w kwestii przyciągania elementów raportu do linii siatki oraz do linii przyciągania.

> **UWAGA**
> Polecenie `Assets Manager` (Menedżer zasobów) umożliwia korzystanie w raportach z zapisanych lokalnie plików, np. obrazów, jednak ta funkcjonalność nie jest obsługiwana w nAxiom. Użyte w raporcie pliki lokalne nie będą widoczne w raportach generowanych z witryny nAxiom, nie będą także migrowane. Aby używać obrazów w raportach, które nie są załącznikami i nie są zapisane w bazie danych, należy je osadzić w szablonie raportu jako *Embeded Image*.

- nazwa pliku raportu: nazwa pliku raportu jest domyślnie tworzona z kodu raportu i rozszerzenia *trdp*; pliki raportów są domyślnie zapisywane w systemowych tabelach bazy danych.
- przyciski `Undo` (Cofnij) i `Redo` (Ponów).
- przycisk `Preview` (Podgląd), który wyświetla podgląd raportu; w trybie podglądu ten przycisk zmienia się na `Design` (Projekt) i służy do powrotu do trybu projektowania.

Poniżej znajdują się trzy panele:

- panel z zakładkami `Components` (Komponenty) i `Explorer` (Eksplorator) z lewej strony.
- panel z projektem raportu w środku.
- panel właściwości wybranego elementu z prawej strony.

Strona raportu widoczna na panelu pośrodku składa się z trzech sekcji:

- nagłówka
- szczegółów
- stopki

Przykład projektowania prostego raportu opisano w kolejnej sekcji.

6.3.2. Projektowanie raportów

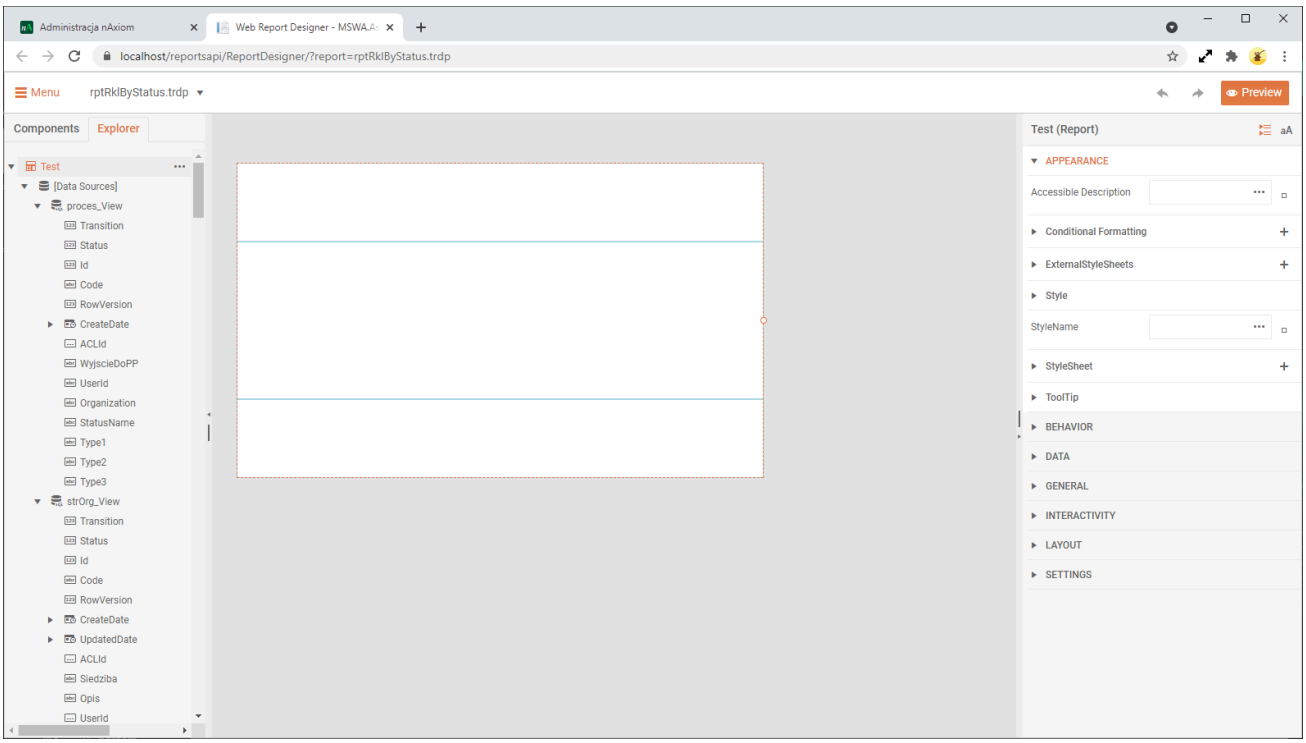
W TYM TEMACIE

- Projekt prostego raportu
- Definiowanie parametrów dla raportu
- Generowanie reportów
- Eksport do pliku Excel

PROJEKT PROSTEGO RAPORTU

W tej sekcji opisano przykład projektu prostego raportu w formie tabeli z parametrem oraz wykorzystanie go do wygenerowania raportu w przykładowej aplikacji nAxiom.

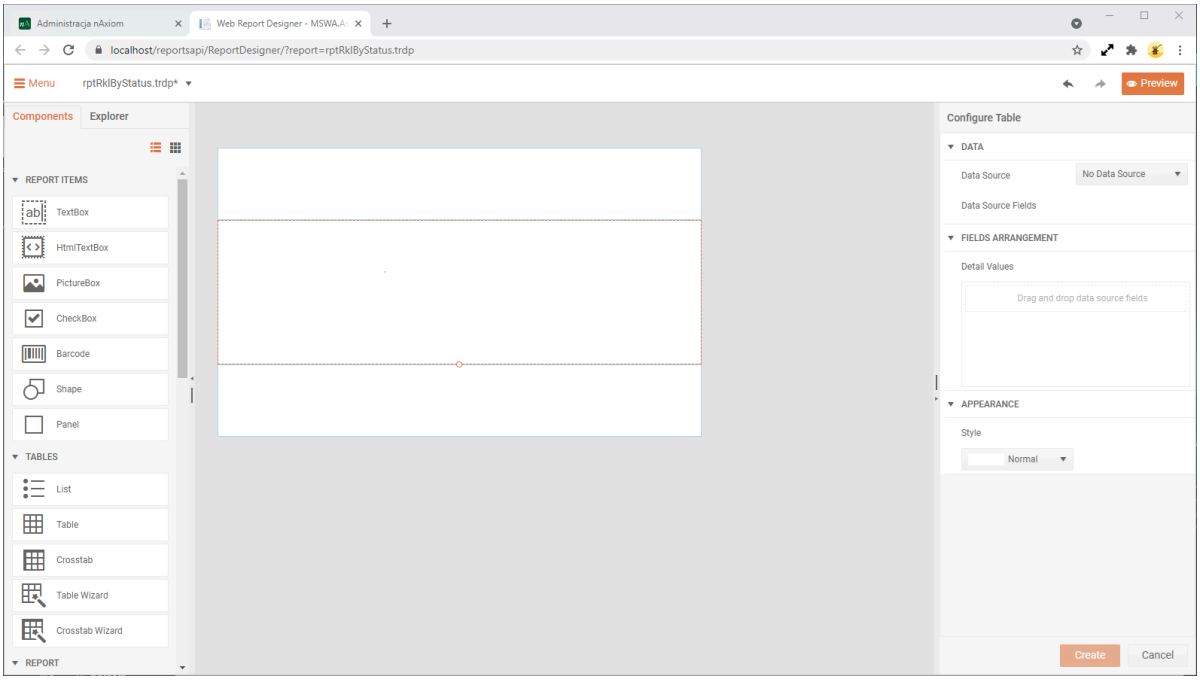
Po zadeklarowaniu nowego raportu na stronie `Raporty` (patrz Raporty) na osobnej karcie przeglądarki zostanie wyświetlona strona `Web Report Designer`.



Report Designer

1. Kliknij sekcję szczegółów (środkową) w projekcie raportu na środkowym panelu.

2. Kliknij zakładkę `Components` (Komponenty) na panelu z lewej strony, znajdź pozycję `Table Wizard` (Kreator tabel) i przeciągnij ją na sekcję szczegółów projektu raportu.
   Na panelu z prawej strony zostaną wyświetlone właściwości potrzebne do skonfigurowania tabeli raportu.

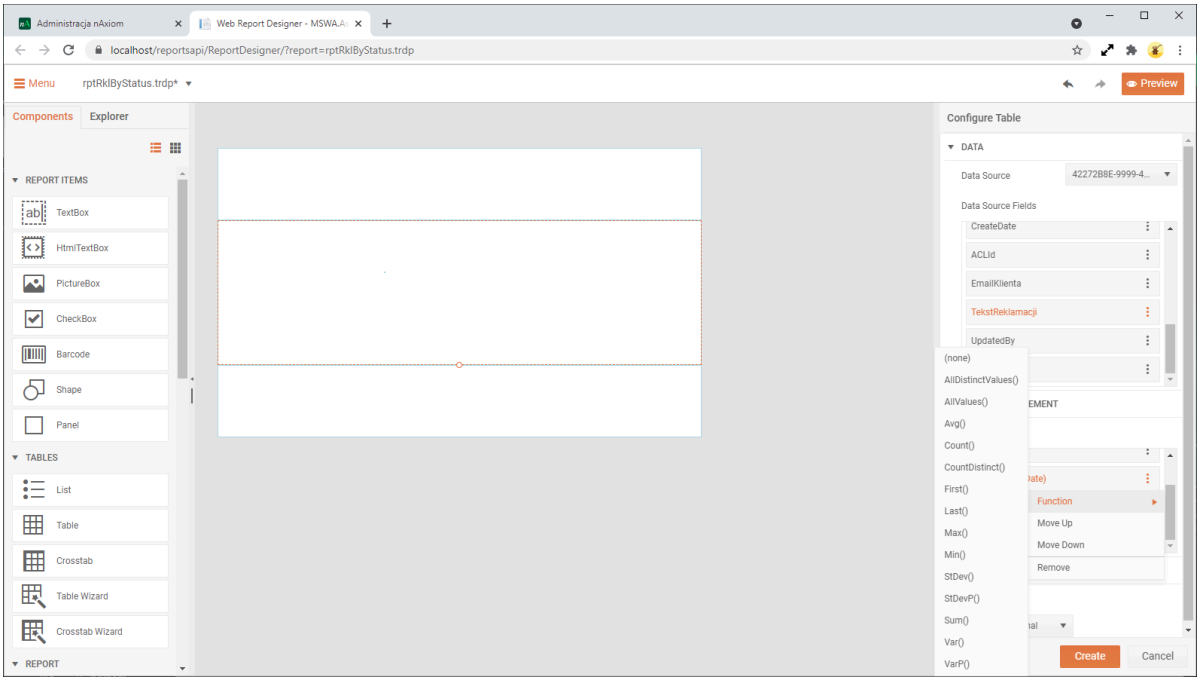


Konfigurowanie raportu tabelarycznego

3. Kliknij listę rozwijaną `Data Source` (Źródło danych) i wybierz tabelę dokumentu biznesowego lub widok, który będzie źródłem danych dla raportu.
   W obszarze `Data Source Fields` (Pola źródła danych) zostaną wyświetlone nazwy kolumn ze źródła danych.

4. Przeciągnij nazwy kolumn, które mają być wyświetlane w raporcie do obszaru `Detail Values` (Wartości szczegółów) w sekcji `FIELDS ARRANGMENT` (Uporządkowanie pól); możesz także kliknąć menu obok nazwy kolumny i wybrać polecenie `Add to Detail Values` (Dodaj do wartości szczegółów).

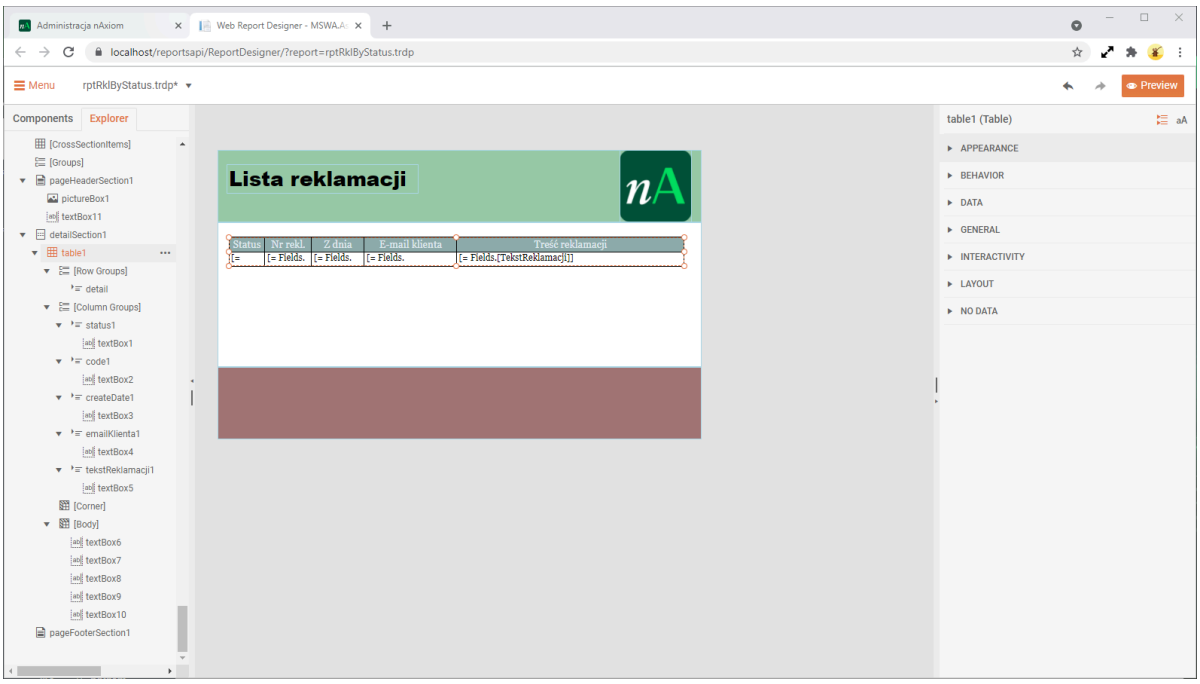
5. Usuń funkcje agregujące dodawane domyślnie przez designera, w tym celu kliknij przycisk menu obok nazwy kolumny dodanej do raportu, kliknij menu podrzędne `Function` (Funkcja) i kliknij pozycję `(none)` (brak).



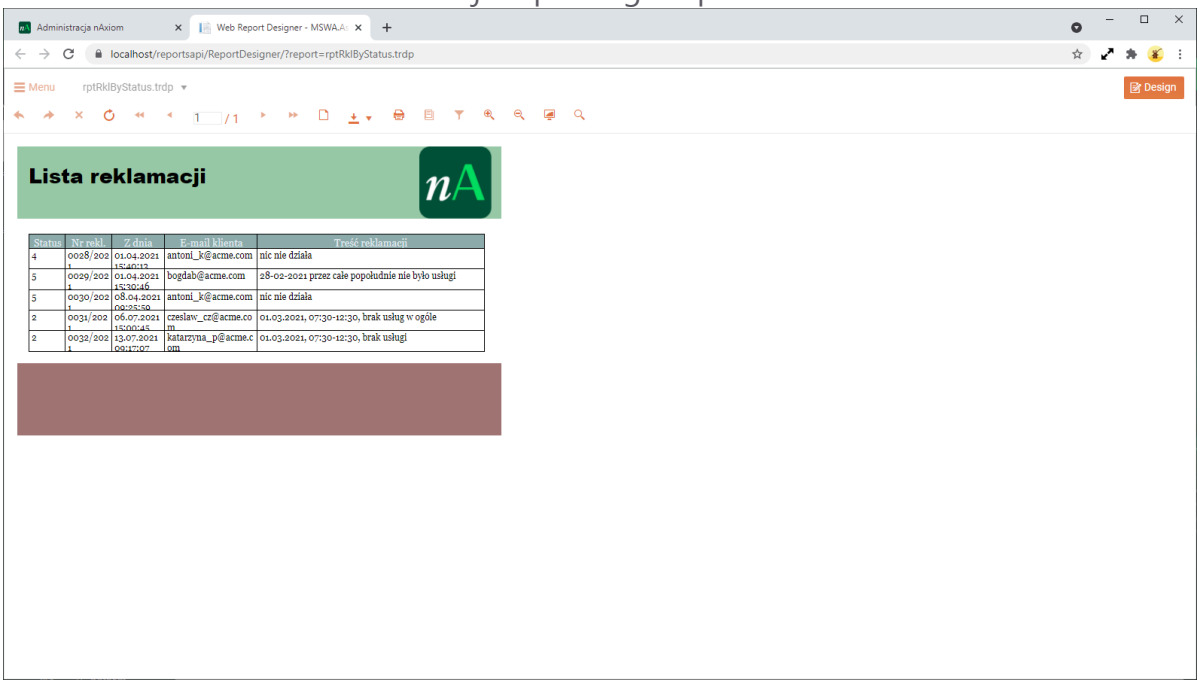
Usuwanie funkcji agregujących z kolumn wybranych do raportu

6. (Opcjonalnie) Wybierz jeden z predefiniowanych stylów z listy `Style` (Styl) w sekcji `APPEARANCE` (Wygląd).
7. Kliknij przycisk `Create` (Utwórz). W sekcji szczegółów zostanie wyświetlona zaprojektowana tabela w formie wiersza z nagłówkami kolumn i wiersza z polami danych.

8. Opracuj projekt raportu pod względem wizualnym, dodaj tytuł, logo, kolory i style.



Projekt prostego raportu



Podgląd raportu

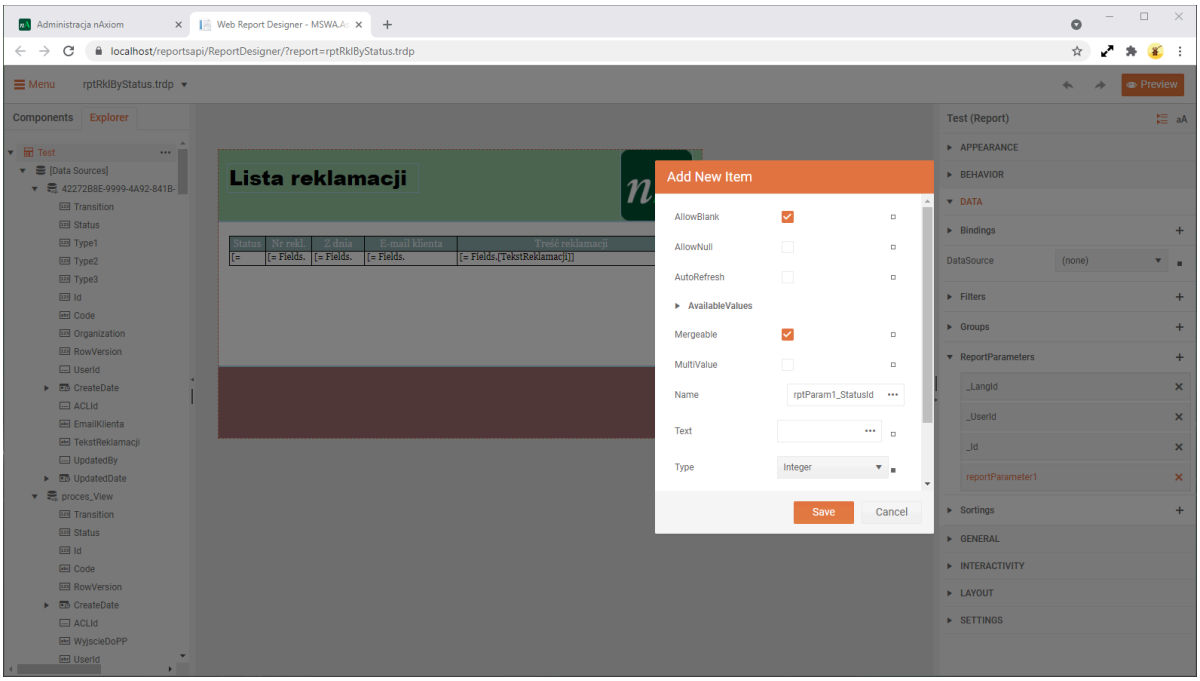
DEFINIOWANIE PARAMETRÓW DLA RAPORTU

1. Kliknij zakładkę `Explorer` na panelu z lewej strony i kliknij element główny projektu raportu o nazwie `Test`.
2. Na panelu z prawej strony rozwiń kolejno sekcje `DATA` (Dane) i `ReportParameters` (Parametry raportu); w tej sekcji znajdują się predefiniowane parametry:
    - `_LangId`: identyfikator języka;
    - `_UserId`: identyfikator bieżącego użytkownika;

- `_Id`: identyfikator bieżącego dokumentu.

3. Kliknij znak `+` z prawej strony nagłówka sekcji `ReportParameters`.
   Zostanie wyświetlone okno dialogowe `Add New Item` (Dodaj nową pozycję).

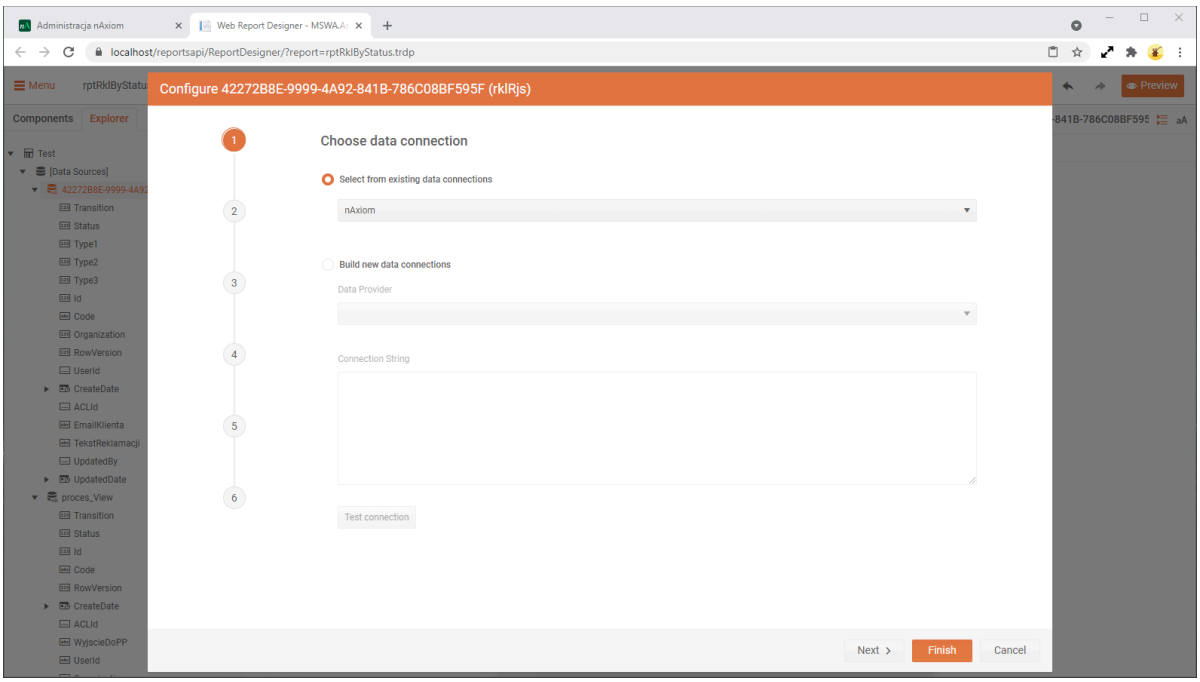
4. Wpisz nazwę parametru w polu `Name` (Nazwa), wybierz typ z listy `Type` (Typ) i kliknij przycisk `Save` (Zapisz).
   Parametr zostanie dodany do listy.



Dodawanie parametru

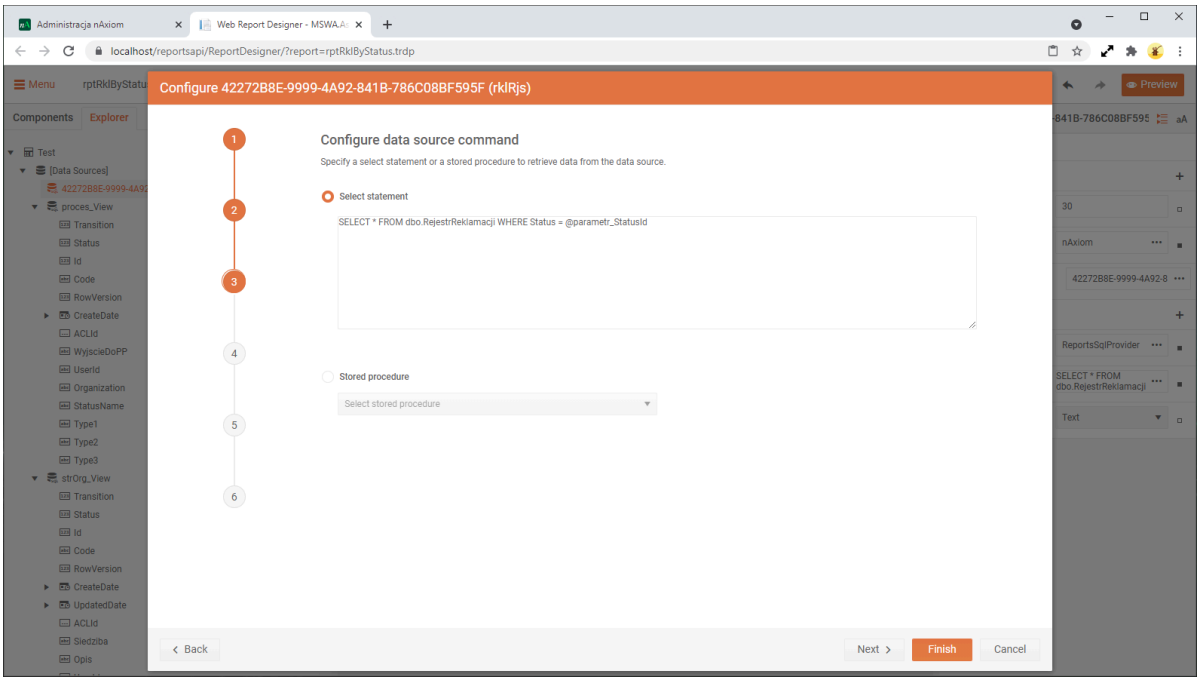
5. Zmodyfikuj źródło danych dla raportu, aby uwzględnić parametr. W tym celu na panelu z lewej strony kliknij tabelę źródłową raportu — pierwsza pozycja od góry w sekcji `Data Sources` (Źródła danych) i kliknij przycisk z wielokropkiem z prawej strony nazwy tabeli. Z wyświetlonego menu wybierz polecenie `Edit` (Edytuj).
   Zostanie wyświetlone okno dialogowe `Configure [nazwa tabeli]` (Konfiguruj [nazwa tabeli]), w którym konfiguruje się źródło danych w sześciu krokach.

> **WSKAZÓWKA**
> W celu wyświetlenia przycisku menu może być konieczne zwiększenie szerokości panelu z prawej strony.



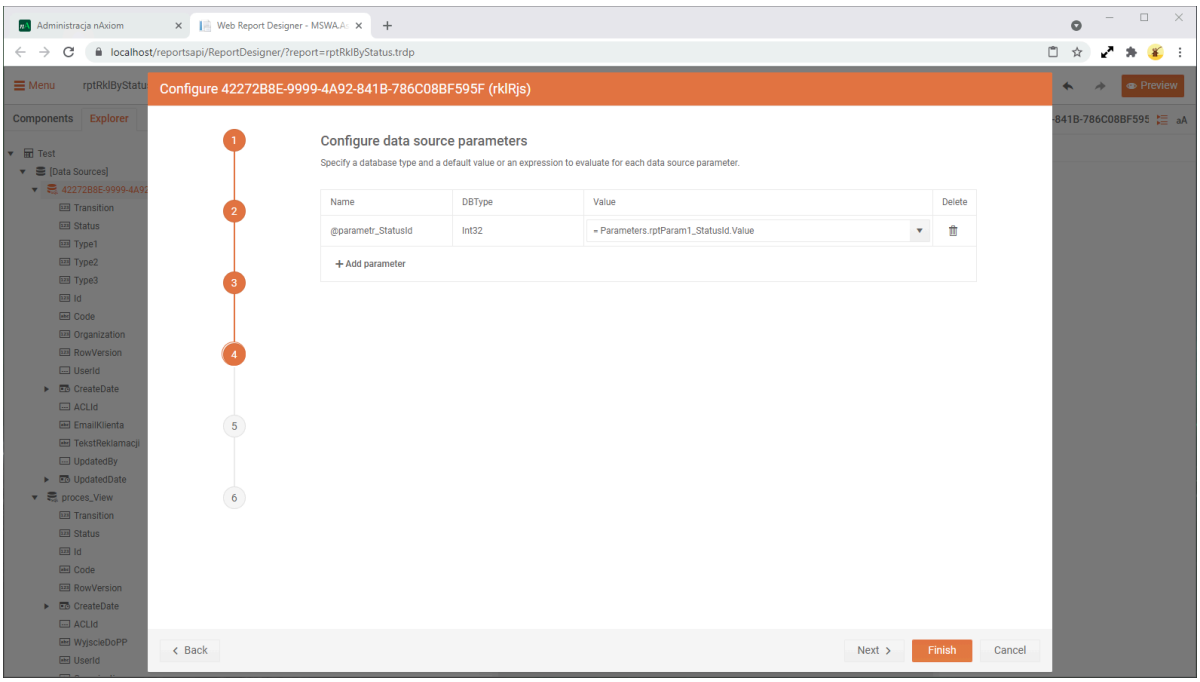
Konfigurowanie źródła danych

6. Przejdź do kroku 3 i dopisz do zapytania SQL warunek `WHERE` z użyciem parametru zapisanego w formie ***@NazwaParametru***, gdzie ***NazwaParametru*** to dowolna nazwa.



Warunek z parametrem

7. Przejdź do kroku 4 i zdefiniuj przypisanie wartości parametrowi użytemu w zapytaniu. W tym celu w kolumnie `DBType` (Typ BD) wybierz typ parametru, a w kolumnie `Value` (Wartość) wybierz z listy parametr zdefiniowany dla raportu.

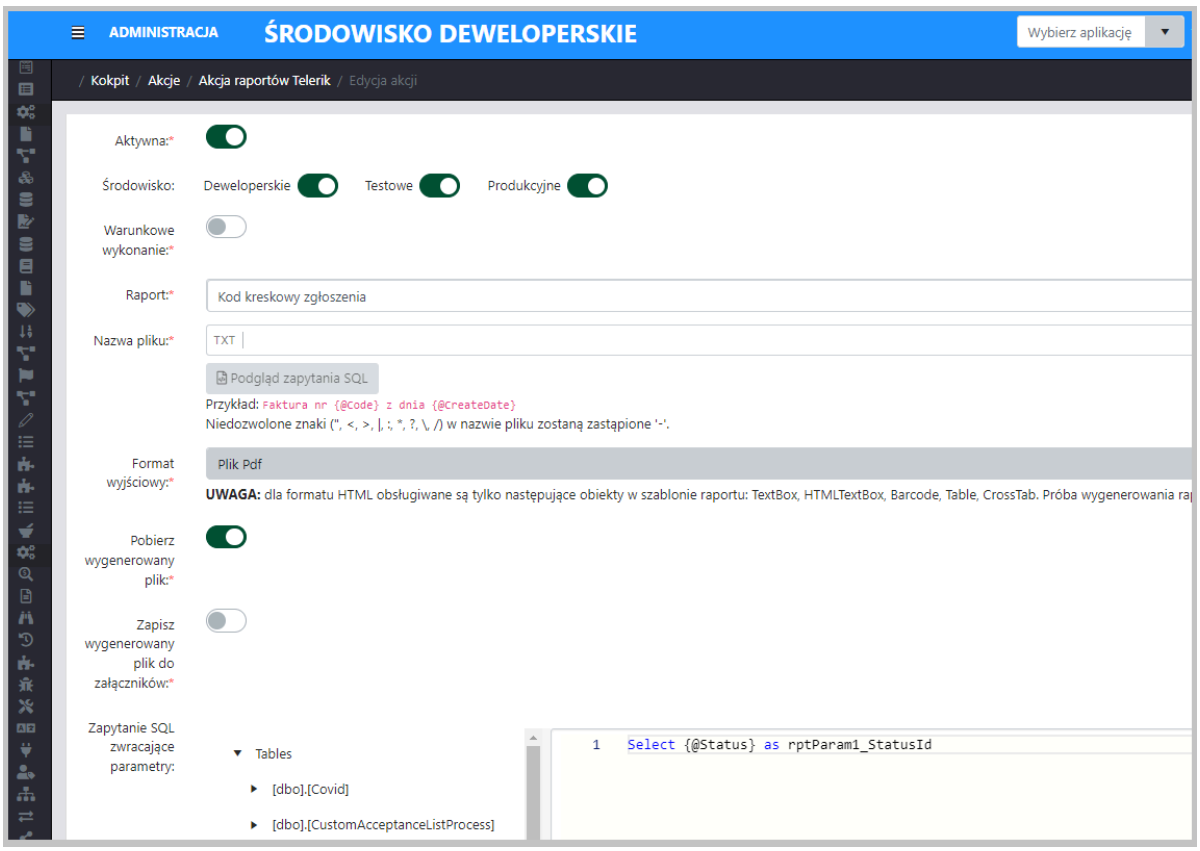


Konfiguracja parametrów źródła danych

8. Opcjonalnie, w kroku 5. podaj wartość parametru, która będzie używana do wyświetlenia podglądu raportu.
9. Opcjonalnie w kroku 6. sprawdź poprawność konfiguracji. W tym celu kliknij przycisk `Execute Query` (Wykonaj zapytanie).
10. Kliknij przycisk `Finish` (zakończ), zapisz projekt raportu i wróć do modułu back-office w nAxiom.

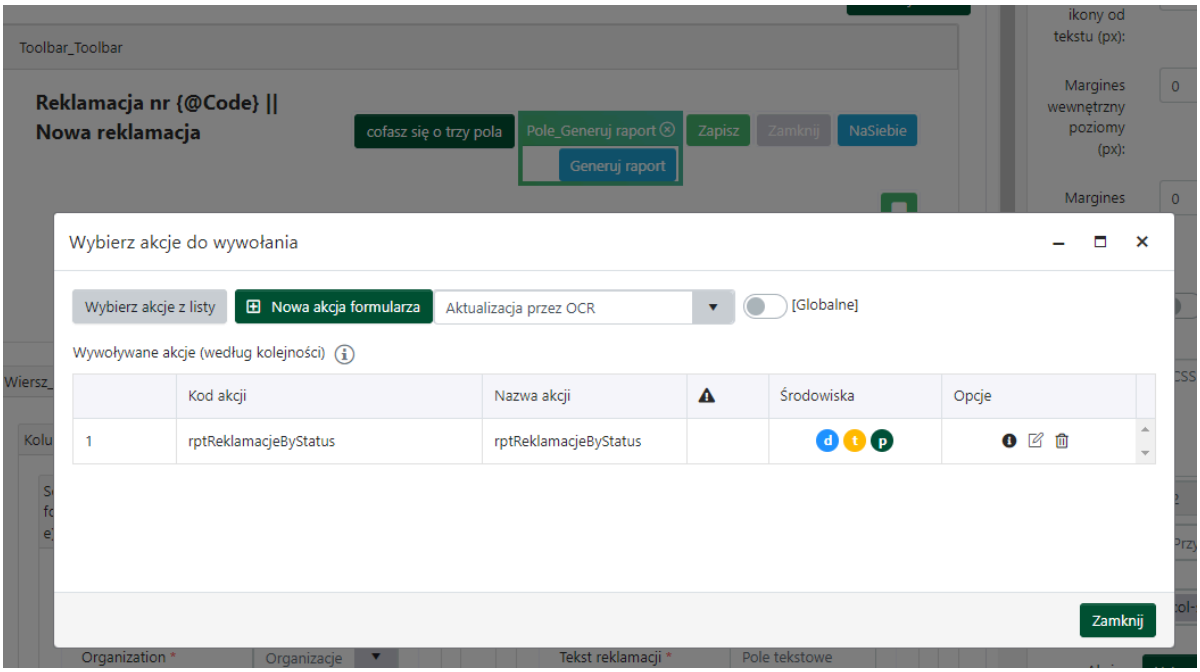
GENEROWANIE REPORTÓW

1. Zdefiniuj akcję raportów Telerik, która będzie generować zaprojektowany raport; patrz Raportowanie . W zapytaniu użyj nazwy parametru zdefiniowanej w projekcie raportu.



Akcja raportów Telerik

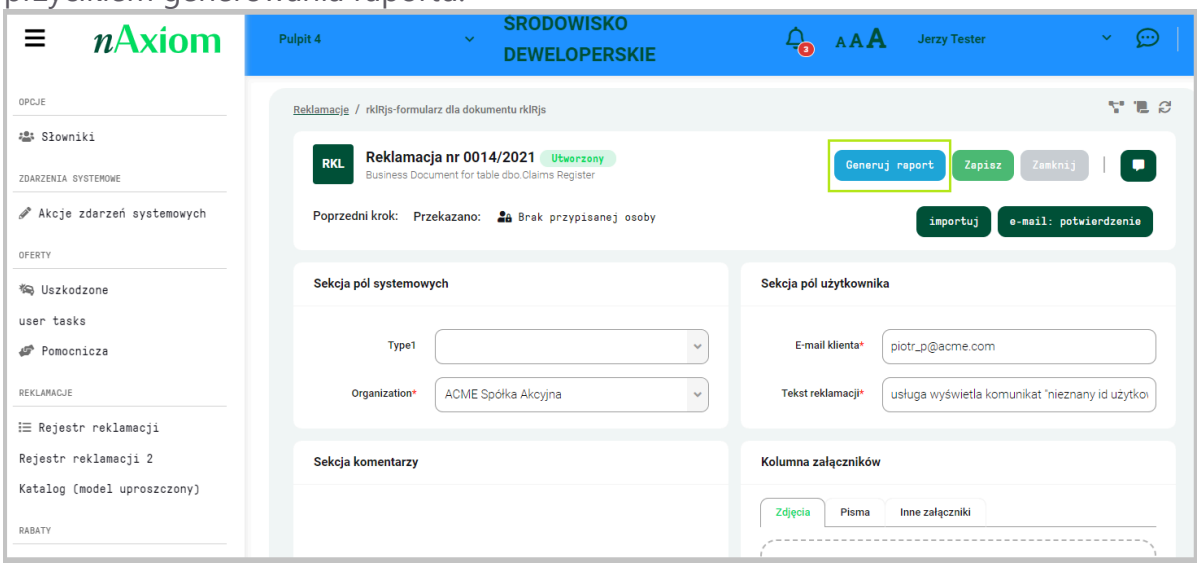
2. Na wybranym formularzu umieść przycisk, który będzie uruchamiał akcję zdefiniowaną w poprzednim kroku. W przykładzie prezentowanym na ilustracjach przycisk generuje zestawienie dokumentów w tym samym statusie, co dokument bieżący.



Projekt formularza, akcja raportu Telerik przypisana do przycisku

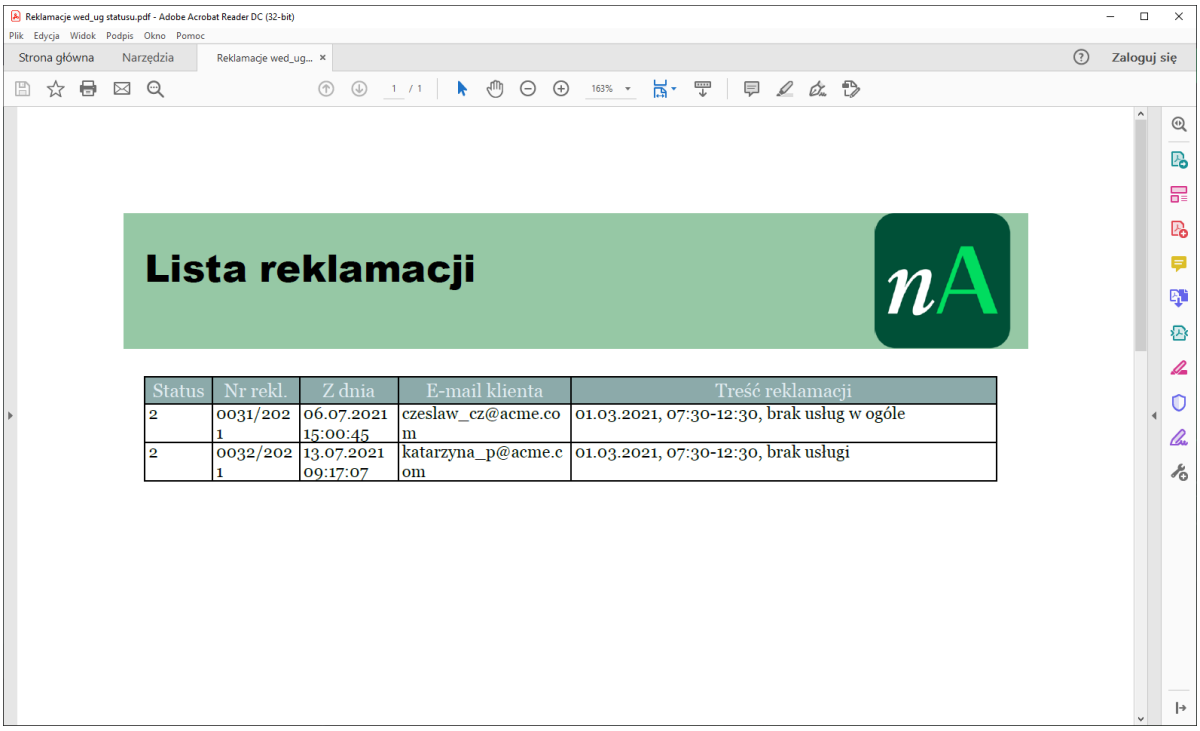
3. Zaloguj się do komponentu front-office jako użytkownik z odpowiednimi

uprawnieniami, aby przetestować generowanie raportu. Wyświetl formularz z przycikiem generowania raportu.



Formularz z przyciskiem do generowania raportu

4. Kliknij przycisk `Generuj raport` na formularzu, aby wygenerować raport z zestawieniem dokumentów w tym samym statusie, co dokument bieżący. Wygenerowany raport zostanie pobrany w przeglądarce, a następnie wyświetlony w programie ustawionym jako domyślny do obsługi formatu PDF.



Status	Nr rekl.	Z dnia	E-mail klienta	Treść reklamacji
2	0031/2021	06.07.2021 15:00:45	czeslaw_cz@acme.com	01.03.2021, 07:30-12:30, brak usług w ogóle
2	0032/2021	13.07.2021 09:17:07	katarzyna_p@acme.com	01.03.2021, 07:30-12:30, brak usługi

Podgląd raportu

EKSPORT DO PLIKU EXCEL

Moduł raportów Telerik oferuje możliwość wyeksportowania raportu do pliku programu Excel. Ta funkcja wymaga odpowiedniego podejścia do formatu eksportowanych danych. Domyślnie moduł raportów eksportuje dane i przesyła tzw. rozszerzone informacje o formacie, które jednak nie są prawidłowo interpretowane przez program Excel. Ustawienie to można wyłączyć w pliku *appsettings.json* serwisu *reportsapi*, dodając sekcję:

```
"telerikReporting": {
  "extensions": [
    {
      "name": "XLSX",
      "parameters": [
        {
          "name": "UseExtendedFormatting",
          "value": "False"
        }
      ]
    }
  ]
}
```

Jednak ze względu na potwierdzony błąd, nie działa to prawidłowo dla typów danych *datetime* i *time*. W tym przypadku Excell identyfikuje formatowanie danych jako *Niestandardowe*.

Należy zauważyć, że bez względu na opis, typ danych jest zawsze prawidłowo interpretowany przez program Excel. Należy także zwrócić uwagę na fakt, że zastosowanie do wartości dowolnego pola funkcji *Format()* lub *FormatwithCulture()* spowoduje przekształcenie tej wartości w tekst (to samo dotyczy innych funkcji w kategorii *Text*).

Aby prawidłowo sformatować wartości należy dla każdego pola w szablonie raportu wpisać odpowiedni kod formatu w pole `Format` na panelu właściwości. Poniżej podano przykłady:

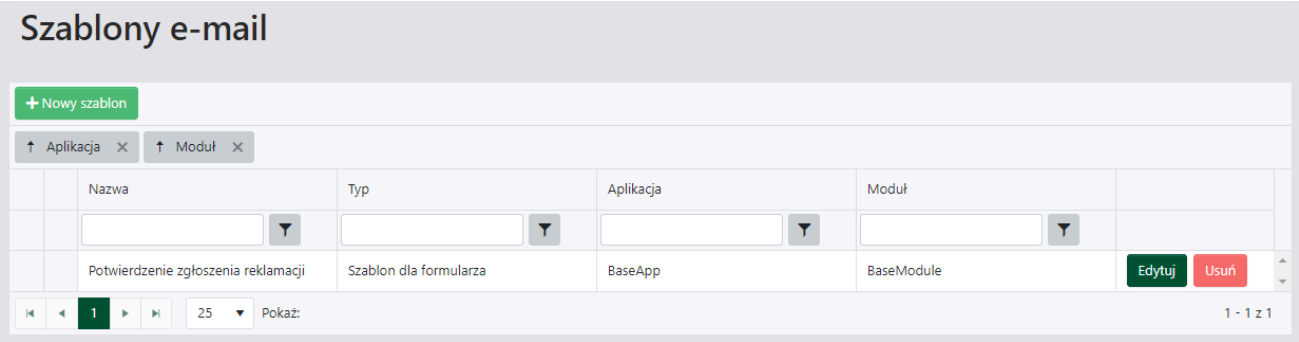
- *{0:p3}*: wartości procentowe
- *{0:c}*: wartości walutowe
- *{0:N0}*: wartości liczbowe; liczba na prawo od symbolu formatowania określa liczbę miejsc po znaku dziesiętnym, jeśli nie zostanie podana, wartości liczbowe będą wyświetlane z dwoma miejscami po przecinku (sama wartość nie zostanie zaokrąglona)

Jeśli formatowanie w ogóle nie zostanie zastosowane, wartości liczbowe, tekstowe i logiczne będą wyświetlane przez Excel jako dane w formatowaniu *Ogólne*. Wartości daty i czasu są zawsze wyświetlane jako *Niestandardowe*. Jak już wspomniano, niezależnie od identyfikacji przez Excel, wszystkie wartości mogą być przetwarzane zgodnie ze swoim typem danych.

6.4. SZABLONY E-MAIL

Podsumowanie:
Funkcja szablonów e-mail pozwala definiować szablony wiadomości e-mail, które można wypełniać danymi z dokumentów biznesowych i wysyłać pocztą elektroniczną. Szablony można również tworzyć i edytować z poziomu akcji akcji wysyłki e-mail.

☛ *AdminSPA* > `NARZĘDZIA` > `Szablony` > `Szablony e-mail`; *AdminSPA* > `NARZĘDZIA` > `Akcje` > `Wyślij/Odbierz e-mail`



Lista szablonów e-mail

Przycisk `Nowy szablon` otwiera okno dialogowe `Dodanie szablonu`, w którym należy podać ustawienia szablonu e-mail zgodnie z opisem poniżej. Przyciski `Edycja` i `Usuń` służą odpowiednio do edytowania ustawień szablonu e-mail oraz do usunięcia szablonu e-mail z listy.

UWAGA
Korzystanie z szablonu e-mail wymaga zdefiniowania akcji wysyłki e-mail.

OKNO `DODANIE/EDYCJA SZABLONU`

Dodanie szablonu

Nazwa:

Aplikacja: *

Moduł: *

Typ: Szablon dla formularza

Opis:

Edytor HTML:

Aby w oknie **Edytor HTML** skorzystać z modelu danych powiązanego z formularzem użyj składni: **{@NazwaPola}**
Aby zamiast indeksów słownika skorzystać z wartości nazw ze słownika, lub innych danych umieszczonych poza modelem danych powiązanych z formularzem:

- Utwórz zapytanie SQL w oknie **Edytor SQL Query**
- Umieść w obrębie edytora HTML element **<RowTemplate> </RowTemplate>**
- Wewnątrz elementu **<RowTemplate> </RowTemplate>** umieść parametr **{@Query_NazwaPola}**, gdzie **NazwaPola** jest nazwą kolumny lub jej aliasu zwracanej przez zapytanie SQL definiowane w oknie **Edytor SQL Query**

Element **<RowTemplate> </RowTemplate>** jest wymagany do obsługi pojedynczego rekordu oraz listy rekordów zwracanych przez zapytanie SQL.
Aby dodać tabelę dla zwróconych wielu wierszy np.:

```
select 'w1 c1' as col1, 'w1 c1' as col2
union all
select 'w2 c1', 'w2 c2'
```

należy utworzyć szablon:

```
<table>
 <tr>
  <th>Nagłówek A</th>
  <th>Nagłowek B</th>
 </tr>
 <RowTemplate>
  <tr>
     <td>{@Query_col1}</td>
     <td>{@Query_col2}</td>
  </tr>
 </RowTemplate>
</table>
```

Aby dodać link i ręcznie użyć zmiennych kliknij przycisk: Wklej link
Aby dodać link i wybrać istniejący formularz/listę kliknij przycisk: Utwórz link
Przykładowy link wygląda następująco: **{#Link/TYP/ID/RECORDID}**
Gdzie **TYP** to jedna z dwóch wartości: **Grid** lub **Form** oznaczające odpowiednio widok listowy i formularz
Gdzie **ID** to identyfikator listy lub formularza
Gdzie **RECORDID** to identyfikator rekordu (opcjonalny)
Aby dodać link do wtyczki MSO: Wklej link do wtyczki nAxiom dla MS Outlook
Aby dodać link do resetowania hasła użytkownika: Wklej link resetowania hasła

1

Edytor SQL Query:

Aby w oknie **Edytor SQL Query** skorzystać z modelu danych powiązanego z formularzem użyj składni: **{@NazwaPola}**

1

Załączniki:

Przeciągnij i upuść plik lub wybierz z dysku

Nazwa pliku	Przesłano	Opcje

Edytor SQL Załączników:

Aby wysłać załączniki wraz z wiadomością e-mail, zapytanie powinno zwrócić następujące kolumny: `FileNameLocal`, `FileNameOriginal`, `AttachmentRepositoryId`, `SaveLocation` oraz `RowGuid`.
Przykładowe zapytanie zwracające załączniki z obecnego dokumentu biznesowego:

```
SELECT att.FileNameLocal, att.FileNameOriginal, cat.AttachmentRepositoryId, att.SaveLocation, att.RowGuid
FROM core.Attachments AS att
INNER JOIN core.AttachmentCategories AS cat ON att.AttachmentCategoryId = cat.Id
WHERE att.BusinessDocumentId = {@DocumentId}
```

Tables
- [dbo].[Covid]
- [dbo].[CustomAcceptance
- [dbo].[Rozliczenia]

1

Odśwież

Dodanie/Edycja szablonu

W wyświetlonym oknie dialogowym należy określić szereg ustawień szablonu:

`Nazwa`: nazwa szablonu e-mail.

`Aplikacja`: nazwa aplikacji, do której zostanie przypisany tworzony obiekt; wszystkie obiekty przypisane do aplikacji można wraz z nią wyeksportować, używając polecenia `Migrator aplikacji` w sekcji `ADMINISTRACJA` menu głównego.

`Moduł`: nazwa modułu aplikacji, do którego zostanie przypisany tworzony obiekt.

`Typ`: rodzaj szablonu; obecnie obsługiwany jest tylko jeden rodzaj szablonu: `Szablon dla formularza`.

`Opis`: opis szablonu e-mail.

`Edytor HTML`: kod HTML treści wiadomości; dozwolone jest użycie wartości z tabeli źródłowej formularza przy użyciu zmiennych w formacie *{@PoleFormularza}*; aby użyć wartości z innych tabel zwróconych przez zapytanie SQL wpisane poniżej w oknie `Edytor SQL Query`, należy je przywołać wewnątrz znaczników * *, używając formatu *{@Query_NazwaKolumnySQL}*; w opisie w polu dialogowym są dostępne następujące przyciski:

- `Wklej link`: wstawia w edytorze HTML wzorzec linku, który można dostosować ręcznie;
- `Utwórz link`: generuje link do określonego dokumentu;
- `Wklej link do wyczki nAxiom dla MS Outlook`: wstawia link, który po kliknięciu zostanie obsłużony w programie MS Outlook przez wtyczkę nAxiom; referowany dokument zostanie wyświetlony we wtyczce w programie Outlook;
- `Wklej link resetowania hasła`: wstawia link, którego kliknięcie pozwala zresetować hasło użytkownika nAxiom.

`Edytor SQL Query`: pozwala zdefiniować zapytanie, które zwróci dane do zamieszczenia w wiadomości e-mail. W zapytaniu można odwoływać się do wartości z kontekstu formularza, używając składni *{@NazwaPola}*.

`Załączniki`: pole na które można upuścić załącznik lub kliknąć w celu wyświetlenia pola wyboru pliku załącznika; załączony plik będzie wysyłany wraz z wiadomością e-mail generowaną na podstawie szablonu.

`Edytor SQL załączników`: zapytanie musi zwrócić wartości z następujących kolumn: *FileNameLocal*, *FileNameOriginal*, *AttachmentRepositoryId*, *SaveLocation* oraz *RowGuid* z tabeli *core.Attachments*, odpowiadające plikowi załącznika, który ma zostać wysłany.

Aby użyć zdefiniowanego szablonu do generowania i wysyłania wiadomości e-mail, należy skonfigurować akcję wysyłki e-mail i wskazać w niej nazwę szablonu. Wykonanie akcji, na przykład przez kliknięcie na formularzu przycisku, do którego została przypisana, spowoduje wysłanie na podany adres e-mail wiadomości wygenerowanej na podstawie szablonu, wraz z załącznikiem, o ile został wskazany.

Powiązane tematy:

- Akcja wysyłki e-mail

7. INTEGRACJE

W tej sekcji opisano polecenia dostępne w sekcji menu `INTEGRACJE` w aplikacji *AdminSPA*.

7.1. MAIL MONITOR

Podsumowanie:
Mail monitor to proces, który monitoruje konto pocztowe i wykonuje akcję w chwili wykrycia nowej wiadomości na tym koncie. Obecnie jedyną obsługiwaną akcją jest utworzenie nowej instancji wskazanego dokumentu biznesowego. Ponadto mail monitor może:

- dodać załącznik z e-maila do dokumentu biznesowego;
- dodać wiadomość e-mail do dokumentu biznesowego jako załącznik w formacie *txt* lub *eml*;
- wykonać określoną akcję, np. wypełnić pola dokumentu biznesowego wartościami z wiadomości e-mail;
- usunąć/przenieść do podfolderu wiadomość e-mail po jej przetworzeniu.

☛ *AdminSPA* > `INTEGRACJE` > `Monitoring poczty e-mail`

UWAGA
Mail monitor przetwarza tylko wiadomości e-mail o statusie *nieprzeczytane*. Jeśli nowa wiadomość zostanie oznaczona jako *przeczytana* na przykład bezpośrednio w programie pocztowym, mail monitor ją pominie.

Przycisk `Nowy mail monitor` otwiera okno dialogowe `Dodawanie mail monitora`, w którym należy podać ustawienia mail monitora na kartach `Ogólne`, `Akcje` i `Postproces`, zgodnie z opisem w poniższych sekcjach. Przyciski `Edycja` i `Usuń` służą odpowiednio do edytowania ustawień mail monitora oraz do usunięcia mail monitora z listy.

7.1.1. Karta Ogólne

Podsumowanie:
Na karcie `Ogólne` konfiguruje się podstawowe ustawienia mail monitora, w tym parametry monitorowanej skrzynki pocztowej.

`Kod`: kod mail monitora.

`Nazwa`: nazwa mail monitora.

`Opis`: opis mail monitora.

`Priorytet`: kolejność wykonywania mail monitora w przypadku zdefiniowania kilku; wartość domyślna `0` (najwyższy priorytet); mail monitory o tym samym priorytecie są wykonywane według kolejności zapisu definicji w bazie danych.

`Aplikacja`: nazwa aplikacji, do której zostanie przypisany tworzony obiekt; wszystkie obiekty przypisane do aplikacji można wraz z nią wyeksportować, używając polecenia `Migrator aplikacji` w sekcji `ADMINISTRACJA` menu głównego.

`Moduł`: nazwa modułu aplikacji, do którego zostanie przypisany tworzony obiekt.

`Typ serwera`: rodzaj dostępu do serwera poczty, na którym znajduje się monitorowana skrzynka pocztowa; szczegóły konfiguracji dla poszczególnych typów opisano poniżej; dostępne wartości:

- `MS Exchange`
- `MS Graph`
- `IMAP`

> **WSKAZÓWKA**
> We wszystkich polach konfiguracyjnych dostępu do monitorowanej skrzynki pocztowej jest obsługiwana składnia smart numbers.

`Aktywny`: pozwala dezaktywować, tj. wyłączyć mail monitor.

`Weryfikuj format pliku`: (domyślnie wyłączone) po włączeniu sprawdza, czy format pliku jest zgodny z rozszerzeniem nazwy pliku.

`Akceptowalne rozszerzenia`: (dostępne tylko po włączeniu weryfikacji formatu) rozszerzenia nazw plików, które mają być obsługiwane przez mail monitor; kolejne rozszerzenia należy rozdzielać średnikiem (;).

`Maksymalna wielkość pliku`: maksymalna obsługiwana wielkość pliku załącznika w bajtach.

`Minimalna wielkość pliku`: minimalna obsługiwana wielkość pliku załącznika w bajtach.

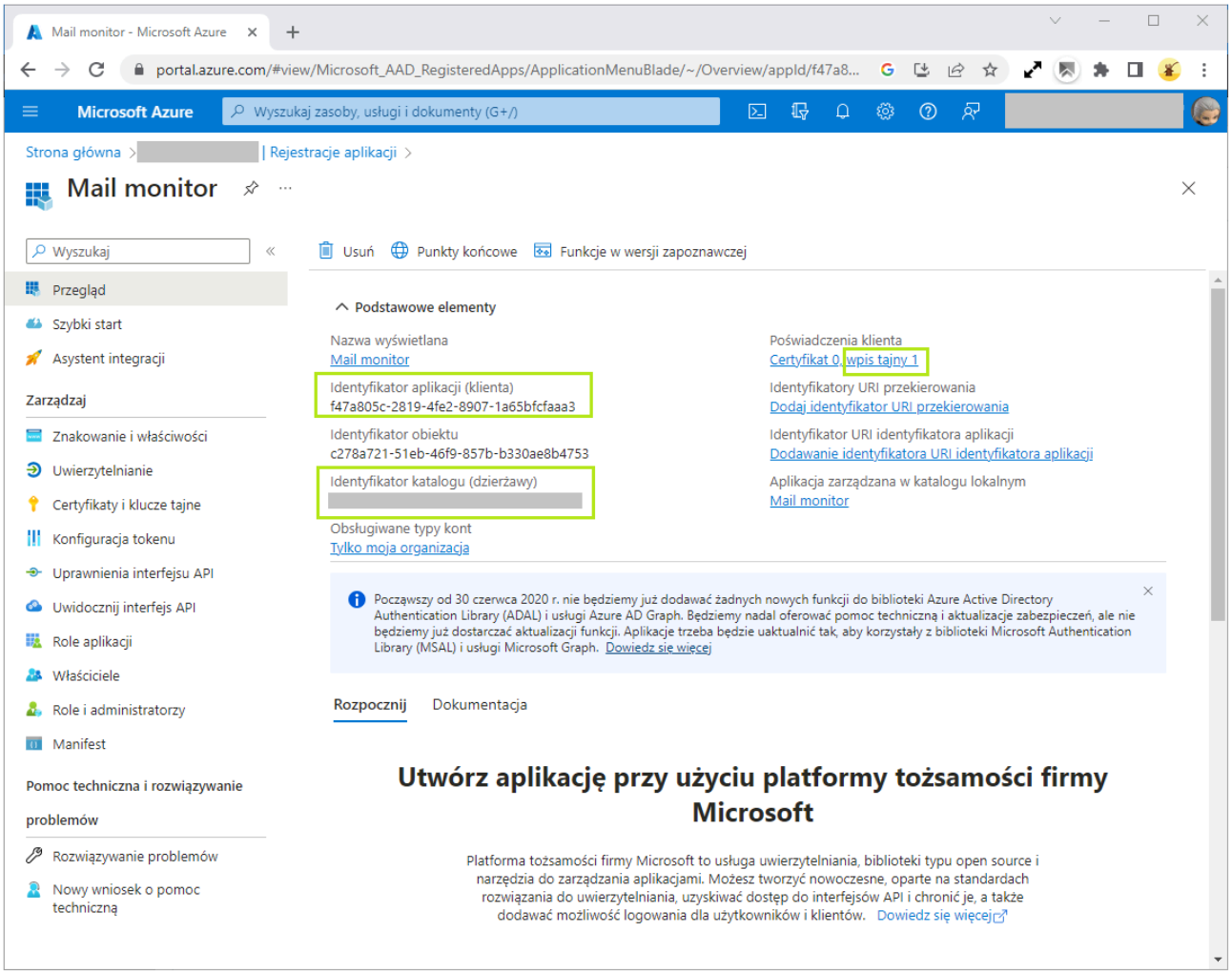
> **UWAGA**
> Wartości powyższych parametrów są porównywane z wielkością załącznika zakodowanego w wiadomości e-mail. Pliki zakodowane algorytmem *Base64* są o ok. 37% większe niż przed kodowaniem. Dlatego na przykład, aby ograniczyć przetwarzanie do plików o wielkości (przed zakodowaniem) od 100 000 B do 1 000 000 B, należy wpisać odpowiednio wartości 137 000 i 1 370 000.

`Wszystkie załączniki umieść w jednym dokumencie biznesowym`: (domyślnie wyłączone) po włączeniu dodaje wszystkie pliki załączników do tej samej instancji dokumentu biznesowego; w przeciwnym razie tworzy osobną instancję dla każdego załącznika.

`Przetwarzaj tylko wiadomości z załącznikami`: (domyślnie wyłączone) pozwala pominąć przetwarzanie wiadomości, które nie mają załączników.

Poniżej opisano szczegóły konfiguracji mail monitora zależnie od typu dostępu do serwera poczty.

DOSTĘP PRZEZ USŁUGI EXCHANGE WEB SERVICES



Dane rejestracyjne aplikacji w portalu Azure

`Typ uwierzytelniania`: zależnie od wybranej wartości wymaga określenia różnych parametrów; dostępne wartości:

- `Podstawowa autentykacja (Base64)`: metoda niewspierana obecnie przez firmę Microsoft, wymaga określenia następujących parametrów:
  - `Serwer pocztowy`: nazwa serwera, na którym znajduje się monitorowane konto pocztowe; powinien kończyć się ciągiem */EWS/Exchange.asmx*.
  - `Obserwowany folder`: nazwa folderu, w którym będą monitorowane wiadomości.
  - `Nazwa użytkownika`: adres e-mail skrzynki pocztowej.
  - `Hasło`: hasło do skrzynki pocztowej.
- `OAuth 2.0 Password`: wymaga określenia następujących parametrów:

    - `Identyfikator tenanta`: identyfikator dzierżawy z danych rejestracyjnych aplikacji w portalu Azure.
    - `Serwer pocztowy`: nazwa serwera, na którym znajduje się monitorowane konto pocztowe; powinien kończyć się ciągiem */EWS/Exchange.asmx*.
    - `Obserwowany folder`: nazwa folderu, w którym będą monitorowane wiadomości.
    - `Nazwa użytkownika`: adres e-mail skrzynki pocztowej.
    - `Hasło`: hasło do skrzynki pocztowej.
    - `ClientId`: identyfikator aplikacji (klienta) z danych rejestracyjnych aplikacji w portalu Azure.
    - `Scope`: zakres uprawnień; domyślnie *https://outlook.office365.com/.default*; ten parametr jest wymagany.
- `OAuth 2.0 Client Credentials`: metoda zalecana w przypadku uwierzytelniania aplikacji, wymaga określenia następujących parametrów:
    - `Identyfikator tenanta`: identyfikator dzierżawy z danych rejestracyjnych aplikacji w portalu Azure.
    - `Serwer pocztowy`: nazwa serwera, na którym znajduje się monitorowane konto pocztowe; powinien kończyć się ciągiem */EWS/Exchange.asmx*.
    - `Obserwowany folder`: nazwa folderu, w którym będą monitorowane wiadomości.
    - `Nazwa użytkownika`: adres e-mail skrzynki pocztowej.
    - `ClientId`: identyfikator aplikacji (klienta) z danych rejestracyjnych aplikacji w portalu Azure.
    - `ClientSecret`: wpis tajny aplikacji (klienta) z danych rejestracyjnych aplikacji w portalu Azure; należy pamiętać o okresowej aktualizacji ze względu na określony termin ważności.
    - `Scope`: zakres uprawnień; domyślnie *https://outlook.office365.com/.default*; ten parametr jest wymagany.

DOSTĘP PRZEZ INTERFEJS MS GRAPH API

`Identyfikator tenanta`: identyfikator dzierżawy z danych rejestracyjnych aplikacji w portalu Azure.

`Obserwowany folder`: nazwa folderu, w którym będą monitorowane wiadomości.

`Nazwa użytkownika`: adres e-mail skrzynki pocztowej.

`ClientId`: identyfikator aplikacji (klienta) z danych rejestracyjnych aplikacji w portalu Azure.

`ClientSecret`: wpis tajny aplikacji (klienta) z danych rejestracyjnych aplikacji w portalu Azure; należy pamiętać o okresowej aktualizacji ze względu na określony termin ważności.

`Scope`: zakres uprawnień; domyślnie *https://graph.microsoft.com/.default*; ten parametr jest wymagany.

DOSTĘP PRZY UŻYCIU PROTOKOŁU IMAP

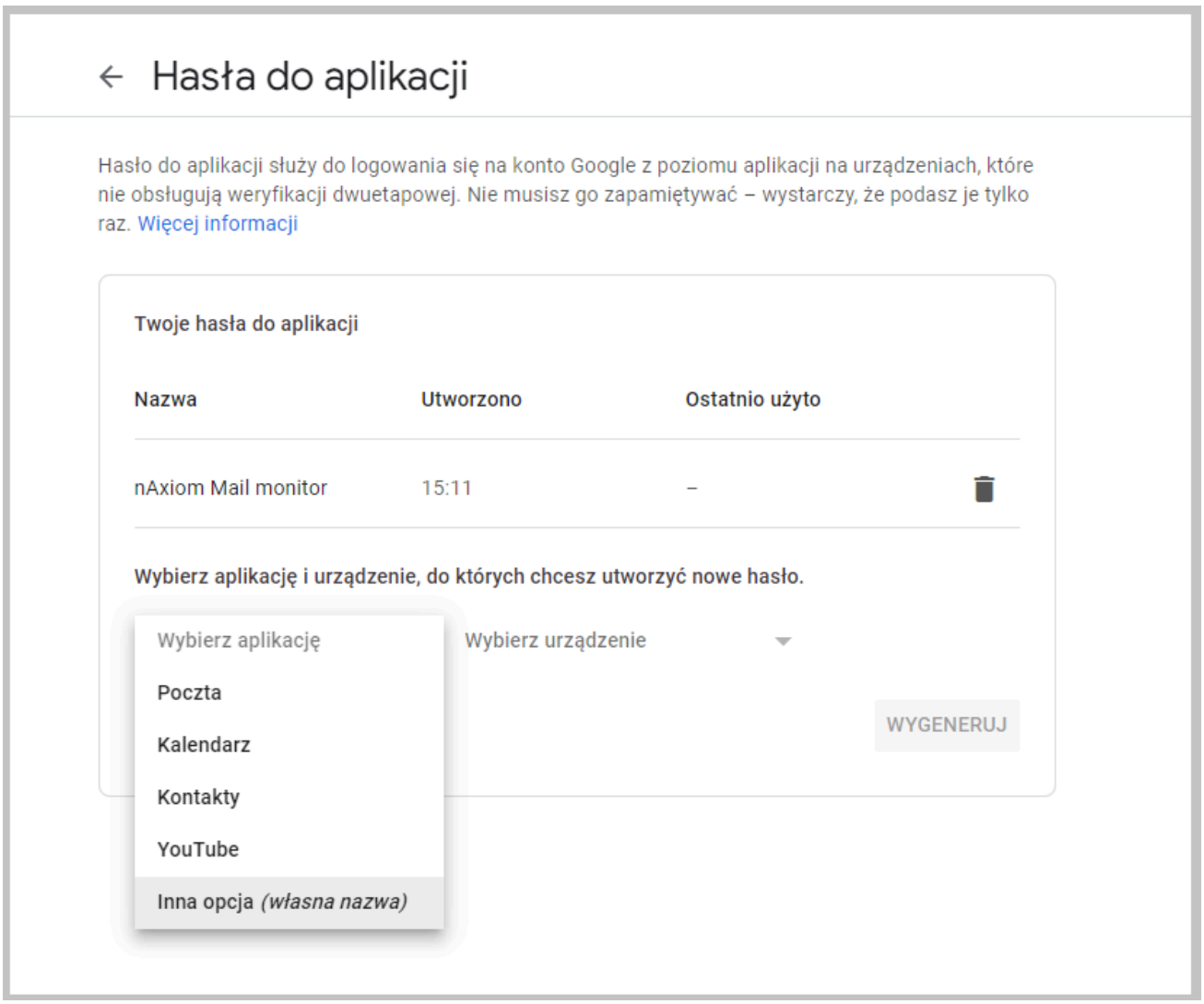
`Serwer pocztowy`: nazwa serwera, na którym znajduje się monitorowane konto pocztowe.

`Obserwowany folder`: nazwa folderu, w którym będą monitorowane wiadomości.

`Port`: numer portu na serwerze.

`Nazwa użytkownika`: adres e-mail skrzynki pocztowej.

`Hasło`: hasło do skrzynki pocztowej. Jeśli dla danego dostawcy jest włączona weryfikacja dwuetapowa, konieczne jest zarejestrowanie aplikacji i uzyskanie dla niej osobnego hasła. W przypadku konta Google należy przejść do ustawień konta i kliknąć kolejno `Bezpieczeństwo` > `Hasła aplikacji`. Na wyświetlonej stronie należy utworzyć hasło dla nowej aplikacji.



Tworzenie hasła dla aplikacji na koncie Google

Powiązane tematy:

- Integracja nAxiom z usługami Microsoft 365

7.1.2. Karta Akcje

Podsumowanie:
Na karcie `Akcje` można wskazać definicję dokumentu biznesowego, którego instancja ma zostać utworzona po przetworzeniu wiadomości, oraz określić, czy załącznik e-maila ma zostać dodany jako załącznik do dokumentu, a także wskazać kategorię i sposób zapisu załącznika.

W przypadku nowego mail monitora na karcie `Akcje` znajduje się tylko przycisk `Dodaj akcję`. Po jego kliknięciu są wyświetlane kontrolki umożliwiające skonfigurowanie opisanych poniżej ustawień. Obecnie w mail monitorze można dodać tylko jedną akcję.

- `Typ`: typ akcji wykonywanej przez mail monitor, obecnie dostępny jest tylko jeden typ akcji: `Utwórz nowy proces do obsługi dokumentu`; ta akcja powoduje utworzenie nowej instancji dokumentu biznesowego po wykryciu nieprzeczytanej wiadomości e-mail w skonfigurowanej skrzynce pocztowej.
- `Dokument biznesowy`: definicja dokumentu, którego instancję ma utworzyć mail monitor.
- `Ustaw wartości pól dokumentu`: tabelka, w której można podawać wartości pól tworzonej instancji dokumentu; można przypisywać wartości stałe lub zwracane przez zapytanie SQL, można korzystać ze zmiennych systemowych, np. *{@_UserId}* oraz ze składni smart numbers, ponadto można korzystać z następujących zmiennych zwracanych przez mail monitor:
    - *@Email_MonitorCode*: kod instancji mail monitora
    - *@Email_MonitorName*: nazwa instancji mail monitora
    - *@Email_ObservedFolder*: nazwa obserwowanego folderu w skrzynce e-mail
    - *@Email_Sender*: nadawca wiadomości
    - *@Email_Subject* - temat wiadomości
    - *@Email_Content*: treść wiadomości
    - *@Email_HTMLContent*: zawartość wiadomości w formacie HTML
    - *@Email_ReceivedDate*: data otrzymania wiadomości
    - *@Email_ToRecipients*: adresaci
    - *@Email_CcRecipients*: adresaci z pól DW (CC)
- `Dodaj plik jako załącznik`: powoduje dodanie pliku załączonego do wiadomości e-mail jako załącznika do utworzonej instancji dokumentu biznesowego.
- `Kategoria załączników`: kategoria, do której ma zostać dodany załącznik; domyślnie `Domyślna kategoria`.
- `Typ zapisu`: tryb dodawania załącznika do dokumentu biznesowego; dostępne wartości:
    - `Zawsze jako nowy plik lub nowa wersja`: jest to jedyne obsługiwane ustawienie, ponieważ obecnie mail monitor obsługuje tylko utworzenie nowego dokumentu biznesowego;
    - `Zawsze jako update poprzedniego pliku lub wersji`: (nieobsługiwane) jeśli dokument ma już załącznik o takiej samej nazwie, nadpisuje go lub nadpisuje wersję, jeśli jest włączona kontrola wersji.
    - `Blokada podczas zapisu`: (nieobsługiwane) jeśli dokument ma już załącznik o takiej samej nazwie, nowy załącznik nie zostanie zapisany.

Przycisk `Usuń akcję` pozwala usunąć zdefiniowaną akcję.

7.1.3. Karta Postproces

Podsumowanie:
Na karcie `Postproces` można skonfigurować zachowanie mailmonitora po przetworzeniu wiadomości e-mail i jej załączników. Dostępne opcje podzielono na dwie sekcje: po poprawnej obsłudze załącznika i w przypadku nierozpoznanego formatu pliku.

Na karcie `Postproces` są dostępne następujące ustawienia:

Poprawne przetworzenie wiadomości

- `Obsługa wiadomości`: pozwala określić zachowanie mail monitora w przypadku pomyślnego przetworzenia wiadomości; dostępne wartości:
  - `Przenieś do folderu`: przenosi przetworzoną wiadomość do podanego folderu w skrzynce odbiorczej; po włączeniu należy określić nazwę folderu w polu `Folder docelowy` poniżej.
  - `Usuń`: usuwa przetworzoną wiadomość e-mail ze skrzynki odbiorczej (jeśli włączono opcję przenoszenia, to ustawienie jest ignorowane).
  - `Brak`: nie wykonuje żadnej czynności.
- `Dodaj treść wiadomości jako załącznik (.txt)`: dodaje do utworzonego dokumentu biznesowego treść wiadomości e-mail (tylko zawartość sekcji *Content-Type: text/plain*) jako załącznik w formacie *txt*; jeśli ta sekcja będzie pusta, plik załącznika nie zostanie dodany i zostanie zalogowana odpowiednia informacja.
- `Dodaj treść wiadomości jako załącznik (.eml)`: dodaje do utworzonego dokumentu biznesowego całą wiadomość e-mail jako załącznik w formacie *eml*.
- `Dodaj treść wiadomości jako załącznik (.html)`: dodaje do utworzonego dokumentu biznesowego treść wiadomości e-mail (tylko zawartość sekcji *Content-Type: html/text*) jako załącznik w formacie *html*; jeśli ta sekcja będzie pusta, plik załącznika nie zostanie dodany i zostanie zalogowana odpowiednia informacja.
- `Wykonaj akcję po dodaniu załącznika`: powoduje wykonanie akcji wybranej z listy wyboru po dodaniu załącznika z e-maila do dokumentu biznesowego; w przypadku akcji SQL w zapytaniu można użyć następujących zmiennych:
  - *{@Email_Content}*: treść wiadomości e-mail (sekcja *Content-Type: text/plain*).
  - *{@Email_HTMLContent}*: treść wiadomości e-mail (sekcja *Content-Type: text/html*).
  - *{@Email_Sender}*: adres e-mail nadawcy wiadomości.
  - *{@Email_Subject}*: temat wiadomości.
  - *{@Email_ReceivedDate}*: data otrzymania wiadomości.

- *{@Email_ToRecipients}*: adresy e-mail odbiorców wiadomości.
- *{@Email_CcRecipients}*: adresy e-mail odbiorców wiadomości z pól CC i BC.

Kliknięcie przycisku `Przykład SQL` powoduje wyświetlenie okienka z przykładem zapytania korzystającego z tych zmiennych.

> **UWAGA**
> Na liście są dostępne tylko niektóre rodzaje akcji, w tym akcje algorytmu C#.

- `Przenieś do folderu`: przenosi przetworzoną wiadomość do podanego folderu w skrzynce odbiorczej; po włączeniu należy określić nazwę folderu w polu `Nazwa folderu` poniżej.

- `Dodaj treść wiadomości jako załącznik (.txt)`: dodaje do utworzonego dokumentu biznesowego treść wiadomości e-mail (tylko zawartość sekcji *Content-Type: text/plain*) jako załącznik w formacie *txt*; jeśli ta sekcja będzie pusta, plik załącznika nie zostanie dodany i zostanie zalogowana odpowiednia informacja.
- `Dodaj treść wiadomości jako załącznik (.eml)`: dodaje do utworzonego dokumentu biznesowego całą wiadomość e-mail jako załącznik w formacie *eml*.
- `Dodaj treść wiadomości jako załącznik (.html)`: dodaje do utworzonego dokumentu biznesowego treść wiadomości e-mail (tylko zawartość sekcji *Content-Type: html/text*) jako załącznik w formacie *html*; jeśli ta sekcja będzie pusta, plik załącznika nie zostanie dodany i zostanie zalogowana odpowiednia informacja.
- `Wykonaj akcję po dodaniu załącznika`: powoduje wykonanie akcji wybranej z listy wyboru po dodaniu załącznika z e-maila do dokumentu biznesowego; w przypadku akcji SQL w zapytaniu można użyć następujących zmiennych:
  - *{@Email_Content}*: treść wiadomości e-mail (sekcja *Content-Type: text/plain*).
  - *{@Email_HTMLContent}*: treść wiadomości e-mail (sekcja *Content-Type: text/html*).
  - *{@Email_Sender}*: adres e-mail nadawcy wiadomości.
  - *{@Email_Subject}*: temat wiadomości.
  - *{@Email_ReceivedDate}*: data otrzymania wiadomości.
  - *{@Email_ToRecipients}*: adresy e-mail odbiorców wiadomości.
  - *{@Email_CcRecipients}*: adresy e-mail odbiorców wiadomości z pól CC i BC.

Kliknięcie przycisku `Przykład SQL` powoduje wyświetlenie okienka z przykładem zapytania korzystającego z tych zmiennych.

> **UWAGA**
> Na liście są dostępne tylko niektóre rodzaje akcji, w tym akcje algorytmu C#.

- `Usuń`: usuwa przetworzoną wiadomość e-mail ze skrzynki odbiorczej (jeśli włączono opcję przenoszenia, to ustawienie jest ignorowane).

Dodatkowo można skonfigurować zachowanie mail monitora, jeśli format pliku nie zostanie rozpoznany; dostępne parametry:

`Przenieś do folderu`: przenosi nierozpoznany plik do podanego folderu w skrzynce odbiorczej; po włączeniu należy określić nazwę folderu w polu `Nazwa folderu` poniżej.

`Usuń`: usuwa nierozpoznany plik (jeśli włączono opcję przenoszenia, to ustawienie jest ignorowane).

Obsługa błędów przetwarzanie wiadomości

- `Obsługa wiadomości`: pozwala określić zachowanie mail monitora w przypadku, gdy format pliku nie zostanie rozpoznany; dostępne wartości:
- `Przenieś do folderu`: przenosi wiadomość do podanego folderu w skrzynce odbiorczej; po włączeniu należy określić nazwę folderu w polu `Folder docelowy` poniżej.
- `Usuń`: usuwa wiadomość e-mail ze skrzynki odbiorczej (jeśli włączono opcję przenoszenia, to ustawienie jest ignorowane).
- `Brak`: nie wykonuje żadnej czynności.

7.2. ZADANIA CYKLICZNE

Podsumowanie:
Funkcja zadań cyklicznych służy do okresowego wykonywania akcji zgodnie ze zdefiniowanym wyrażeniem Cron. W zadaniach cyklicznych można używać akcji następujących typów:

- SQL do bazy systemowej,
- Wysyłka e-mail,
- Zapytania do Webserwisów,
- Algorytm C#,
- Importowanie dokumentów.

☛ *AdminSPA* > `INTEGRACJE` > `Zadania cykliczne` > `Lista...`/`Historia...`

DEFINIOWANIE ZADANIA CYKLICZNEGO

Aby dodać nowe zadanie cykliczne, kliknij kolejno `INTEGRACJE` > `Zadania cykliczne` > `Lista zadań cyklicznych`, a następnie kliknij przycisk `Nowe zadanie` w lewym górnym rogu. Zostanie wyświetlone okno dialogowe `Nowe zadanie cykliczne`.

W tym oknie określ następujące ustawienia:

- `Aplikacja`: nazwa aplikacji, do której zostanie przypisany tworzony obiekt; wszystkie obiekty przypisane do aplikacji można wraz z nią wyeksportować, używając polecenia `Migrator aplikacji` w sekcji `ADMINISTRACJA` menu głównego.
- `Moduł`: nazwa modułu aplikacji, do którego zostanie przypisany tworzony obiekt.
- `Kod`: kod zadania cyklicznego.
- `Nazwa`: nazwa zadania cyklicznego.
- `Aktywne`: określa, czy dla zadania ma się rozpocząć odliczanie czasu do wykonania zadania zgodnie z wyrażeniem *cron*.
- `Cron`: wyrażenie definiujące interwał wykonania zadania cyklicznego; szczegóły poniżej.
- `Typ akcji`: wybierz typ akcji; dostępne wartości:
  - `Akcja SQL`
  - `Akcja Email`
  - `Akcja WebService`
  - `Akcja Algorytm C#`
  - `Akcja importu dokumentów`
- `Wybrana akcja`: wybierz akcję do wykonania w ramach zadania cyklicznego; dostępne są tylko akcje globalne.

Dla zdefiniowanych zadań cyklicznych rejestrowana jest wspólna historia wykonania (`INTEGRACJE` > `Zadania cykliczne` > `Historia wykonania zadań`). Utworzenie nowego aktywnego zadania oraz zmiana stanu z nieaktywnego na aktywne powoduje utworzenie w historii wpisu ze statusem *Dodano do harmonogramu* (rozpoczęcie odliczania interwału cron). Zmiana stanu z aktywnego na nieaktywne, a także usunięcie zadania skutkuje wpisem *Usunięto z harmonogramu*. W przypadku aktualizacja zadania odliczanie interwału cron jest resetowane (wpis o usunięciu, a następnie dodaniu). Ponadto rejestrowane są statusy *Rozpoczęto* (rozpoczęcie wykonywania zadania), *Zakończono* (zakończenie wykonywania zadania) oraz *Błąd*. W tym ostatnim przypadku rejestrowane są szczegóły wystąpienia błędu.

DEFINIOWANIE HARMONOGRAMU

Harmonogramy definiuje się w formie specjalnych wyrażeń składających się z siedmiu pól, które rozdziela się znakiem spacji i które reprezentują (od lewej):

- sekundy
- minuty
- godziny
- dzień miesiąca
- miesiąca
- dzień tygodnia
- rok (opcjonalnie)

Na przykład wyrażenie `0 0 12 ? * WED` oznacza „w każdą środę o godzinie 12.”.

W poszczególnych polach można wpisywać:

- pojedyncze wartości, np. `WED`, jak w powyższym przykładzie)
- listy wartości, tj. kilka wartości oddzielonych przecinkami, np. `"MON, WED, SAT"` („w poniedziałek, środę i sobotę”)
- zakresy wartości, np. `"MON-WED"` („od poniedziałku do środy”)

Możliwe jest także używanie zakresów na listach, np. `"MON-WED, SAT"` („od poniedziałku do środy i w soboty”).

Znak wieloznaczny `*` oznacza dowolną możliwą wartość dla danego pola. Gwiazdka w polu miesiąca oznacza „w każdym miesiącu”, a w polu dnia tygodnia oznacza „w każdym dniu tygodnia”.

Wszystkie pola mają zestaw obsługiwanych wartości. Dla sekund i minut są to liczby od 0 do 59 dla sekund i minut, dla godzin liczby od 0 do 23 dla godzin. W polu dnia miesiąca można wpisać dowolną liczbę od 0 do 31 (pamiętając o liczbie dni w danym miesiącu). Miesiące można określać liczbowo od 0 do 11 lub skrótami tekstowymi JAN, FEB, MAR, APR, MAY, JUN, JUL, AUG, SEP, OCT, NOV i DEC. Dni tygodnia można określać liczbami od 1 do 7 (1 oznacza niedzielę) lub skrótami SUN, MON, TUE, WED, THU, FRI i SAT.

Znak `/` oznacza przyrosty wartości. Na przykład wpisanie `0/15` w polu minut będzie oznaczało „co kwadrans od minuty 0”, natomiast wartość `3/20` w tym polu oznacza „co dwadzieścia minut, poczynając od 3. minuty” (to samo, co wpisanie `3, 23, 43`).

W pola dnia miesiąca i dnia tygodnia można używać pytajnika (`?`), który oznacza „brak określonej wartości”. Przydaje się to w przypadku konfigurowania harmonogramu z użyciem tylko niektórych pól. W tych polach można także używać litery `L`, która oznacza ostatni (ang. Last) dzień miesiąca (31 stycznia, ale 28 lutego) lub ostatni dzień tygodnia (`7` lub

`SAT`). Jednak użycie tej litery wraz z określeniem dnia tygodnia, np. `FRIL` (lub `6L`), będzie oznaczało ostatni taki dzień tygodnia w miesiącu, czyli w tym przykładzie „ostatni piątek miesiąca”. Z opcją `L` nie można używać zakresów ani list.

Litera `W` oznacza dzień roboczy (ang. Weekday) przypadający najbliżej podanego dnia miesiąca. Na przykład `15W` oznacza „najbliższy dzień roboczy po 15 dniu miesiąca”.

Symbol `#` jest używany w polu dnia tygodnia służy do określania kolejnego dnia tygodnia w miesiącu. Na przykład `FRI#3` (lub `6#3`) oznacza „trzeci piątek miesiąca”.

> **WSKAZÓWKA**
> Jako pomocy w konstrukcji wyrażeń cron można użyć generatora online: https://www.freeformatter.com/cron-expression-generator-quartz.html.

Przykłady

- `0 0/5 * * * ?`: co pięć minut
- `10 0/5 * * * ?`: co pięć minut, zaczynając od 10 sekundy po każdej pełnej godzinie (np. 10:00:10, 10:05:10 itp.)
- `0 30 10-13 ? * WED,FRI`: o godzinie 10:30, 11:30, 12:30 i 13:30 w każdą środę i piątek.
- `0 0/30 8-9 5,20 * ?`: co pół godziny między 8:00 i 10:00 w każdy 5. i 20. dzień miesiąca; NIE obejmuje godziny 10:30, tylko 8:00, 8:30, 9:00 i 9:30.

KONFIGURACJA MODUŁU QUARTZ

Za uruchamianie zadań cyklicznych odpowiada moduł Quartz w serwisie *TaskService*. W pliku *appsettings.json* serwisu TaskService znajduje się sekcja *Quartz* (domyślne wpisy przedstawiono poniżej), w której można dopisywać odpowiednie ustawienia konfiguracyjne. Dokumentację dostępnych opcji konfiguracji można znaleźć pod adresem https://www.quartz-scheduler.net/documentation/quartz-3.x/configuration/reference.html.

```
"Quartz": {
  "quartz.threadPool.threadCount": "10"
}
```

To ustawienie określa liczbę wątków w puli, do której wysyłane są instrukcje uruchamiania zadań cyklicznych. W przypadku, kiedy przydzielona liczba będzie zbyt mała na obsłużenie zaplanowanych zadań, może dojść do sytuacji, że indywidualna iteracja zadania cyklicznego nie zostanie wykonana. Moduł Quartz obsługuje tę sytuacją w taki sposób, że po upływie

zadanego czasu (ustawienie *quartz.jobStore.misfireThreshold*, domyślnie 60 000 ms) ponowi próbę wykonania takiego zadania.

8. ADMINISTRACJA

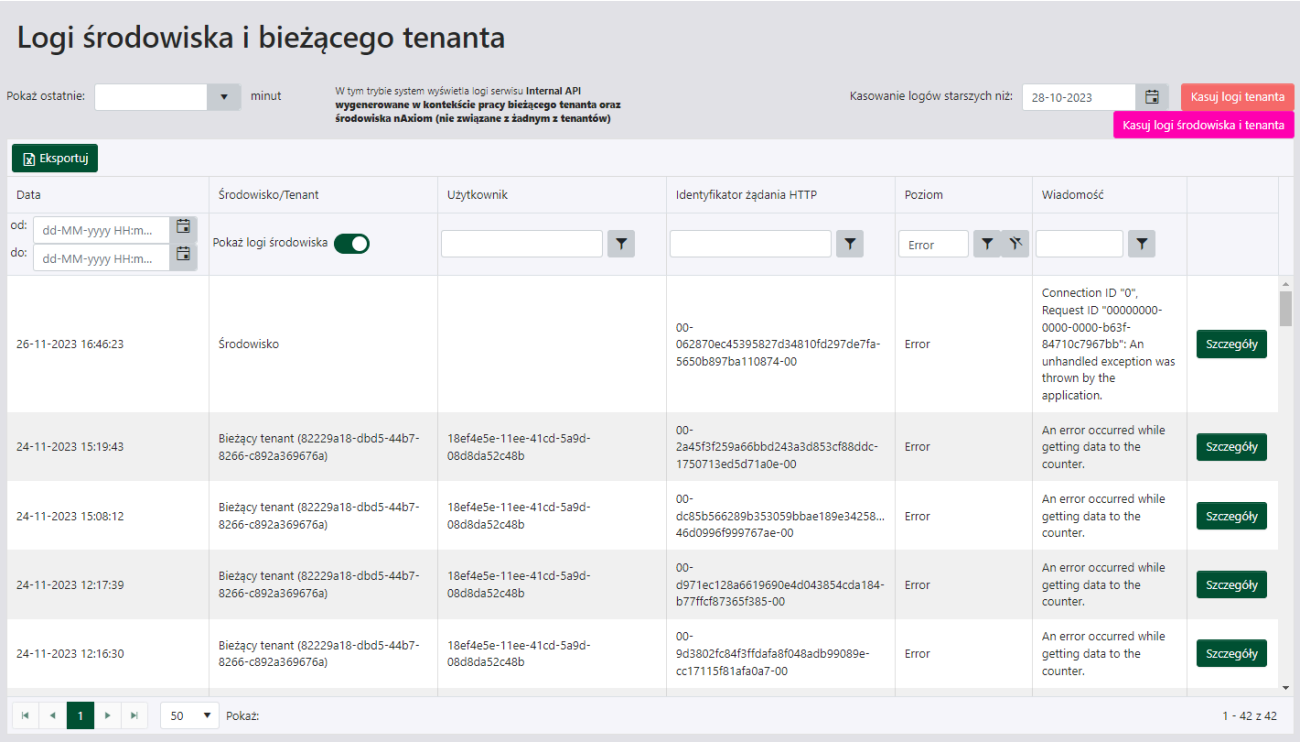
> ☛ *AdminSPA* > `ADMINISTRACJA`

W tej części opisano polecenia menu `ADMINISTRACJA`.

8.1. LOGI ŚRODOWISKA I BIEŻĄCEGO TENANTA

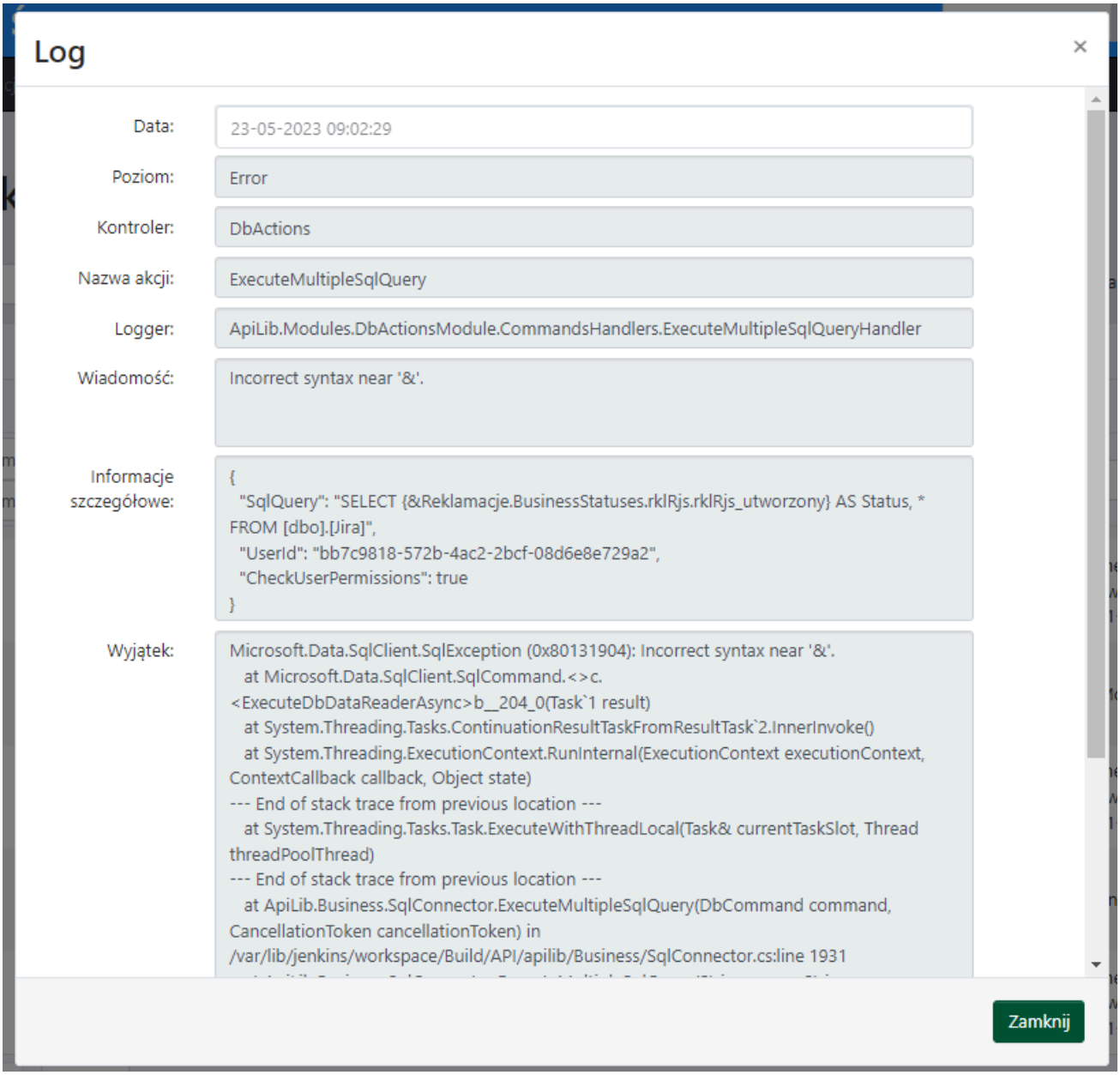
> **Podsumowanie:**
> Logi środowiska i bieżącego tenanta to lista komunikatów z serwisu *api* zarejestrowanych podczas pracy użytkowników w witrynie nAxiom. Wpisy w logach mogą być oznaczone identyfikatorem tenanta lub nie (tzw. logi środowiska). Użytkownicy widzą w logach tylko wpisy dotyczące swojego tenanta oraz logi środowiska.

> ☛ *AdminSPA* > `ADMINISTRACJA` > `Logi środowiska i bieżącego tenanta`



Logi środowiska i bieżącego tenanta

Każdy wpis na liście ma z prawej strony przycisk `Szczegóły`, który wyświetla okno dialogowe ze szczegółowymi informacjami o błędzie lub zwróconym wyjątku. W ustawieniach systemu znajduje się przełącznik `Ukryj wartości parametrów w logowanym zapytaniu SQL`. Jego wyłączenie powoduje wyświetlenie w szczegółach błędu parametrów zapytania SQL, dla którego wystąpił błąd. Na podstawie tych informacji można debugować tworzoną aplikację.



Szczegóły błędu

Wpisy na liście można filtrować według wartości w poszczególnych kolumnach. Lista `Pokaż ostatnie` zawiera kilka interwałów w minutach, które także ograniczają liczbę wpisów wyświetlanych na liści. Przycisk `Eksportuj` zapisuje wpisy (wszystkie, niezależnie od filtrów) w pliku programu Excel.

U góry listy znajduje się pole daty i przyciski `Kasuj logi tenanta` oraz `Kasuj logi tenanta i środowiska`, które pozwalają usunąć wpisy sprzed ustawionej daty. Regularne usuwanie wpisów poprawia wydajność platformy nAxiom.

Poziom istotności rejestrowanych komunikatów określa administrator tenantów w aplikacji *TenantsAdminSPA* w ustawieniach globalnych (*TenantsAdminSPA* > `Global settings` > `Error logging level`).

Powiązane tematy:

- Diagnostyka w nAxiom
- Ustawienia systemu, sekcja Logi

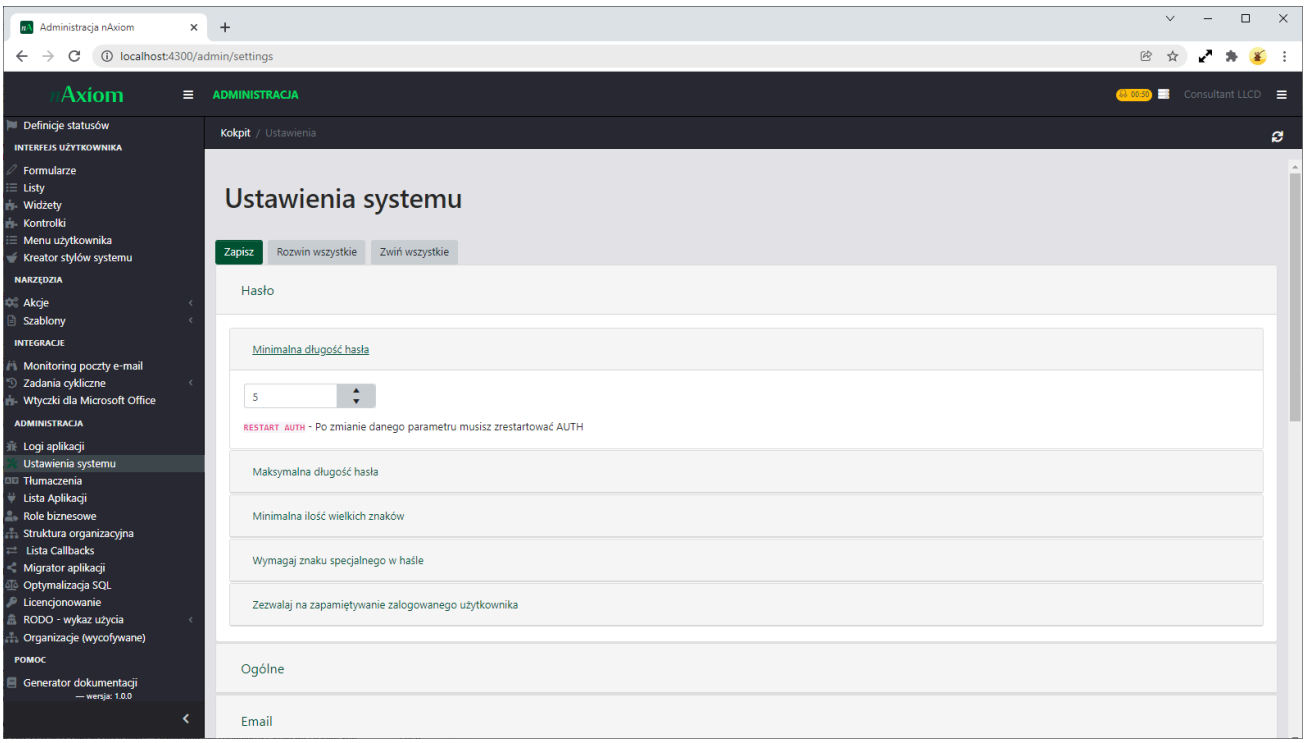
8.2. USTAWIENIA SYSTEMU

Podsumowanie:
Ustawienia systemowe dotyczą różnych aspektów tworzenia i działania aplikacji budowanych w nAxiom. Są to ustawienia, które obowiązują dla wszystkich aplikacji tworzonych i używanych w danej witrynie. Ustawienia nie są migrowane wraz z aplikacjami z witryny do witryny.

☛ *AdminSPA* > `ADMINISTRACJA` > `Ustawienia systemu`

W TYM TEMACIE

- Hasło
- Ogólne
- Email
- Załączniki
- Logi
- Notyfikacje
- LDAP
- Struktura organizacyjna
-
- Profile użytkowników
- Breadcrumbs
- Kreator modeli danych
- Polityka resetowania hasła
- ACL
- Zarządzanie wartościami parametrów
- Czas pracy użytkownika
- eSign
- Audyt
- Pomoc i dokumentacja
- Procesy biznesowe
- Widok Moje zadania
- OCR
- Tryb pojedynczej aplikacji
- Zastępstwa
- Praca na urządzeniach mobilnych
- Import użytkowników
- Cache



Ustawienia systemu

Poszczególne ustawienia znajdują się w zwiniętych sekcjach, Przyciski `Rozwiń wszystkie` i `Zwiń wszystkie` u góry pozwalają odpowiednio rozwinąć wszystkie sekcje ustawień i zwinąć je.

Po zmianie wartości ustawienia należy kliknąć przycisk `Zapisz`, aby zapisać zmianę. W niektórych wypadkach zmiana zaczyna obowiązywać dopiero po ponownym uruchomieniu witryny nAxiom lub jej komponentu.

W witrynie nAxiom można skonfigurować następujące ustawienia systemowe:

HASŁO

> **UWAGA**
> Po zmianie każdego z ustawień w tej sekcji należy ponownie uruchomić aplikację *Auth*.

- `Minimalna długość hasła`: wartość domyślna to 5 znaków.
- `Maksymalna długość hasła`: wartość domyślna to 25 znaków.
- `Minimalna ilość wielkich znaków`: wartość domyślna to 1 znak.
- `Wymagaj znaku specjalnego w haśle`: domyślnie włączone.
- `Zezwalaj na zapamiętanie zalogowanego użytkownika`: domyślnie włączone.

OGÓLNE

- `Ważność linku do aktywacji konta (w godzinach)`: wartość domyślna to 24.
- `Własny CSS`: pozwala wprowadzić własną definicję stylu CSS; odpowiednie selektory można ustalić, korzystając z inspekcji w przeglądarce; poniższy przykład pozwala zmienić kolor panelu bocznego we front office:

```
nax-aside-menu-desktop { background-color:lightgrey }
```

- `Pokaż eksperymentalne funkcjonalności`: domyślnie wyłączone; po włączeniu wyświetla pozycje menu oznaczone jako funkcjonalności eksperymentalne.
- `Pomiń błędy importu`: domyślnie wyłączone; po włączeniu ignoruje błędy podczas importu aplikacji (`ADMINISTRACJA` > `Migrator aplikacji`).
- `Włącz wyszukiwarkę`: włącza wyszukiwanie pełnotekstowe w dokumentach.
- `System – Repozytorium systemowe dla zapisu głównych plików aplikacji, np. logo`: pozwala wybrać jedno z domyślnych repozytoriów (`DANE` > `Repozytoria załączników`)
- `Rozmiar w procentach pola z notyfikacjami w popupie`: określa w procentach wysokość panelu powiadomień (wyświetlanego we front-office po kliknięciu ikony dzwonka) względem wysokości okna przeglądarki. Każdy użytkownik może indywidualnie zmienić to ustawienie w profilu (`Nazwa użytkownika` > `Mój profil` > `Procentowy rozmiar panelu notyfikacji`).

EMAIL

> **UWAGA**
> Zmiana tego ustawienia wymaga restartowania komponentu *TaskService* (należy restartować odpowiednią pulę aplikacji w *Menedżerze internetowych usług informacyjnych*).

- `Repozytorium zapisu plików dla szablonów e-mail`: pozwala wybrać jedno z repozytoriów (`DANE` > `Repozytoria załączników`) na załączniki używane w szablonach e-mail.
- `Exchange – Repozytorium zapisu plików dla załączników pobieranych z wiadomości`: pozwala wybrać jedno z repozytoriów (`DANE` > `Repozytoria załączników`)
- `Typ serwera pocztowego`: wartość domyślna `SMTP`; alternatywnie dostępna wartość `Exchange`.
- `Exchange – Konto użytkownika`: nazwa konta użytkownika na serwerze Exchange.

- `Exchange – Hasło użytkownika`: hasło użytkownika na serwerze Exchange.
- `Exchange – cykliczne pobieranie e-mail`: domyślnie wyłączone.
- `Nazwa hosta smtp`: nazwa hosta protokołu SMTP, np. *smtp.office365.com*.
- `Port hosta smtp`: numer portu hosta protokołu SMTP, domyślnie *587*.
- `Nazwa użytkownika powiązana z poświadczeniami`: nazwa użytkownika skrzynki pocztowej.
- `Hasło dla nazwy użytkownika powiązanej z poświadczeniami`: hasło do skrzynki pocztowej.
- `Czy klient Smtp używa SSL do szyfrowania połączenia`: domyślnie wyłączone.
- `Interwał przetwarzania wysyłania maili w sekundach`: wartość domyślna to 60.

- `Repozytorium zapisu plików dla szablonów e-mail`: pozwala wybrać jedno z repozytoriów (`DANE` > `Repozytoria załączników`) na załączniki używane w szablonach e-mail.
- `Exchange – Repozytorium zapisu plików dla załączników pobieranych z wiadomości`: pozwala wybrać jedno z repozytoriów (`DANE` > `Repozytoria załączników`)
- `Typ serwera pocztowego`: dostępne wartości:
  - `SMTP (Basic Authentication)`: wymagane parametry:
    - `Nazwa hosta smtp`: np *smtp.office365.com*
    - `Port hosta smtp`: domyślnie *587*
    - `Nazwa użytkownika powiązana z poświadczeniami`: np. *imie.nazwisko@firma.pl*
    - `Hasło dla nazwy użytkownika powiązanej z poświadczeniami`: np. *P@ssw0rd*
    - `Czy klient Smtp używa SSL do szyfrowania połączenia`: włączone (domyślnie) lub wyłączone
  - `Exchange`: wymagane ustawienie parametrów zaczynających się od *Exchange...*
  - `SMTP Microsoft Entra Client Credentials Authentication`: wymagane parametry:
    - `Tenant ID dla uwierzytelniania OAuth2 w SMTP`: np.*xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx*
    - `Client ID dla uwierzytelniania OAuth2 w SMTP`: np. *yyyyyyyy-yyyy-yyyy-yyyy-yyyyyyyyyyyy*

        - `Client Secret dla uwierzytelniania OAuth2 w SMTP`: np. ********
        - `Scope dla uwierzytelniania OAuth2 w SMTP`: np. *https://outlook.office365.com/.default*
        - `Adres URL autoryzacji dla uwierzytelniania OAuth2 w SMTP`: np. *https://login.microsoftonline.com/{TenantId}*
        - `Nazwa użytkownika powiązana z poświadczeniami`: np. *imie.nazwisko@firma.pl*
        - `Hasło dla nazwy użytkownika powiązanej z poświadczeniami`: np. *P@ssw0rd*
        - `Czy klient Smtp używa SSL do szyfrowania połączenia`: włączone (domyślnie) lub wyłączone
    - `SMTP Microsoft Entra Client Certificate Authentication`: wymagane podanie parametrów jak wyżej, tylko zamiast *Client Secret* używany jest certyfikat:
        - `Certyfikat klienta dla uwierzytelniania OAuth2 w SMTP`
        - `Hasło do certyfikatu klienta dla uwierzytelniania OAuth2 w SMTP`
- `Exchange – Konto użytkownika`: nazwa konta użytkownika na serwerze Exchange.
- `Exchange – cykliczne pobieranie e-mail`: domyślnie wyłączone.
- `Exchange – Hasło użytkownika`: hasło użytkownika na serwerze Exchange.
- `Interwał powtarzania wysyłanie maili w sekundach`: interwał (w sekundach) z jakim *taskservice* sprawdza, czy są wiadomości e-mail do wysłania.
- `Adres URL autoryzacji dla uwierzytelniania OAuth2 w SMTP`: adres serwera logowania Microsoft, zwykle *https://login.microsoftonline.com/{TenantId}*
- `Certyfikat klienta dla uwierzytelniania OAuth2 w SMTP`: certyfikat klienta w formacie PFX zakodowany jako Base64
- `Hasło do certyfikatu klienta dla uwierzytelniania OAuth2 w SMTP`: hasło podane podczas tworzenia pliku PFX
- `Client ID dla uwierzytelniania OAuth2 w SMTP`: identyfikator aplikacji zarejestrowanej w Entra ID
- `Scope dla uwierzytelniania OAuth2 w SMTP`: zakres uprawnień żądanych przez aplikację kliencką
- `Client Secret dla uwierzytelniania OAuth2 w SMTP`: hasło do aplikacji klienckiej (*Client Id*)
- `Czy klient Smtp używa SSL do szyfrowania połączenia`: domyślnie

wyłączone.

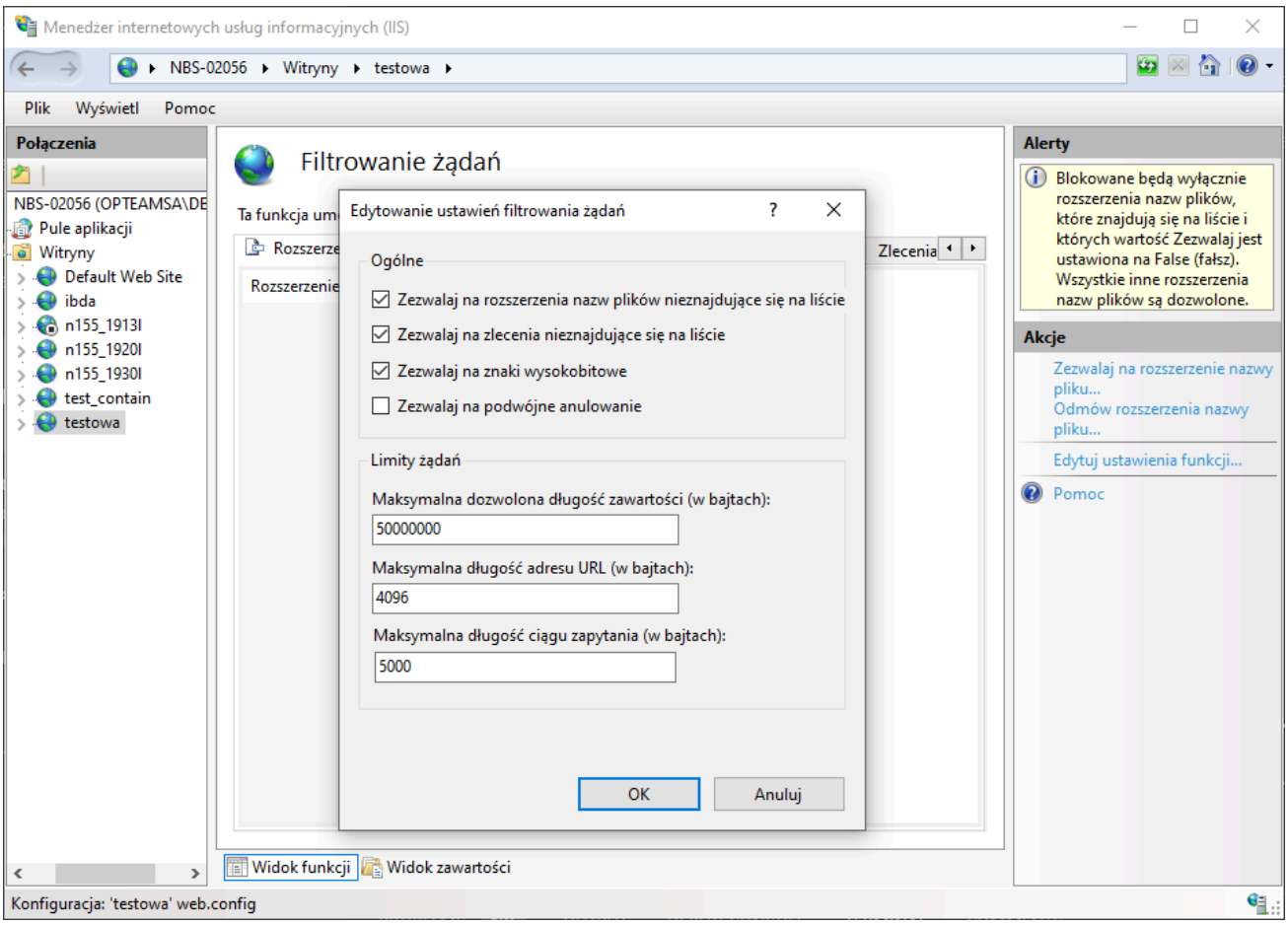
- `Nazwa hosta smtp`: nazwa hosta protokołu SMTP, np. *smtp.office365.com*.
- `Port hosta smtp`: numer portu hosta protokołu SMTP, domyślnie *587*.
- `Tenant ID dla uwierzytelniania OAuth2 w SMTP`:identyfikator dzierżawy Microsoft Entra ID (organizacja)
- `Nazwa użytkownika powiązana z poświadczeniami`: nazwa użytkownika skrzynki pocztowej.
- `Hasło dla nazwy użytkownika powiązanej z poświadczeniami`: hasło do skrzynki pocztowej.

ZAŁĄCZNIKI

- `Dozwolone typy plików`: (tylko do odczytu) lista rozszerzeń plików dozwolonych jako załączniki. Dozwolone formaty plików konfiguruje się w aplikacji TenantsAdmin SPA .
- `Maksymalny rozmiar pliku`: maksymalny rozmiar pliku załącznika; wartość domyślna dla nowej witryny to 51200 MB; wartość *0* oznacza brak limitu; pierwszeństwo ma limit określony w kategorii załączników, jednak najwyższy priorytet ma konfiguracja serwera WWW, patrz informacje poniżej.

WSKAZÓWKA

Maksymalny rozmiar załącznika w nAxiom zależy od parametru `Maksymalna dozwolona długość zawartości (w bajtach)` dostępnego w programie *Menedżer internetowych usług informacyjnych (IIS)*. Aby skonfigurować ten parametr, należy kliknąć dwukrotnie funkcję *Filtrowanie żądań*, a następnie kliknąć polecenie `Edytuj ustawienia funkcji` w menu z prawej strony. W wyświetlonym oknie dialogowym należy wpisać żądaną wartość parametru. Parametr można zdefiniować dla wszystkich witryn na poziomie serwera (hosta) lub indywidualnie dla każdej witryny.



Ustawianie maks. wielkości załącznika w Menedżerze IIS

LOGI

- `Czas wykonywania zapytania w milisekundach`: limit czasu zapytania; wartość domyślna to 1500.
- `Ukryj wartości parametrów w logowanym zapytaniu SQL`: (domyślnie włączone) po wyłączeniu logi systemowe (`ADMINISTRACJA` > `Logi systemowe`) zawierają wartości parametrów zapytań SQL, których dotyczą wpisy.

NOTYFIKACJE

> **UWAGA**
> W przypadku ustawień dotyczących wyświetlania powiadomień wpisanie wartości 0 spowoduje, że okienko powiadomienia będzie wyświetlane do momentu kliknięcia go.

- `Czas pokazywania pomyślnej notyfikacji w sekundach`: czas wyświetlania okna komunikatu o pozytywnym wyniku operacji; wartość domyślna to 10.
- `Czas pokazywania błędnej notyfikacji w sekundach`: czas wyświetlania okna komunikatu o niepowodzeniu operacji; wartość domyślna to 10.

- `Czas pokazywania ostrzegającej notyfikacji w sekundach`: czas wyświetlania okna komunikatu z ostrzeżeniem; wartość domyślna to 10.
- `Czas pokazywania zwykłej notyfikacji w sekundach`: czas wyświetlania okna komunikatu informacyjnego; wartość domyślna to 10.
- `Pozycja notyfikacji`: położenie okienka komunikatu na stronie; dostępne wartości:
    - `TopRight`: prawy górny róg.
    - `TopCenter`: środek u góry.
    - `TopLeft`: lewy górny róg.
    - `BottomRight`: prawy dolny róg.
    - `BottomCenter`: środek u dołu.
    - `BottomLeft`: lewy dolny róg.
    - `ApplicationDefault`: domyślne ustawienie aplikacji.
- `Adres email do wysyłki powiadomień o błędach od użytkowników`: adres e-mail używany do wysyłania z panelu powiadomień i alertów informacji o błędach występujących podczas pracy użytkownika. E-maile do wysyłki są zapisywane w tabeli *core.Emails*, a następnie wysyłane przez *taskservice*. Patrz Powiadomienia i alerty .

LDAP

> **UWAGA**
> Zmiana tych ustawień wymaga restartowania komponentu *Auth* (należy restartować odpowiednią pulę aplikacji w *Menedżerze internetowych usług informacyjnych* lub odpowiedni pod w klastrze Kubernetes).

- `Ścieżka do serwera LDAP (LDAP path)`: ścieżka do serwera LDAP bez przedrostka *LDAP://* (*LDAPS://*).
- `Nazwa użytkownika`: nazwa użytkownika z dostępem do serwera LDAP.
- `Hasło użytkownika`: hasło użytkownika LDAP.
- `Domena LDAP`: nazwa domeny LDAP.
- `Nazwa DNS lub adres IP serwera LDAP`: nazwa hosta lub adres IP serwera LDAP.
- `Port do serwera LDAP`: wartość domyślna 389 dla połączeń nieszyfrowanych.
- `Połączenie przez SSL`: domyślnie wyłączone.
- `Base DN do serwera LDAP`: nazwa wyróżniająca określająca miejsca rozpoczęcia wyszukiwania na serwerze LDAP.

- `Typ połączenia LDAP`: dostępne wartości *ActiveDirectory*, *OpenLDAP* i *NovellActiveDirectory* i *OpenLDAP*.
- `Nazwa identyfikatora użytkownika`: *sAMAccountName* dla połączeń *ActiveDirectory* lub *uid* dla *OpenLDAP*.

W celu weryfikacji parametrów połączenia należy włączyć uwierzytelnianie na serwerze LDAP (w *TenantsAdminSPA*, `Edit tenant` > `Configuration` > `Settings`), utworzyć profil użytkownika nAxiom z włączonym przełącznikiem `Użytkownik LDAP` i sprawdzić, czy logowanie z użyciem tego profilu przebiega prawidłowo.

STRUKTURA ORGANIZACYJNA

- `ID dokumentu biznesowego`: identyfikator dokumentu biznesowego wygenerowanego na potrzeby struktury organizacyjnej; można znaleźć w tabeli *core.BusinessDocuments* w bazie danych SQL Server.
- `ID definicji formularza`: identyfikator formularza, który zostanie użyty jako karta `Dane dodatkowe` w oknie dialogowym `Edycja jednostki organizacyjnej` (patrz Struktura organizacyjna)
- `Włącz własne ustawienia listy użytkowników`: domyślnie wyłączone; po włączeniu umożliwia zdefiniowanie dwóch indywidualnych zapytań dla list użytkowników wyświetlanych na formularzu struktury organizacyjnej.
- `Własne zapytanie dla dostępnych użytkowników`: zapytanie źródłowe dla listy użytkowników wyświetlanej po kliknięciu przycisku `Dodaj` na karcie `Użytkownicy` na formularzu `Edycja jednostki organizacyjnej`.
- `Własne zapytanie dla użytkowników w jednostce organizacyjnej`: zapytanie źródłowe dla listy użytkowników wyświetlanej na karcie `Użytkownicy` na formularzu `Edycja jednostki organizacyjnej`.
- `Wyświetlanie ścieżki jednostek w profilu uzytkownika`: określa sposób wyświetlania ścieżki do jednostki organizacyjnej w sekcji `Role biznesowe w jednostkach organizacyjnych` w profilu użytkownika; dostępne wartości:
  - `Używaj nazw jednostek`
  - `Używaj kodów jednostek`
- `Tooltip dla ścieżki jednostek w profilu uzytkownika`: określa zawartość podpowiedzi wyświetlanej po wskazaniu myszą pola `Jednostka organizacyjna` w profilu użytkownika; dostępne wartości:
  - `Pokaż nazwy i kody`
  - `Pokaż tylko kody`
  - `Pokaż tylko nazwy`

    - `Nie wyświetlaj tooltipa`

PROFILE UŻYTKOWNIKÓW

- `ID dokumentu biznesowego`: identyfikator dokumentu biznesowego wygenerowanego na potrzeby danych użytkowników; można znaleźć w tabeli *core.BusinessDocuments* w bazie danych SQL Server.
- `ID definicji formularza`: identyfikator formularza, który zostanie użyty jako karta `Dane dodatkowe` w oknie dialogowym `Edycja profilu użytkownika` (patrz Profile użytkowników)

BREADCRUMBS

- `Pokazuj jako pierwszy węzeł`: pozwala określić element wyświetlany na początku ścieżki breadcrumb; dostępne wartości:
    - `Nazwa aplikacji`
    - `Opis aplikacji`
- `Jako identyfikator dokumentu wyświetlaj`: atrybut identyfikujący formularz/widok listy w ścieżce breadcrumb; dostępne wartości:
    - `Id`: identyfikator formularza/listy
    - `Code`: kod formularza/listy
    - `Name`: nazwa formularza/listy (wartość domyślna)

KREATOR MODELI DANYCH

- `Wyrazy zabronione w nazwach obiektów (rozdzielone ";")`: lista słów zastrzeżonych, których nie można używać w kreatorze modeli danych.

POLITYKA RESETOWANIA HASŁA

- `Pozwól na reset hasła przez niezalogowanych użytkowników nAxiom`: domyślnie włączone; po włączeniu umożliwia zmianę hasła np. kiedy użytkownik zapomni go.
- `Pozwól na zmianę hasła przez zalogowanych użytkowników nAxiom`: domyślnie włączone.
- `Termin (w minutach) ważności linku do resetu hasła`: czas, po upływie którego, wysłany w e-mailu link do zresetowania hasła traci ważność.

- `Wskazanie akcji email dla powiadomienia o resecie hasła`: zdefiniowana akcja e-mail, która zostanie wykonana, kiedy użytkownik podejmie próbę zresetowania hasła; ta akcja jest wykonywana także podczas zapisu profilu użytkownika bez podanego hasła.

> **UWAGA**
> Po wskazaniu tej akcji należy ponownie uruchomić witrynę (np. w programie Menedżer internetowych usług informacyjnych).

ACL

- `Czas życia wpisów ACL`: czas (w godzinach), po upływie którego nieużywane wpisy ACL są usuwane z pamięci podręcznej; domyślnie 24 godziny.

ZARZĄDZANIE WARTOŚCIAMI PARAMETRÓW

- `Wyłącz konwersję na nvarchar dla zmiennych @Nazwa`: (domyślnie wyłączone) włączenie powoduje, że wartości zmiennych *{@Nazwa}* w zapytaniach SQL są przetwarzane zgodnie z typem danych określonym w tabeli bazy danych; jeśli jest wyłączone, wartości tych zmiennych są konwertowane na tym *nvarchar* (jest to zachowanie domyślne we wcześniejszych wersjach systemu).
- `Formatuj tekst przekazywany w parametreach @Nazwa na datę`: (domyślnie wyłączony) włączenie powoduje, że wartości tekstowe, które wyglądają jak data w formacie *dd/MM/yyyy* lub *yyyy-MM-dd* będą konwertowane na typ *datetime* w formacie *dd.MM.yyyy HH:mm:ss* z godziną *00:00:00*.

CZAS PRACY UŻYTKOWNIKA

- `Czas do ostrzeżenia o nieaktywności (w minutach)`: czas braku aktywności użytkownika (mysz, klawiatura), po upływie którego zostanie wyświetlone ostrzeżenie o wylogowaniu; domyślnie 30 minut.

- `Wyświetlanie ostrzeżenia o przymusowym wylogowaniu (w sekundach)`: czas od wyświetlenia ostrzeżenia o wylogowaniu do wylogowania; wznowienie aktywności przez użytkownika w tym czasie anuluje wylogowanie; domyślnie 300 sekund.

WSKAZÓWKA

- Ustawienie wartości *0* dla obu parametrów jest równoznaczne z ustawieniem wartości domyślnej.
- Wpisanie liczby ujemnej dla pierwszego parametru spowoduje, że ostrzeżenie o wylogowaniu będzie wyświetlane po każdym poruszeniu myszą, co uniemożliwi korzystanie z witryny; przywrócenie obsługiwanej wartości wymaga edycji wartości w kolumnie *NValue* dla pozycji *IDLE_TIME* (kolumna *Code*) w tabeli *core.Appsettings*.
- Wpisanie liczby ujemnej dla drugiego parametru spowoduje, że ostrzeżenie o wylogowaniu nie będzie wyświetlane, tylko po upływie czasu nieaktywności nastąpi wylogowanie.
- Logując się, użytkownik otrzymuje token dostępowy, którego czas ważności wynosi 15 minut i który jest automatycznie odnawiany; wylogowanie powoduje unieważnienie tokenu.

ESIGN

- `Podstawowy adres API`: adres domeny, w której dostępna jest usługa *eSign*.
- `Adres URL dla autentykacji żądań`: pełny adres endpointu usługi *eSing*.
- `ClientId dostępu do API`: identyfikator użytkownika API używany do autoryzacji w usłudze *eSign*.
- `ClientSecret dostępu do API`: hasło użytkownika API używane do autoryzacji w usłudze *eSign*.

AUDYT

- `Dostępność przycisku audytu`: (domyślnie włączone) wyłączenie przełącznika spowoduje, że w obszarze przycisków systemowych formularzy w aplikacji *FrontSPA* nie będzie wyświetlany przycisk audytu (ikona zegara); to ustawienie można zmieniać indywidualnie na poziomie definicji dokumentu biznesowego, z kolei widoczność tego przycisku dla indywidualnych użytkowników wymaga posiadania odpowiedniego uprawnienia PBA (`Użytkowanie aplikacji` > `Formularze` > `Opcje systemowe` > `Dostęp do audytu dokumentu`).
- `Lista akcji wykonywanych podczas audytu`: kliknij przycisk `Wybierz` i wybierz z listy akcje, które mają zostać wykonane, kiedy użytkownik kliknie przycisk audytu danych; wybrana sekwencja akcji musi zawierać akcję otwierania formularza, która wyświetli własny formularz z danymi audytu.

POMOC I DOKUMENTACJA

- `Zbiór akcji wywoływanych po kliknięciu w przycisk pomocy w formularzach`: wybierz odpowiednie akcje; wśród nich musi znajdować się akcja otwierania formularza.
- `Zbiór akcji wywoływanych po kliknięciu w przycisk pomocy w listach`: wybierz odpowiednie akcje; wśród nich musi znajdować się akcja otwierania formularza.

PROCESY BIZNESOWE

- `Interwał powtarzania uruchamiania oczekujących procesów (sek)`: w podanych odstępach czasu komponent *TaskService* sprawdza, czy upłynął termin ustawiony w bloczkach oczekiwania na diagramie procesu; określony czas powinien być odpowiednio krótszy od najkrótszego czasu bloczku oczekiwania.
- `Częstotliwość sprawdzania listy eskalatorów do wysyłki (w minutach)`: domyślnie 0 (bez sprawdzania); zmiana tego ustawienia wymaga ponownego uruchomienia aplikacji *Taskservice* w programie *Menedżer internetowych usług informacyjnych*.

> **UWAGA**
> Zmiana tych ustawień wymaga restartowania serwisu *taskservice* (należy restartować odpowiednią pulę aplikacji w Menedżerze internetowych usług informacyjnych).

WIDOK MOJE ZADANIA

- `Domyślny język widoku 'Moje zadania'`: określa domyślny język dla funkcji tabelarycznej *UserTasks_View*; używany w sytuacji, gdy w wywołaniu tej funkcji parametr języka nie zostanie podany.

> **UWAGA**
> Zmiana tej wartości powoduje aktualizację danych w tabeli *core.ACLInstanceEntries* na dane w nowym języku pobierane z tabeli tłumaczeń. Taka operacja jest czasochłonna i należy ją wykonywać gdy obciążenie aplikacji jest najmniejsze.

OCR

- `Częstotliwość sprawdzania kolejki procesów wykonywanych przez OCR`

`(w sekundach)`: interwał, z jakim *taskservice* sprawdza, czy w kolejce OCR (tabela *core.OCRProcessesQueue*) są wpisy wymagające przetworzenia; wartość domyślna 0 oznacza, że kolejka nie jest sprawdzana, czyli procesy OCR nie są przetwarzane. W przypadku korzystania z funkcji OCR zaleca się ustawienie wartości 60 lub 120 s.

> **WAŻNE**
> Zmiana wartości tego ustawienia wymaga restartowania witryny lub puli aplikacji *taskservice*.

TRYB POJEDYNCZEJ APLIKACJI

Grupa opcji umożliwiających włączenie trybu pojedynczej aplikacji we FrontSPA, to jest filtrowanie sekcji i pozycji menu według wybranej aplikacji.

- `Włącz tryb pojedynczej aplikacji`: (domyślnie wyłączone) po włączeniu umożliwia filtrowanie sekcji menu wyświetlanych we FrontSPA według aplikacji przypisanej do tych sekcji i pozycji.
- `Domyślna aplikacja`: po włączeniu trybu pojedynczej aplikacji umożliwia wybranie aplikacji, której sekcje menu będą wyświetlane domyślnie. To ustawienie dotyczy użytkowników, którzy nie mają wybranej aplikacji domyślnej w profilu użytkownika.

ZASTĘPSTWA

- `Tworzenie zastępstw za użytkowników, których termin ważności konta wygasł`: domyślnie wyłączone; po włączeniu pozwala dodawać zastępstwa, za użytkowników z profilem, dla którego upłynął termin ważności.

PRACA NA URZĄDZENIACH MOBILNYCH

- `Odświeżanie ekranu przez gest przesunięcia palcem`: (domyślnie włączone) podczas pracy na urządzeniu mobilnym blokuje gest *pull to refresh*, który domyślnie powoduje odświeżenie ekranu po przesunięciu palcem w dół na ekranie.

IMPORT UŻYTKOWNIKÓW

- `Wybierz plik`: kliknij, aby wybrać plik programu Excel z danymi użytkowników do zaimportowania.
- `Importuj użytkowników z pliku Excel`: wykonuje import użytkowników z

wybranego pliku.

- `Eksportuj szablon Excel`: zapisuje pusty arkusz Excel z kolumnami, które należy wypełnić danymi użytkowników do zaimportowania.

> **WSKAZÓWKA**
> Dla zaimportowanych użytkowników automatycznie jest włączane ustawienie `Dostęp do Internal API`.

CACHE

Grupa opcji umożliwiających czyszczenie pamięci podręcznej dla wybranych obiektów nAxiom oraz wyczyszczenie całej pamięci podręcznej przechowywanej w API. W celu wyczyszczenia pamięci podręcznej dla określonego rodzaju obiektów, należy kliknąć przycisk `Wyczyść` obok nazwy tej kategorii. Dostępne kategorie to:

- `Applications` (aplikacje)
- `VarsMode` (tryb zmiennych środowiskowych)
- `Languages` (języki)
- `Translations` (tłumaczenia)
- `Dictionary` (słownik)
- `HierarchicalDictionary` (słownik hierarchiczny)
- `EscalatorNotificationsSettings` (ustawienia powiadomień eskalatora)
- `Grid` (listy)
- `Form` (formularze)
- `BusinessDocuments` (definicje dokumentów biznesowych)
- `Business` (inne obiekty biznesowe)
- `Actions` (akcje)
- `Monaco` (ustawienia asysty podczas wprowadzania, *smart numbers*)

Przycisk `Wyczyść cały cache` opróżnia całą pamięć podręczną witryny nAxiom dla bieżącego tenanta.

> **WSKAZÓWKA**
> Czyszczenie pamięci podręcznej można zautomatyzować za pomocą akcji C#, używając w niej wbudowanej funkcji *ClearCache()*. W odróżnieniu od przycisku `Wyczyść cały cache`, funkcja *ClearCache()* czyści pamięć podręczną dla wszystkich tenantów.

8.3. TŁUMACZENIA I POMOC

Podsumowanie:
Funkcja tłumaczeń służy do zarządzania wersjami językowymi aplikacji tworzonych na platformie nAxiom. Pozwala ona dodawać nowe języki oraz eksportować i importować tłumaczenia nazw i opisów elementów interfejsu użytkownika oraz obiektów biznesowych zdefiniowanych w nAxiom. Zmiana wersji językowej interfejsu aplikacji odbywa się automatycznie na podstawie języka przypisanego w profilu użytkownika.

☛ *AdminSPA* > `ADMINISTRACJA` > `Tłumaczenia i pomoc`

W TYM TEMACIE

- Implementacja mechanizmu tłumaczeń
- Strona Tłumaczenia i pomoc
- Zmiana wersji językowej

IMPLEMENTACJA MECHANIZMU TŁUMACZEŃ

Idea obsługi różnych wersji językowych polega na tym, że wszystkie wartości tekstowe widoczne w interfejsie użytkownika są reprezentowane w tabelach bazy danych przez identyfikatory GUID, a ich wyświetlane wartości są pobierane z tabel *core.TranslationDefinitions* i *core.Translations*.

Funkcjonalność można objaśnić na przykładzie opisu definicji dokumentu. Podczas tworzenia nowej definicji w tabeli *Core.BusinessDocuments* w kolumnie *Description* jest zapisywany identyfikator GUID. Jednocześnie do tabeli *core.TranslationDefinitions* dodawany jest wpis określający *definicję tłumaczenia*. Konstrukcja tej tabeli jest następująca:

Nazwa kolumny	Typ danych	Zezwól NULL	Opis
Id	int	☐	Kolumna klucza głównego, identyfikator rekordu w tabeli
RecordId	int	☑	Identyfikator rekordu w tabeli z kolumny *TableName*
TableName	nvarchar(MAX)	☐	Nazwa tabeli obiektów, której dotyczy definicja tłumaczenia
Field	nvarchar(MAX)	☐	Nazwa pola, do którego odwołuje się definicja tłumaczenia
Description	nvarchar(MAX)	☐	Pomocniczy opis identyfikujący źródłowy obiekt tłumaczenia, np. *Dokument biznesowy: Opis*
Guid	uniqueidentifier	☐	Identyfikator GUID wartości tekstowej
RowGuid	uniqueidentifier	☐	Identyfikator GUID bieżącego rekordu
RecordRowGuid	uniqueidentifier	☐	Identyfikator GUID rekordu w tabeli z kolumny *TableName*

Przykładowy wpis do tabeli *core.TranslationDefinitions*, który określa definicję tłumaczenia wartości tekstowej z kolumny *Description* w pierwszym rekordzie tabeli *BusinessDocuments*, może wyglądać jak niżej:

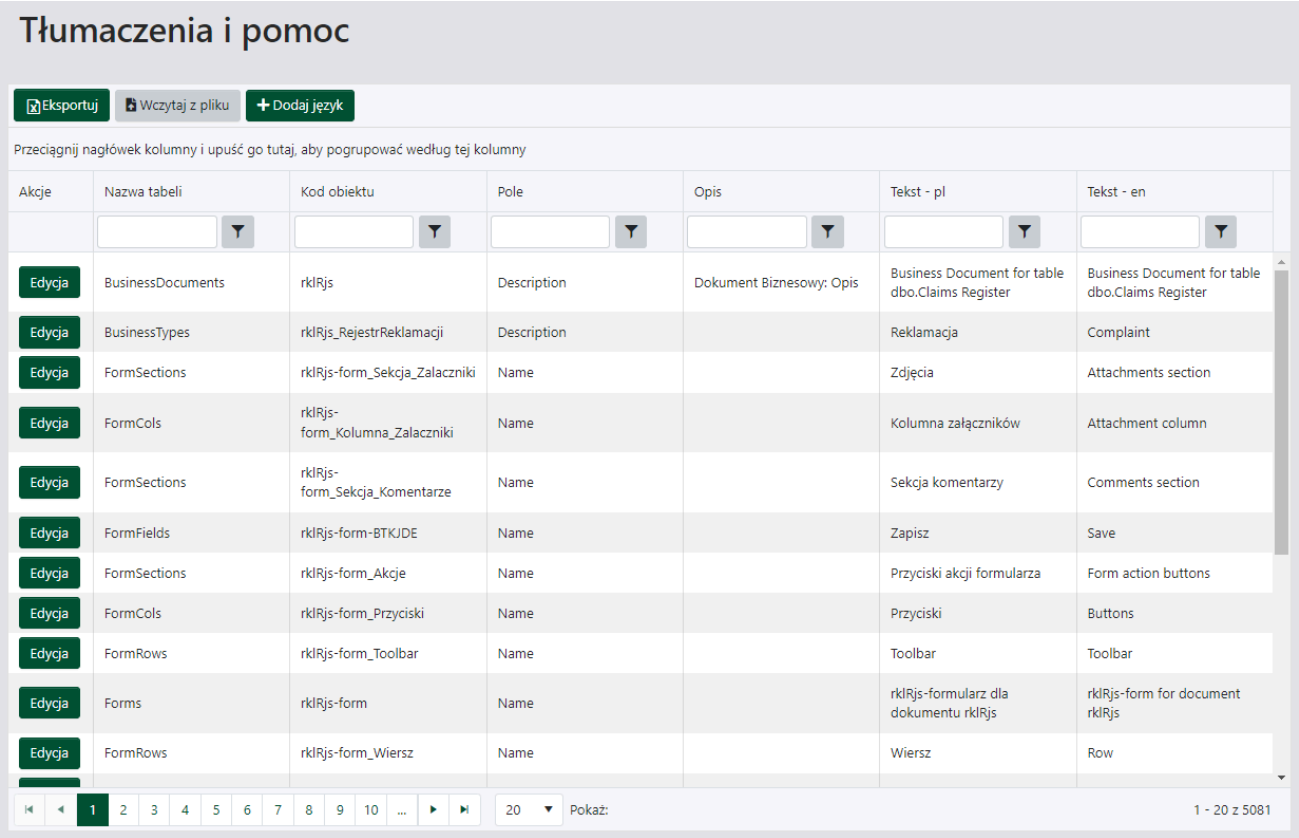
- *Id*: 1
- *RecordId*: 1
- *TableName*: BusinessDocuments
- *Field*: Description
- *Description*: Dokument biznesowy: Opis
- *Guid*: 1CB31058-415F-4F45-B46D-D3A05BBFC2B9
- *RowGuid*: C4A8C819-7BFB-4B92-B118-576C6DD45C71
- *RecordRowGuid*: 631FFC2D-0BE7-4C0D-8993-A95BB3B52B7E

Natomiast rzeczywiste wartości tekstowe odpowiadające zapisanemu w definicji identyfikatorowi GUID są zapisywane w tabeli *core.Translations*:

Nazwa kolumny	Typ danych	Zezwól NULL	Opis
Id	int	☐	Kolumna klucza głównego, identyfikator rekordu w tabeli
LanguageId	int	☐	Identyfikator języka
TranslationDefinitionId	int	☐	Identyfikator definicji tłumaczenia z tabeli *core.TranslationDefinitions*
Value	nvarchar(MAX)	☐	Wartość tekstowa odpowiednia dla identyfikatora języka
RowGuid	uniqueidentifier	☐	Identyfikator GUID bieżącego rekordu

STRONA TŁUMACZENIA I POMOC

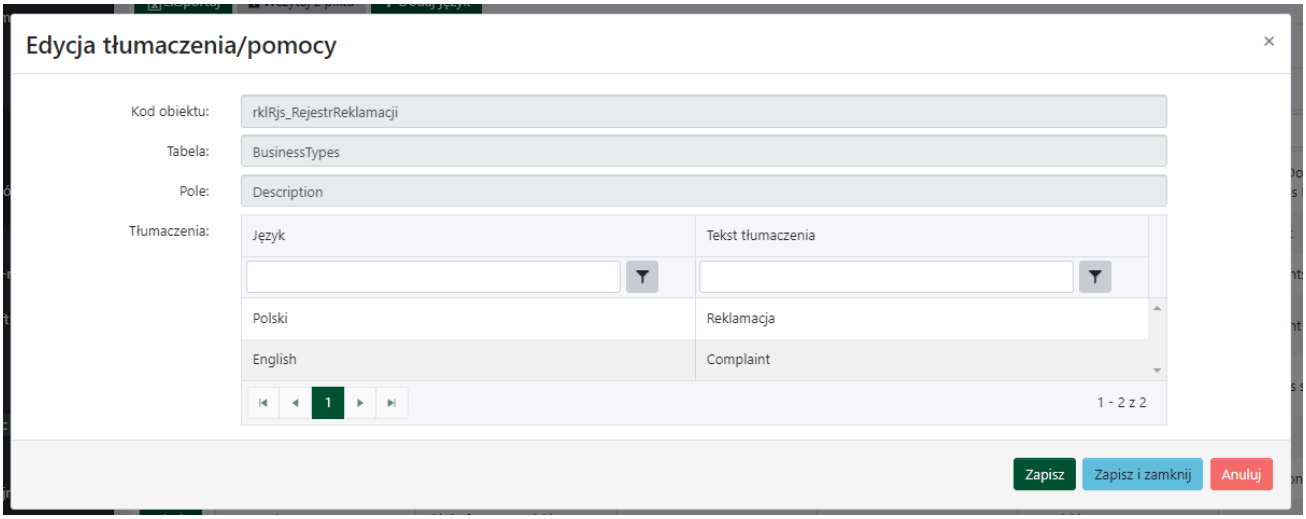
Wartości z obu tabel są wyświetlane w *AdminSPA* na stronie `Tłumaczenia i pomoc` (`ADMINISTRACJA` > `Tłumaczenia i pomoc`).



Tłumaczenia i pomoc

Kolumny wyświetlanej tabeli odpowiadają kolumnom tabel źródłowych. Dostępne są funkcje filtrowania i sortowania, które ułatwiają znalezienie żądanego wpisu. Kliknięcie

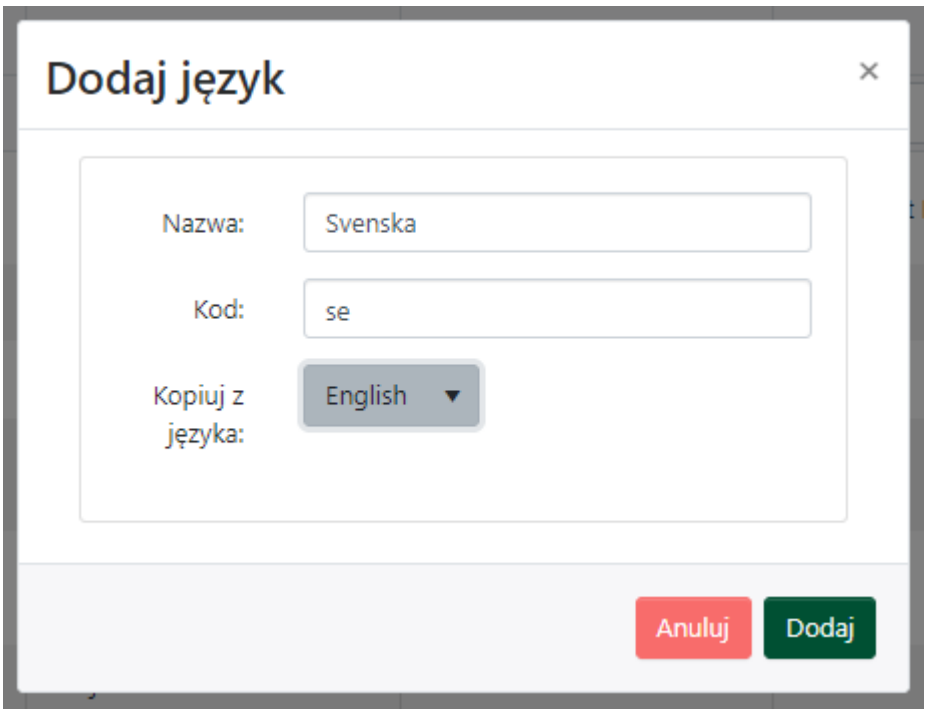
przycisku `Edycja` powoduje wyświetlenie okna dialogowego, w którym można zmienić tekst tłumaczenia.



Edycja tłumaczenia/pomocy

Dodatkowo, z lewej strony u góry są dostępne trzy przyciski:

- `Eksportuj`: eksportuje wyświetlaną listę w formacie programu MS Excel; plik jest zapisywany w domyślnym folderze przeglądarki dla plików pobieranych.
- `Wczytaj z pliku`: wyświetla okno wyboru pliku w celu wybrania pliku z tłumaczeniami; w wyeksportowanym pliku należy wstawić tłumaczenia w kolumnie odpowiedniego języka.
- `Dodaj język`: wyświetla okno dialogowe, w którym należy zdefiniować nowy język; dostępne pola:
  - `Nazwa`: nazwa języka.
  - `Kod`: dwuliterowy kod języka.
  - `Kopiuj z języka`: lista rozwijana umożliwiająca wybór języka źródłowego, z którego ma zostać wybrane tłumaczenie.

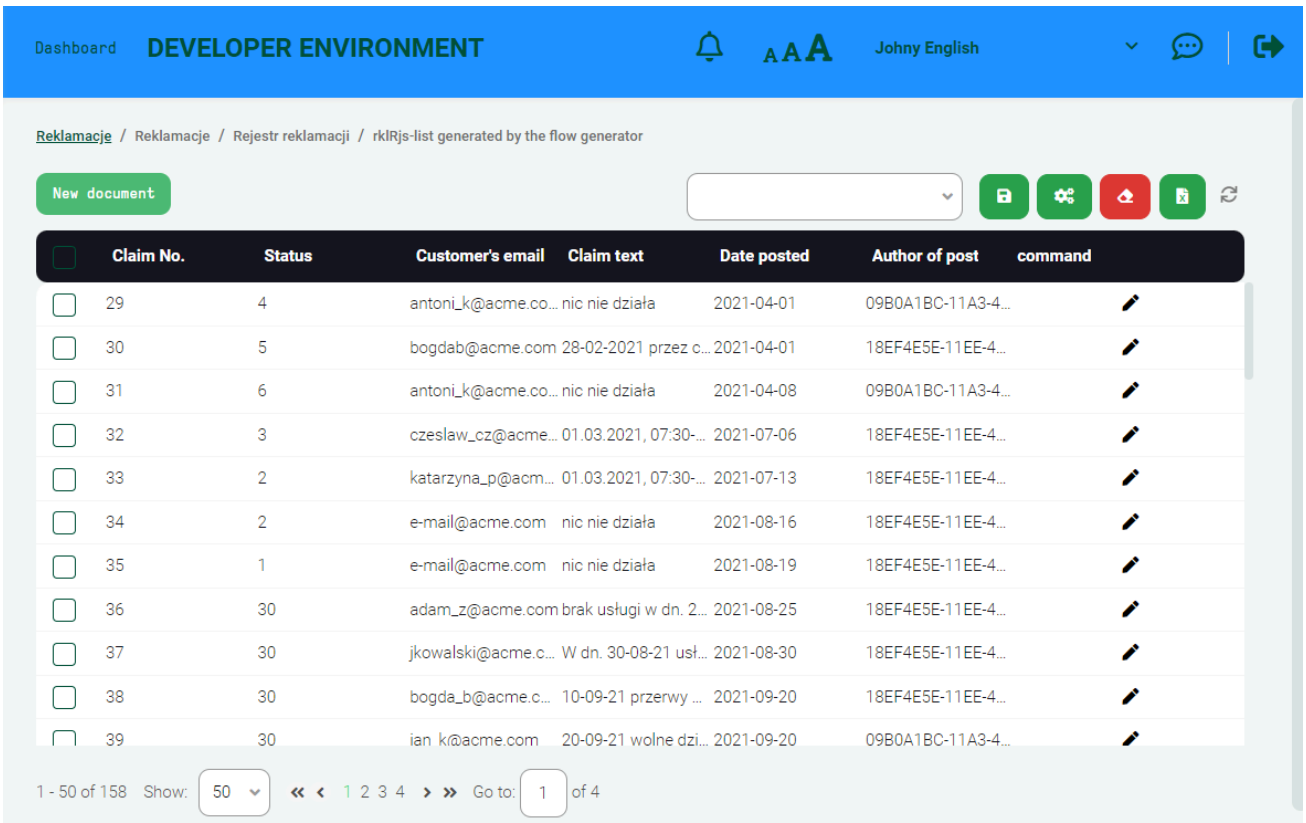


Okno Dodaj język

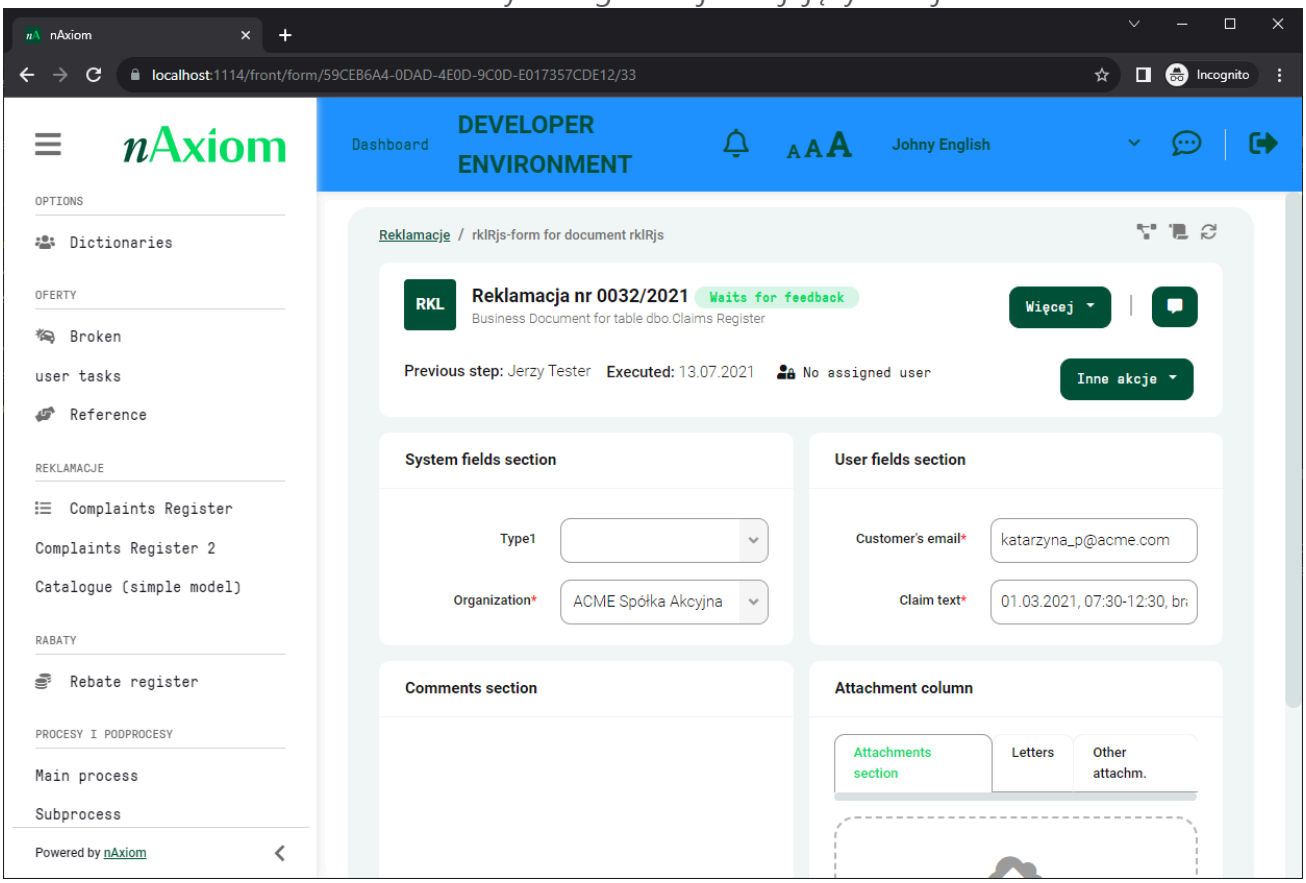
Funkcja tłumaczeń jest też dostępna w oknie kreatora formularzy, aby ją wywołać należy kliknąć przycisk `Tłumaczenia i pomoc` z prawej strony u góry.

ZMIANA WERSJI JĘZYKOWEJ

nAxiom automatycznie przełącza wersję językową aplikacji w przypadku użytkownika, który w swoim profilu ma zadeklarowany język inny niż polski. Na ilustracjach poniżej przedstawiono przykładowy wygląd widoku listy i widoku formularza po stronie front-office.



Widok listy w angielskiej wersji językowej



Widok formularza w angielskiej wersji językowej

Powiązane tematy:

- Tworzenie i edycja profilu użytkownika

8.4. LISTA APLIKACJI

Podsumowanie:
Aplikacje w nAxiom to *kontenery*, do których przypisuje się wszystkie obiekty składające się na określone funkcjonalności, to jest: role, źródła danych, sekcje menu, słowniki, kontrolki, akcje, widoki listy, formularze, definicje dokumentów biznesowych, widgety, szablony e-mail, mail monitory oraz szablony word. Podstawowe funkcje podziału na aplikacje to:

- możliwość eksportu i importu aplikacji w formie paczki z wszystkimi potrzebnymi do jej działania obiektami,
- możliwość zarządzania dostępem konsultantów do indywidualnych aplikacji,
- możliwość wyświetlania dla użytkowników końcowych sekcji menu przypisanych do wybranej aplikacji (tzw. tryb pojedynczej aplikacji)

  Dodatkowym obiektem ułatwiającym organizację rozbudowanych aplikacji są moduły. W każdej aplikacji można zdefiniować wiele modułów

☛ Menu *AdminSPA* > `ADMINISTRACJA` > `Lista aplikacji`.

Polecenie `Lista aplikacji` wyświetla stronę z listą aplikacji zdefiniowanych w danej witrynie nAxiom. Na liście zawsze znajduje się predefiniowana aplikacja systemowa *BaseApp* z modułem *BaseModule*.

INFO
Definicję aplikacji można usunąć tylko wtedy, gdy nie przypisano do niej żadnych obiektów.

1. Kliknij przycisk `Dodaj definicję` w oknie `Lista aplikacji`.
   Zostanie wyświetlone okno dialogowe `Nowa definicja aplikacji...`.

2. Wypełnij pola na karcie `Ogólne` i kliknij przycisk `Zapisz`; dopiero po zapisaniu definicji zostaną udostępnione pozostałe karty w tym oknie dialogowym.

3. Przejdź na kartę `Uprawnienia` i ustaw odpowiednie uprawnienia do tej aplikacji dla ról biznesowych i użytkowników. Zapisz zmienione uprawnienia indywidualnie dla każdej roli i dla każdego użytkownika.

   > **UWAGA**
   > Zmienione uprawnienia zaczną obowiązywać dopiero po wylogowanie się i ponownym zalogowaniu użytkownika, którego dotyczy zmiana.

4. Określ ustawienia aplikacji na pozostałych kartach zgodnie z opisem w kolejnych tematach.

> **Powiązane tematy:**
> - Tryb pojedynczej aplikacji

8.4.1. Karta Ogólne

> **Podsumowanie:**
> Na karcie `Ogólne` definiuje się podstawowe parametry aplikacji, takie jak `Kod`, `Nazwa`, `Opis` oraz ustawienia dotyczące trybu pojedynczej aplikacji.

> ☛ `ADMINISTRACJA` > `Lista aplikacji` > `Dodaj definicję/Edytuj` > `Ogólne`

- `Kod`: kod aplikacji; dozwolone tylko znaki alfanumeryczne i znak podkreślenia; maks. długość 50 znaków.

> **WAŻNE**
> Aby składnia smart numbers działała poprawnie, kod w definicji obiektu nie może zawierać znaków diakrytycznych.

- `Nazwa`: nazwa aplikacji.
- `Opis`: opis aplikacji.
- `Akronim aplikacji`: maksymalnie 4-znakowy skrót, który będzie wyświetlany we FrontSPA w trybie pojedynczej aplikacji.
- `Wyświetl nazwę wersji`: po włączeniu wyświetla nazwę wersji pod nazwą aplikacji w aplikacji FrontSPA w trybie pojedynczej aplikacji; patrz opis karty Wersje .
- `Widoczna na liście wyboru aplikacji`: (domyślnie włączone) określa, czy daną aplikację można wybrać w trybie pojedynczej aplikacji we FrontSPA.
- `Dostęp dla posiadaczy ról globalnych`: użytkownicy z co najmniej jedną z ról

wybranych na tej liście wielowyboru będą mogli wybierać tę aplikację na liście w trybie pojedynczej aplikacji oraz ustawiać ją jako aplikację domyślną w swoim profilu; jeśli żadna rola nie zostanie wybrana, aplikacja będzie dostępna na liście dla wszystkich użytkowników. Jeśli przełącznik widoczności aplikacji będzie wyłączony, to ustawienie będzie ignorowane.

- `Domyślny pulpit`: pulpit aplikacji wyświetlany we FrontSPA po przełączeniu na tę aplikację; dotyczy trybu pojedynczej aplikacji.

WAŻNE
W przypadku nowej aplikacji konieczne jest zapisanie ustawień na karcie `Ogólne`, aby pozostałe karty stały się dostępne.

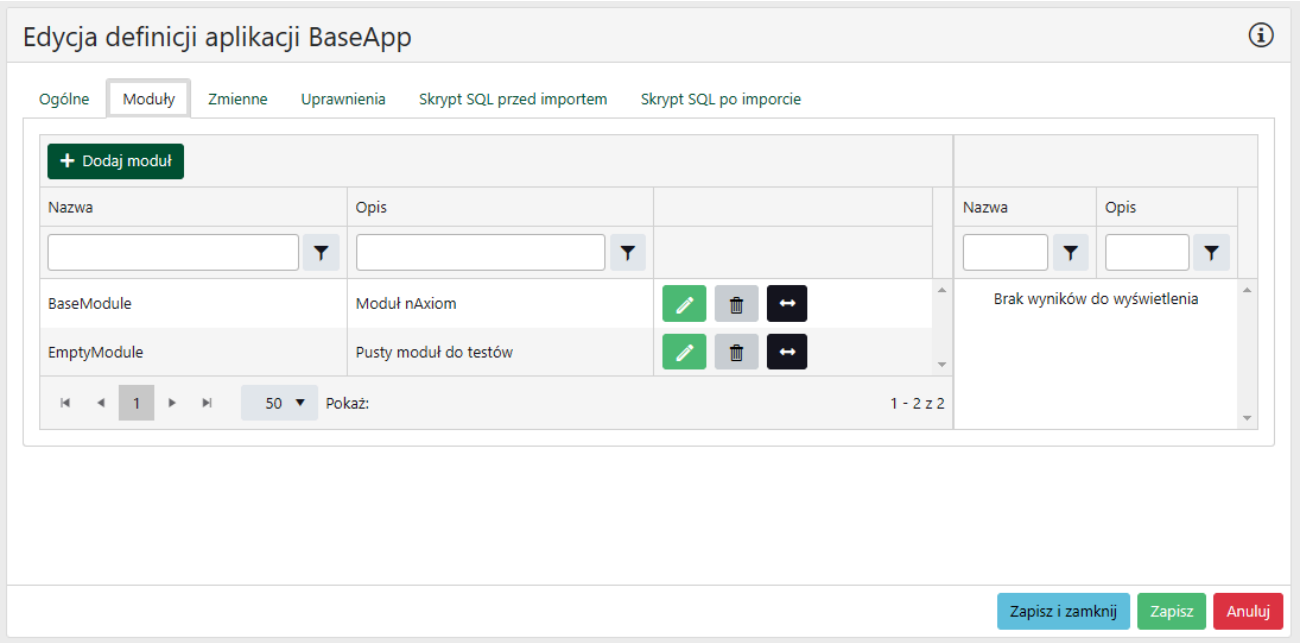
Powiązane tematy:
- Tryb pojedynczej aplikacji
- Pulpity aplikacji

8.4.2. Karta Moduły

Podsumowanie:
Moduł to podzbiór obiektów w ramach aplikacji. Moduły aplikacji definiuje się na karcie `Moduły`, podając nazwę i opis modułu. Kiedy definiujesz nowy obiekt, oprócz nazwy aplikacji musisz podać nazwę modułu z tej aplikacji.

☛ `ADMINISTRACJA` > `Lista aplikacji` > `Dodaj definicję/Edytuj` > `Moduły`

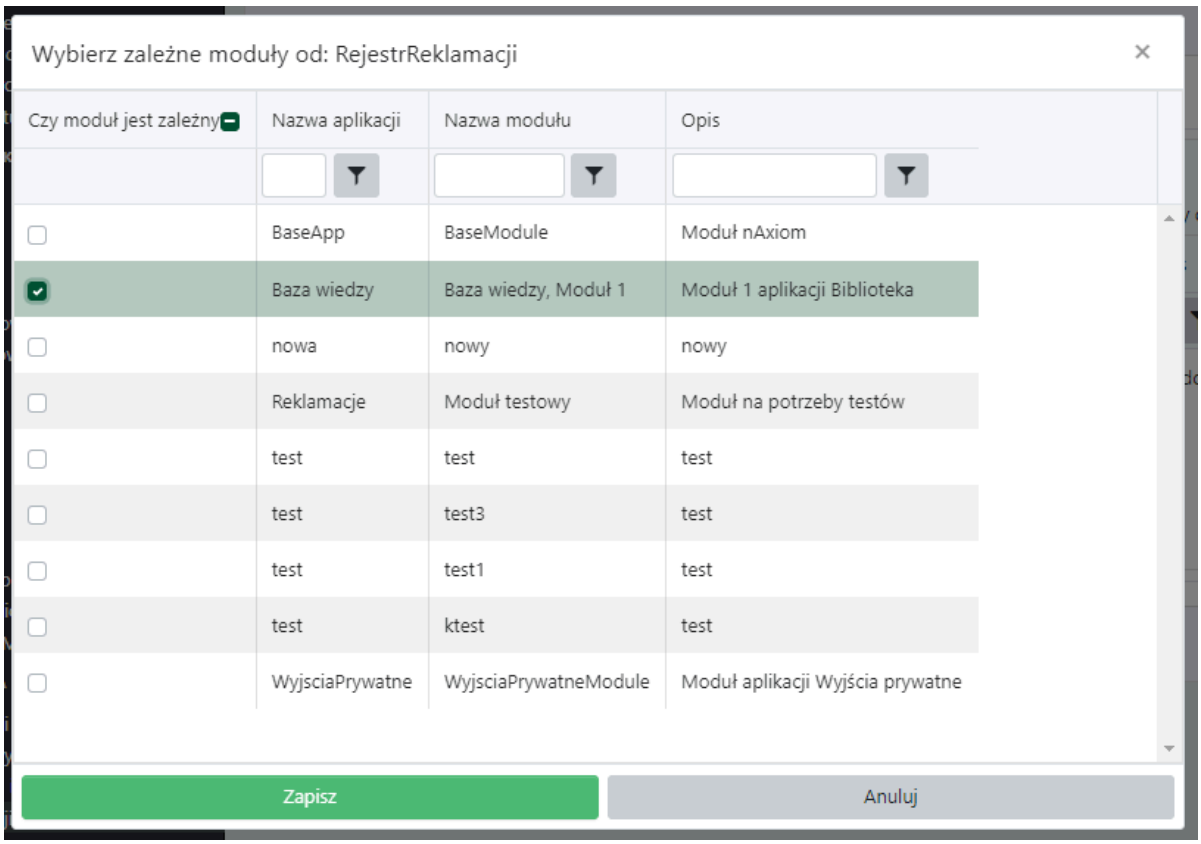
WAŻNE
Aby składnia smart numbers działała poprawnie, nazwa modułu nie może zawierać znaków diakrytycznych.



Karta Moduły

Karta `Moduły` to lista modułów zdefiniowanych w aplikacji. Ponadto na tej karcie z prawej strony jest wyświetlana lista modułów powiązanych z wybranym modułem. Na karcie `Moduły` są dostępne następujące kontrolki:

- `Dodaj moduł`: wstawia na liście modułów pusty wiersz w trybie edycji inline; wpisz wartości w kolumnach `Nazwa` i `Opis`, a następnie kliknij przycisk (`Zapisz`), aby zapisać definicję modułu.
- (`Edytuj`): przełącza wiersz modułu w tryb edycji inline.
- (`Usuń`): usuwa definicję modułu; usunięcie modułu jest możliwe, o ile nie przypisano do niego żadnych obiektów.
- (`Dodaj powiązany moduł`): wyświetla okno dialogowe z listą modułów z innych aplikacji w celu powiązania z bieżącym modułem; zaznacz pola wyboru obok modułów, które chcesz powiązać z bieżącym modułem i kliknij przycisk `Zapisz`; powiązane moduły będą eksportowane razem z modułem bieżącym w *Migratorze aplikacji*.



Wybór modułów zależnych

8.4.3. Karta Wersje

Podsumowanie:
Na karcie `Wersje` można dodawać opisy zmian w aplikacji.

☛ `ADMINISTRACJA` > `Lista aplikacji` > `Dodaj definicję/Edytuj` > `Wersje`

Kliknij przycisk `Nowa wersja` i wypełnij pola `Nazwa wersji` (obowiązkowe, maks. 200 znaków) i `Opis wersji` (opcjonalne, maks. 2000 znaków). Ich wartości wraz z datą wpisu są zapisywane w tabeli `core.ApplicationDefinitionVersions`. Funkcjonalność pozwala rejestrować historię zmian w aplikacji oraz identyfikować kolejne wersje aplikacji. Historia aplikacji podlega migracji.

Wpisami historii wersji można także zarządzać w oknie `Lista aplikacji` po kliknięciu ikony +.

Powiązane tematy:

- Tryb pojedynczej aplikacji
- Pulpity aplikacji

8.4.4. Karta Zmienne

Podsumowanie:
Zmienne aplikacji to definiowane na poziomie definicji aplikacji zmienne, które mogą przyjmować różne wartości, zależnie od rodzaju środowiska, w jakim są używane. Odwołanie do zmiennej wymaga użycia składni smart numbers. Zmienne mogą być używane jako parametry akcji (np. adres serwera dla źródła danych), w edytorach kodu i szablonach oraz wielu polach tekstowych i hybrydowych tekst/SQL (informacja o możliwości użycia składni smart numbers jest podana jako „placeholder”).

☛ `ADMINISTRACJA` > `Lista aplikacji` > `Dodaj definicję/Edytuj` > `Zmienne`

DEFINIOWANIE ZMIENNYCH

Aby zdefiniować zmienną aplikacji, kliknij przycisk `Dodaj` i określ następujące właściwości:

`Kod`: symbol zmiennej używany w odwołaniach, w tym przez składnię smart numbers.

`Nazwa zmiennej`: nazwa zmiennej aplikacji.

`Typ`: typ danych zmiennej; lista wartości z typami danych obsługiwanymi w nAxiom.

`Wartości zmiennej`: wartości zmiennej, które będą używane w odpowiednich środowiskach; dla każdego środowiska (deweloperskie, testowe i produkcyjne) można zdefiniować wartość zmiennej typu:

- `Referencja do obiektu`: służy do pobierania sekretów tekstowych z sejfu zintegrowanego z nAxiom; ma postać ścieżki *TenantsAdmin.Secrets.Environment.<kodSekretu>* w przypadku sekretów środowiska lub *TenantsAdmin.Secrets.CurrentTenant.<kodSekretu>*; przycisk `Wybierz` z prawej strony pola wyświetla okno dialogowe, w którym można wybrać ścieżkę do sekretu; więcej informacji o sekretach zawiera artykuł Integracja nAxiom z OpenBao .
- `Wartość lokalna`: wartość zmiennej jako literał lub referencja smart number,

WSKAZÓWKA

W wartości zmiennej można także odwołać się do innej zmiennej, używając składni smart numbers z selektorem *AppInLineVariables*, oraz do wartości parametru systemowego (np. *{@_TenantBaseUrl}*) i parametrów z kontekstu formularza (z wyjątkiem tablic, np. *OpenedForms*). Można także w polach wartości używać składni smart numbers z innymi selektorami. Wartość zmiennej zostanie w takim przypadku wyliczona rekurencyjnie.

`Migruj wartości tej zmiennej`: zestaw przełączników do sterowania importowaniem wartości zmiennej na środowisko docelowe określonego typu; patrz sekcja poniżej.

Dodawanie zmiennej aplikacji

Zmienne aplikacji można między innymi zastosować do ustawienia różnych czasów kroku w procesie w różnych środowiskach, np. czas kwarantanny w środowisku deweloperskim i testowym może wynosić 30 sekund, a w środowisku produkcyjnym 14 dni. Innym przykładem mogą być różne adresy serwerów, np. *platnosci-blik-test* w środowiskach *dev/test* i *platnosci-blik* w środowisku *prod*.

ODWOŁYWANIE SIĘ DO ZMIENNYCH

Do wartości zmiennych aplikacji można odwoływać się za pomocą składni smart numbers na dwa sposoby:

```
{&NazwaAplikacji.AppVariables.KodZmiennej}
```

- ***AppVariables***: odwołanie przeznaczone do użycia w zapytaniach SQL (np. w klauzuli WHERE jako wartość filtrująca); przekazuje wartość zmiennej jako parametr zgodnie

z zadeklarowanym typem, dla typu tekstowego wartość zmiennej w zapytaniu jest renderowana w apostrofach

lub

```
{&NazwaAplikacji.AppInlineVariables.KodZmiennej}
```

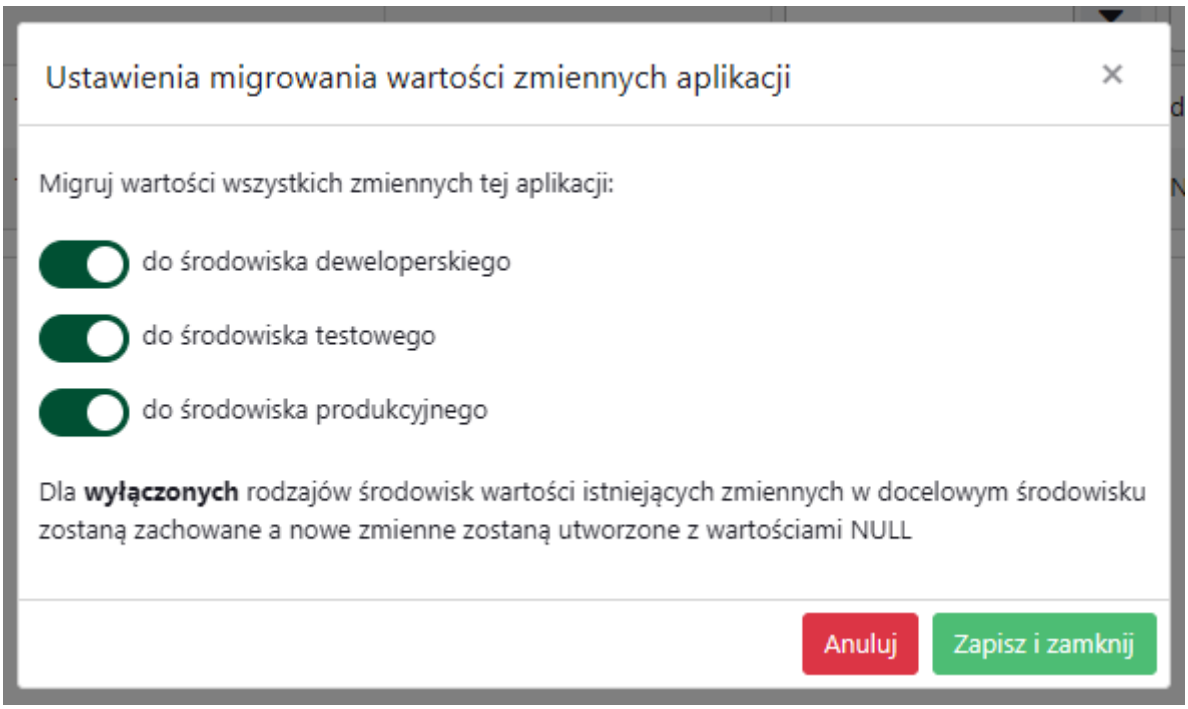
- ***AppInlineVariables***: zwraca zadeklarowaną wartość zmiennej (jako tekst); wywoływane w ten sposób zmienne środowiskowe mogą na przykład być używane do podstawiania różnych nazw tabel w klauzuli *FROM* lub nazw kolumn w klauzuli *WHERE* w zapytaniach SQL, a także w akcjach C#, gdzie wartości typu tekstowego należy ująć w podwójne apostrofy (”).

UWAGA

- Migracja aplikacji powoduje nadpisanie wartości zmiennych aplikacji w środowisku testowym i produkcyjnym. Nie dotyczy to przypadku, kiedy zmienne w środowisku źródłowym mają wartość *NULL*.
- Jeśli w mechanizmie podpowiedzi nie ma kodu zmiennej, która została dopiero co zdefiniowana, należy wyczyścić pamięć podręczną aplikacji w ustawieniach systemu.

USTAWIENIA MIGROWALNOŚCI ZMIENNYCH

Konsultant może decydować o sposobie importu zmiennych aplikacji — a precyzyjniej, ich wartości — do środowiska docelowego. Służy do tego przycisk `Ustawienia migrowania zmiennych` widoczny w nagłówku listy zmiennych. Po kliknięciu tego przycisku jest wyświetlane okno dialogowe z trzema przełącznikami, które umożliwiają wyłączenie importu wartości zmiennych do środowiska określonego typu.



Ustawienia migrowania wartości zmiennych aplikacji

Przełączniki w tym oknie mają zastosowanie do wszystkich zmiennych danej aplikacji. Ustawienia migrowalności można konfigurować indywidualnie dla każdej zmiennej w oknie edycji zmiennej.

Zasada działania przełączników jest następująca:

- przełącznik włączony:
  - wartość danej zmiennej nadpisze wartość tej zmiennej w środowisku docelowym,
  - jeśli zmienna ma wartość *null*, taka wartość zostanie ustawiona w środowisku docelowym
  - jeśli zmienna nie istnieje w środowisku docelowym, zostanie w nim utworzona wraz z bieżącą wartością.
- przełącznik wyłączony:
  - wartość zmiennej w środowisku docelowym nie zmieni się,
  - jeśli zmienna nie istnieje w środowisku docelowym, zostanie w nim utworzona z wartością *null*.

Powiązane tematy:

-
- Smart numbers
- Ustawienia systemu, sekcja Cache

8.4.5. Karta Uprawnienia

Podsumowanie:
Po utworzeniu definicji aplikacji uprawniony konsultant musi nadać innym konsultantom uprawnienia do tworzenia tej aplikacji, w tym do odczytu definicji aplikacji i jej obiektów, aktualizacji definicji aplikacji i jej obiektów oraz usuwania obiektów aplikacji i samej aplikacji. Tymi uprawnieniami zarządza administrator uprawnień do aplikacji (zaznaczone pole wyboru `Dostęp do gałęzi budowania aplikacji oraz do uprawnień aplikacji` na stronie `Administracja uprawnieniami` w aplikacji FrontSPA).

Uprawnienia PBA do aplikacji przypisywane rolom mogą być eksportowane wraz z aplikacją.

☛ `ADMINISTRACJA` > `Lista aplikacji` > `Dodaj definicję/Edytuj` > `Uprawnienia`

Sposób nadawania uprawnień na tej karcie jest analogiczny jak w przypadku uprawnień PBA w aplikacji FrontSPA. Należy wybrać rolę biznesową lub użytkownika, a następnie zaznaczyć pole wyboru obok odpowiedniego uprawnienia.

UWAGA
Zmiany uprawnień należy zapisywać indywidualnie dla każdego użytkownika/roli; w przeciwnym razie przejście do kolejnego użytkownika/roli spowoduje utratę informacji o ustawionych uprawnieniach.

Dostępne uprawnienia to:

- `Dostęp do definicji aplikacji`: pozwala wyświetlić definicję aplikacji oraz definicje wszystkich obiektów przypisanych do tej aplikacji; brak tego uprawnienia pozwala wyłącznie na korzystanie z obiektów aplikacji, ale bez możliwości podglądu ich definicji.
- `Aktualizacja definicji aplikacji oraz jej obiektów/modułów`: pozwala edytować definicję aplikacji oraz definicje wszystkich obiektów przypisanych do aplikacji, w tym tworzyć nowe obiekty.
- `Usuwanie definicji aplikacji oraz jej obiektów`: pozwala usuwać wszystkie obiekty przypisane do aplikacji oraz samą definicję aplikacji; usunięcie aplikacji jest możliwe dopiero po usunięciu wszystkich obiektów aplikacji.
- `Edycja pulpitów aplikacji`: dotyczy FrontSPA, pozwala dodawać/usuwać widżety w pulpitach przypisanych do danej aplikacji (w definicji pulpitu, nie jako

pulpit domyślny)

UWAGI:

1. Uprawnienia PBA są weryfikowane podczas logowania użytkownika, dlatego każda zmiana uprawnień zalogowanego użytkownika zacznie obowiązywać dopiero po jego wylogowaniu i ponownym zalogowaniu.
2. Uprawnienia PBA do aplikacji, które są przypisane na poziomie roli, można eksportować wraz z aplikacją. W tym celu należy włączyć przełącznik w kolumnie `Eksportuj uprawnienia PBA` na liście wyboru aplikacji w Migratorze aplikacji. Domyślnie eksport uprawnień PBA dla ról jest wyłączony.
3. Wszelkie zmiany uprawnień należy zapisać przyciskiem `Zapisz` pod listą uprawnień z prawej strony. Przyciski `Zapisz` ani `Zapisz i zamknij` w oknie dialogowym definicji aplikacji nie zapisują tych zmian!

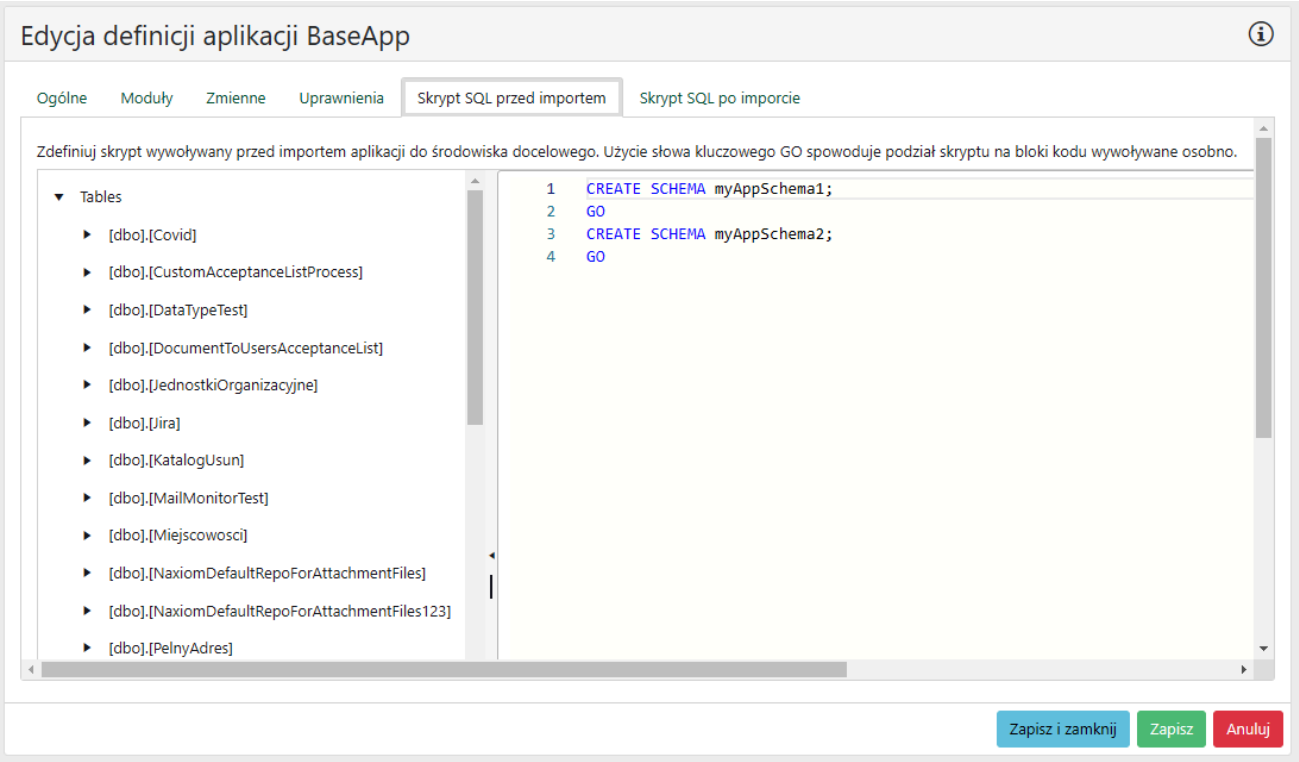
Powiązane tematy:

- Model uprawnień PBA - Wprowadzenie
- Migrator aplikacji

8.4.6. Karty Skrypt SQL przed importem/po imporcie

Podsumowanie:
Dla każdej aplikacji można zdefiniować skrypty SQL, które zostaną wykonane na środowisku docelowym przed importem aplikacji i/lub po jej zaimportowaniu. Takie skrypty pozwalają dostosować środowisko docelowe do potrzeb importowanej aplikacji.

☛ `ADMINISTRACJA` > `Lista aplikacji` > `Dodaj definicję/Edytuj` > `Skrypt SQL przed importem/po imporcie`



Na każdej z tych kart jest dostępne okno edytora SQL, w którym można wpisać instrukcje skryptów. Aby umożliwić uruchamianie skryptów złożonych z wielu osobnych instrukcji, w edytorze SQL na tej karcie zaimplementowano obsługę komendy *GO*.

```
CREATE SCHEMA myAppSchema1;
GO
CREATE SCHEMA myAppSchema2;
GO
```

UWAGA

Komenda *GO* nie jest obsługiwana w żadnych innych instrukcjach SQL, oprócz skryptów migracyjnych. W przypadku sprawdzania poprawności składni SQL dla skryptu jak w przykładzie powyżej, jest wyświetlany komunikat o błędzie, jednak skrypt można zapisać i jest on wykonywany poprawnie. Ten problem zostanie rozwiązany w kolejnej wersji nAxiom.

Skrypty są uruchamiane w ramach transakcji importu; błąd wykonania skryptu o odpowiedniej istotności przerywa wykonanie transakcji.

Powiązane tematy:

- Migrator aplikacji

8.5. ROLE BIZNESOWE

Podsumowanie:
Rola biznesowa to atrybut użytkownika organizacji. W nAxiom role biznesowe są używane do następujących celów:

- udostępnianie użytkownikom sekcji i pozycji menu w aplikacji *FrontSPA*,
- udostępnianie użytkownikom przycisków zmiany statusu dokumentu (przejścia na diagramie procesów),
- wyznaczanie użytkowników do akceptacji dokumentu,
- określanie uprawnień ACL (odczyt, aktualizacja i usuwanie dokumentów),
- określanie uprawnień PBA (dostęp do funkcjonalności nAxiom i aplikacji biznesowych).

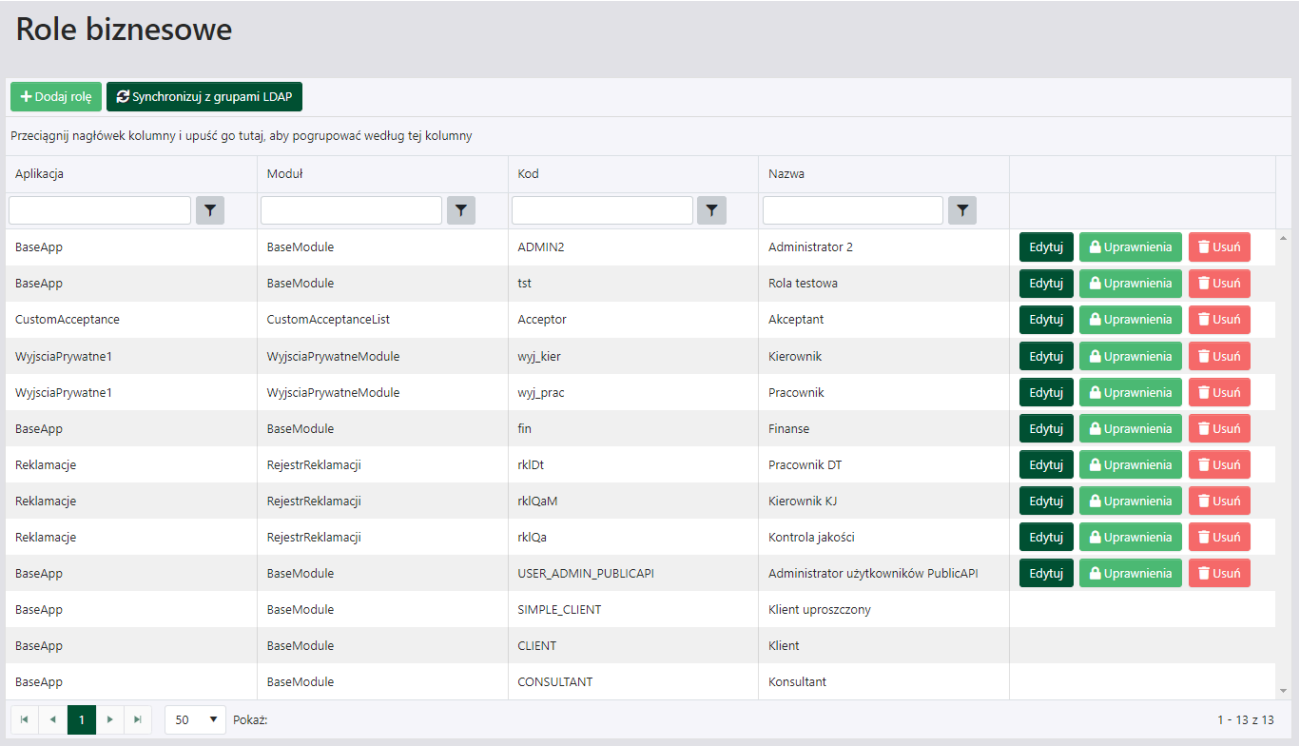
☛ *AdminSPA* > `ADMINISTRACJA` > `Role biznesowe`

W nAxiom wyróżnia się dwie kategorie ról: systemowe i biznesowe. Predefiniowane role systemowe opisano krótko w rozdziale Licencjonowanie . Tematem tego rozdziału będą role biznesowe.

Role przypisuje się użytkownikom w ich profilach. Po przypisaniu użytkownika do jednostki organizacyjnej możliwe jest także przypisanie roli w tej jednostce. W odniesieniu do ról przypisywanych bez powiązania z jednostkami używa się czasem określenia „rola globalna”.

W nAxiom można także używać definicji ról biznesowych zaimportowanych z systemów zewnętrznych.

W celu wyświetlenia listy ról kliknij polecenie `ADMINISTRACJA` > `Role biznesowe` w menu *AdminSPA*.

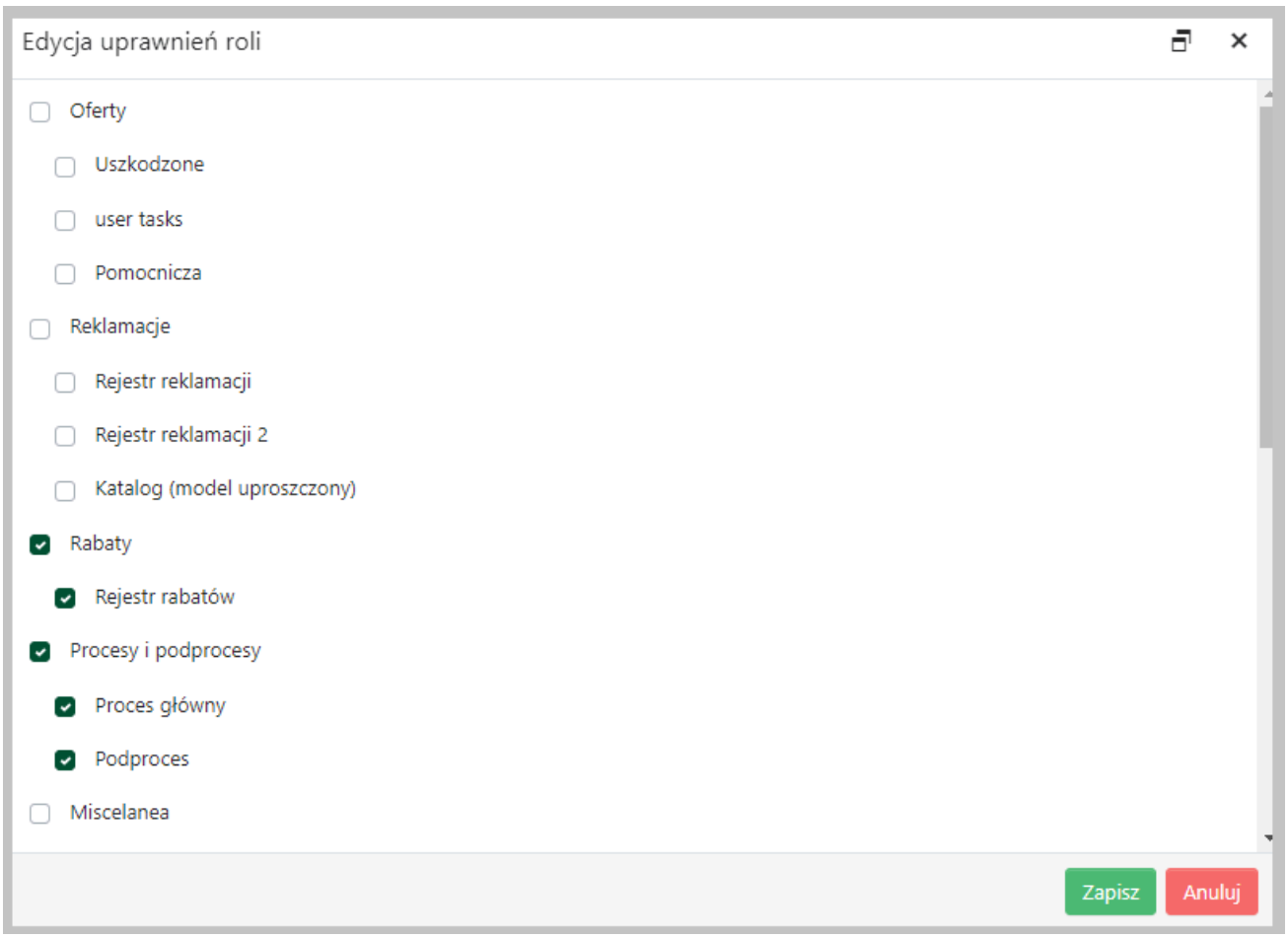


Role biznesowe

Na liście są wyświetlane zarówno role biznesowe, jak i role systemowe. Dla zdefiniowanych ról wyświetlana jest nazwa aplikacji, modułu, kod i nazwa. Dodatkowo dla ról biznesowych dostępne są przyciski `Edytuj`, `Uprawnienia` i `Usuń`. Rolę można usunąć tylko wtedy, gdy nie jest przypisana do żadnego profilu uzytkownika.

Aby dodać nową rolę, kliknij przycisk `Dodaj rolę` u góry lista z lewej strony. W wyświetlonym oknie dialogowym wybierz aplikację i moduł oraz wpisz kod i nazwę roli.

Po kliknięciu przycisku `Uprawnienia` jest wyświetlane okno dialogowe `Edycja uprawnień roli` z listą zdefiniowanych sekcji i pozycji menu. Aby użytkownik, któremu przypisano określoną rolę, miał dostęp do danej pozycji wyboru, należy zaznaczyć pole wyboru obok tej pozycji. Na liście wyświetlane są sekcje i pozycje menu zdefiniowane przez konsultanta. Predefiniowane pozycje menu, w tym: `Lista użytkowników`, `Słowniki`, `Akcje zdarzeń systemowych`, `OCR`, `Struktura organizacyjna`, `Zastępstwa` i `Uprawnienia`, są dostępne w sekcji menu `Ustawienia systemowe`, pod warunkiem posiadania odpowiednich uprawnień PBA.



Edycja uprawnień roli

Przycisk `Synchronizuj z grupami AD` importuje listę ról z zewnętrznego serwera LDAP. Ta funkcja wymaga skonfigurowania połączenia z serwerem w ustawieniach systemu (`ADMINISTRACJA` > `Ustawienia systemu` > `LDAP`).

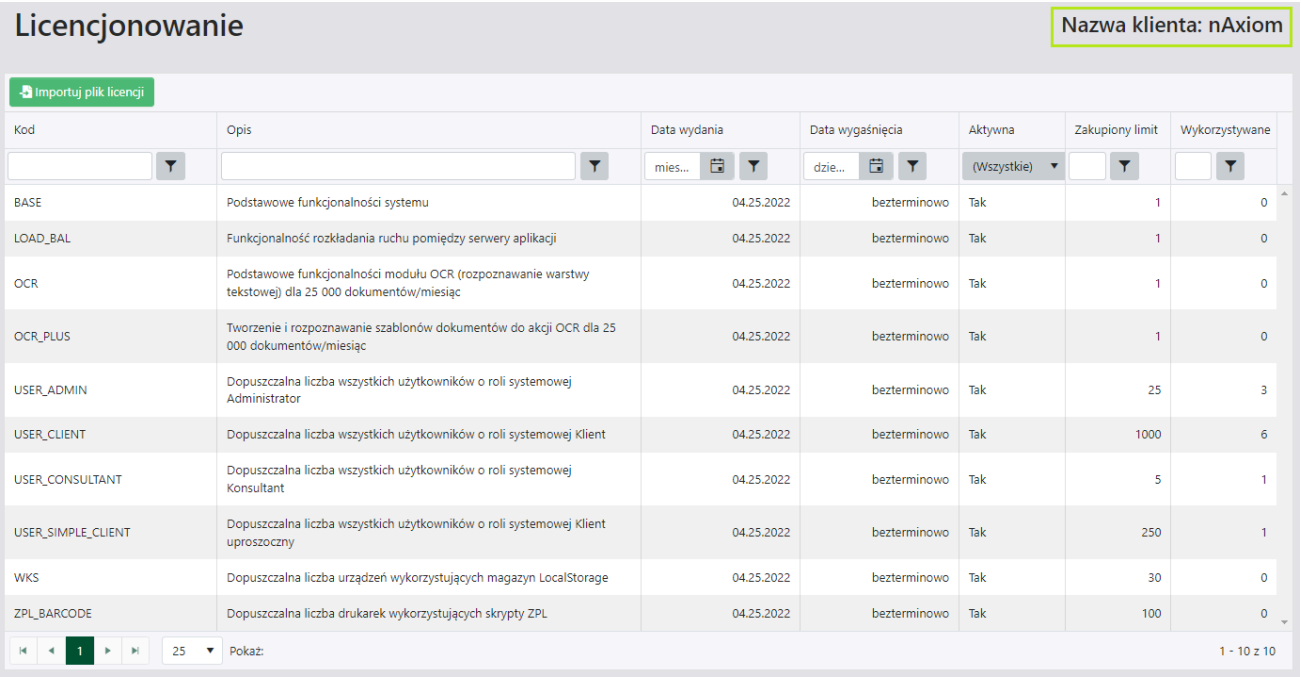
Powiązane tematy:

-
- Zadanie ręczne, Warunek wyświetlania przycisku
- Akceptacja
- Uprawnienia ACL do dokumentów
- Uprawnienia PBA, obszar Użytkowanie aplikacji
- Menu użytkownika
- Administracja, Ustawienia systemu

8.6. LICENCJONOWANIE

Podsumowanie:
Podstawą korzystania z nAxiom jest licencja, która określa liczbę stanowisk dla określonych ról systemowych; niezależnie licencjonowane są dodatkowe funkcjonalności, np. OCR (optyczne rozpoznawanie tekstu) i drukowanie kodów kreskowych. Model licencjonowania nAxiom opiera się na licencjach CAL (*Client Access License*). Każda licencja jest generowana dla określonego ciągu znaków (nazwy klienta). Podczas instalacji należy podać tę nazwę. Jeśli licencja nie zostanie wczytana podczas instalacji, będzie to możliwe podczas pierwszego uruchomienia.

☛ Menu *AdminSPA* > `ADMINISTRACJA` > `Licencjonowanie`



Kod	Opis	Data wydania	Data wygaśnięcia	Aktywna	Zakupiony limit	Wykorzystywane
BASE	Podstawowe funkcjonalności systemu	04.25.2022	bezterminowo	Tak	1	0
LOAD_BAL	Funkcjonalność rozkładania ruchu pomiędzy serwery aplikacji	04.25.2022	bezterminowo	Tak	1	0
OCR	Podstawowe funkcjonalności modułu OCR (rozpoznawanie warstwy tekstowej) dla 25 000 dokumentów/miesiąc	04.25.2022	bezterminowo	Tak	1	0
OCR_PLUS	Tworzenie i rozpoznawanie szablonów dokumentów do akcji OCR dla 25 000 dokumentów/miesiąc	04.25.2022	bezterminowo	Tak	1	0
USER_ADMIN	Dopuszczalna liczba wszystkich użytkowników o roli systemowej Administrator	04.25.2022	bezterminowo	Tak	25	3
USER_CLIENT	Dopuszczalna liczba wszystkich użytkowników o roli systemowej Klient	04.25.2022	bezterminowo	Tak	1000	6
USER_CONSULTANT	Dopuszczalna liczba wszystkich użytkowników o roli systemowej Konsultant	04.25.2022	bezterminowo	Tak	5	1
USER_SIMPLE_CLIENT	Dopuszczalna liczba wszystkich użytkowników o roli systemowej Klient uproszczony	04.25.2022	bezterminowo	Tak	250	1
WKS	Dopuszczalna liczba urządzeń wykorzystujących magazyn LocalStorage	04.25.2022	bezterminowo	Tak	30	0
ZPL_BARCODE	Dopuszczalna liczba drukarek wykorzystujących skrypty ZPL	04.25.2022	bezterminowo	Tak	100	0

Licencje nAxiom

Na stronie `Licencjonowanie` są wyświetlane informacje dotyczące bieżącej licencji w formie listy wpisów dotyczących obszaru objętego licencją, liczby aktywnych licencji w danym obszarze oraz liczby wykorzystanych licencji w danym obszarze. Dodatkowo podana jest data wydania i data wygaśnięcia licencji w danym obszarze.

Licencjonowane obszary są oznaczone następującymi kodami:

- *BASE*: licencja bazowa na środowisko, pozwalająca na korzystanie z platformy w modelu trzech środowisk (deweloperskie, testowe i produkcyjne).
- *LOAD_BAL*: licencja na korzystanie z zaawansowanych mechanizmów uruchamiania platformy na kilku serwerach w celu zwiększenia wydajności poprzez rozproszenie obliczeń.

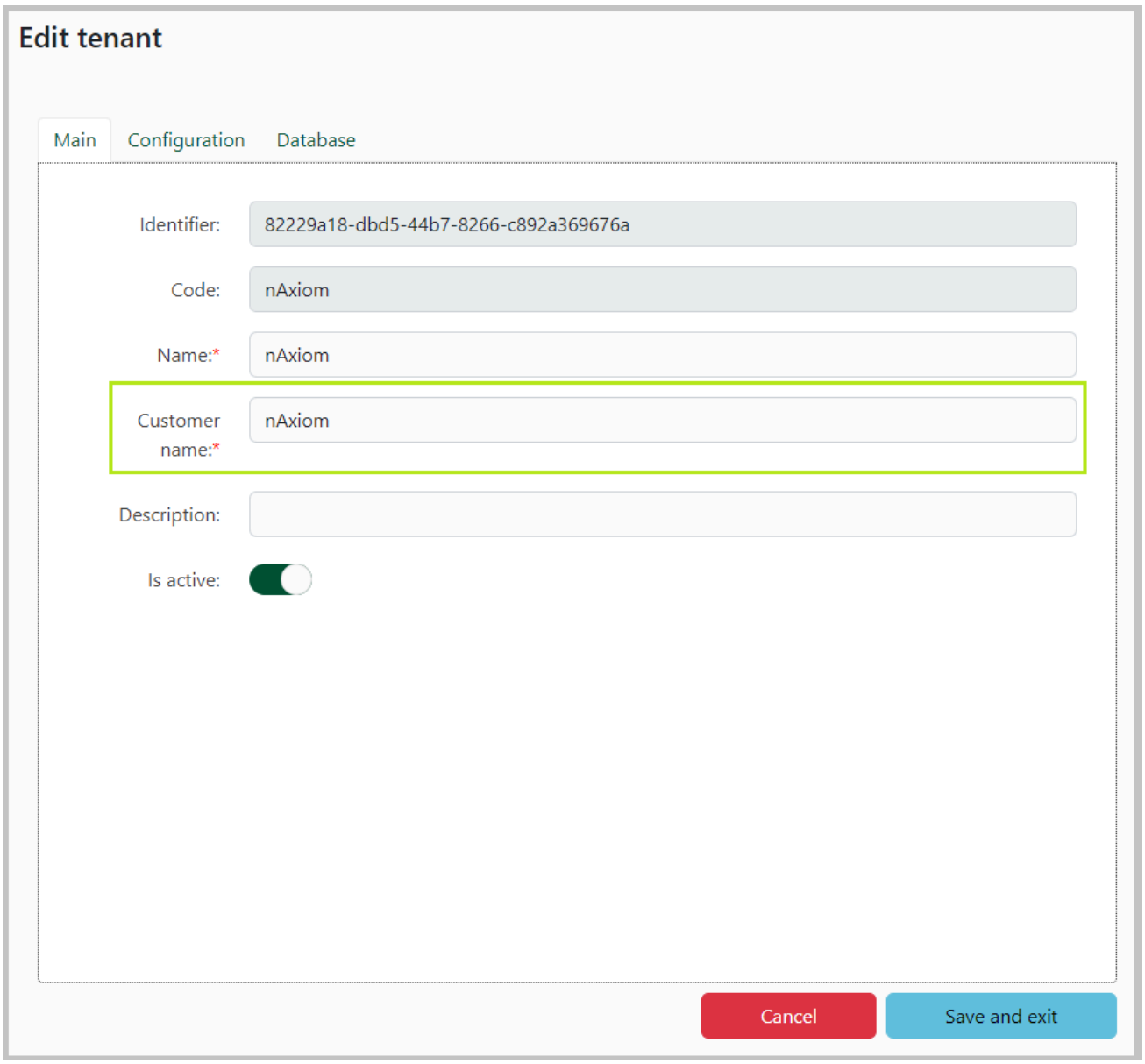
- *OCR*: licencja na moduł OCR (firmy ABBYY), który umożliwia pozyskiwanie warstwy tekstowej z dokumentów typu PDF, JPG, TIF pochodzących np. ze skanera. Warstwa tekstowa jest wykorzystywana przez system do wyszukiwania dokumentów po słowach znajdujących się w jego treści. Ta licencja umożliwia korzystanie z akcji rozpoznawania danych OCR i tworzenia PDF z warstwą tekstową. Patrz także Licencje Abby FR Engine .
- *OCR_PLUS*: rozszerzenie licencji OCR o moduł pozwalający na klasyfikowanie dokumentów i pobieranie z rozpoznanych typów dokumentów wybranych danych, których można później użyć np. do automatycznego wypełniania pól formularzy. Ta licencja umożliwia korzystanie z akcji aktualizacji przez OCR oraz daje dostęp do sekcji `OCR` w menu `Narzędzia`. Patrz także Licencje Abby FR Engine .
- *USER_CLIENT*: licencja na stanowisko użytkownika uprawnionego do pracy w aplikacji *FrontSPA* i korzystania ze zbudowanych w nAxiom aplikacji biznesowych. Odpowiada roli systemowej *Klient*. Taki użytkownik nie ma dostępu do aplikacji *AdminSPA*.
- *USER_SIMPLE_CLIENT*: obecnie ta licencja jest równoważna licencji *USER_CLIENT*. Odpowiada roli systemowej *Klient uproszczony*.
- *USER_CONSULTANT*: licencja na stanowisko użytkownika uprawnionego do pracy w aplikacji *AdminSPA* przeznaczonej do konfigurowania środowiska i budowy aplikacji biznesowych. Odpowiada roli systemowej *Konsultant*. Taki użytkownik nie ma dostępu do aplikacji *FrontSPA*.
- *WKS*: licencja pozwalająca na personalizacje stanowisk pracy i budowę np. info-kiosków.
- *ZPL_BARCODE*: licencja funkcjonalności bazujących na kodach kreskowych: sekcja skanowania kodu kreskowego w aplikacji mobilnej, akcja drukowania kodów kreskowych na drukarkach obsługujących kody ZPL.

WAŻNE

System porównuje liczbę licencji z liczbą ***aktywnych*** profili użytkowników. Jeśli liczba aktywnych profili będzie równa liczbie licencji, nie będzie można dodawać kolejnych profili. Dezaktywowanie profilu powoduje zwolnienie przypisanej mu licencji.

Nieco powyżej listy, z prawej strony jest wyświetlana *nazwa klienta*. Jest to ciąg znaków, dla którego wygenerowano licencję. Użytkownik musi go podać podczas instalacji nAxiom, aby można było wczytać licencję i korzystać z funkcjonalności nAxiom zgodnie z tą licencją.

Jeśli zajdzie potrzeba zmiany nazwy klienta, można to zrobić a ustawieniach tenanta w aplikacji *TenantAdminSPA* (`Tenants List` > `Edit tenant` > `Main` > `Customer name`). Po zmianie nazwy klienta użytkownik z rolą *Konsultant* musi zaimportować nowy plik licencji w aplikacji *AdminSPA*.



Edycja tenanta, nazwa klienta (klucz licencyjny)

LICENCJE ABBY FR ENGINE

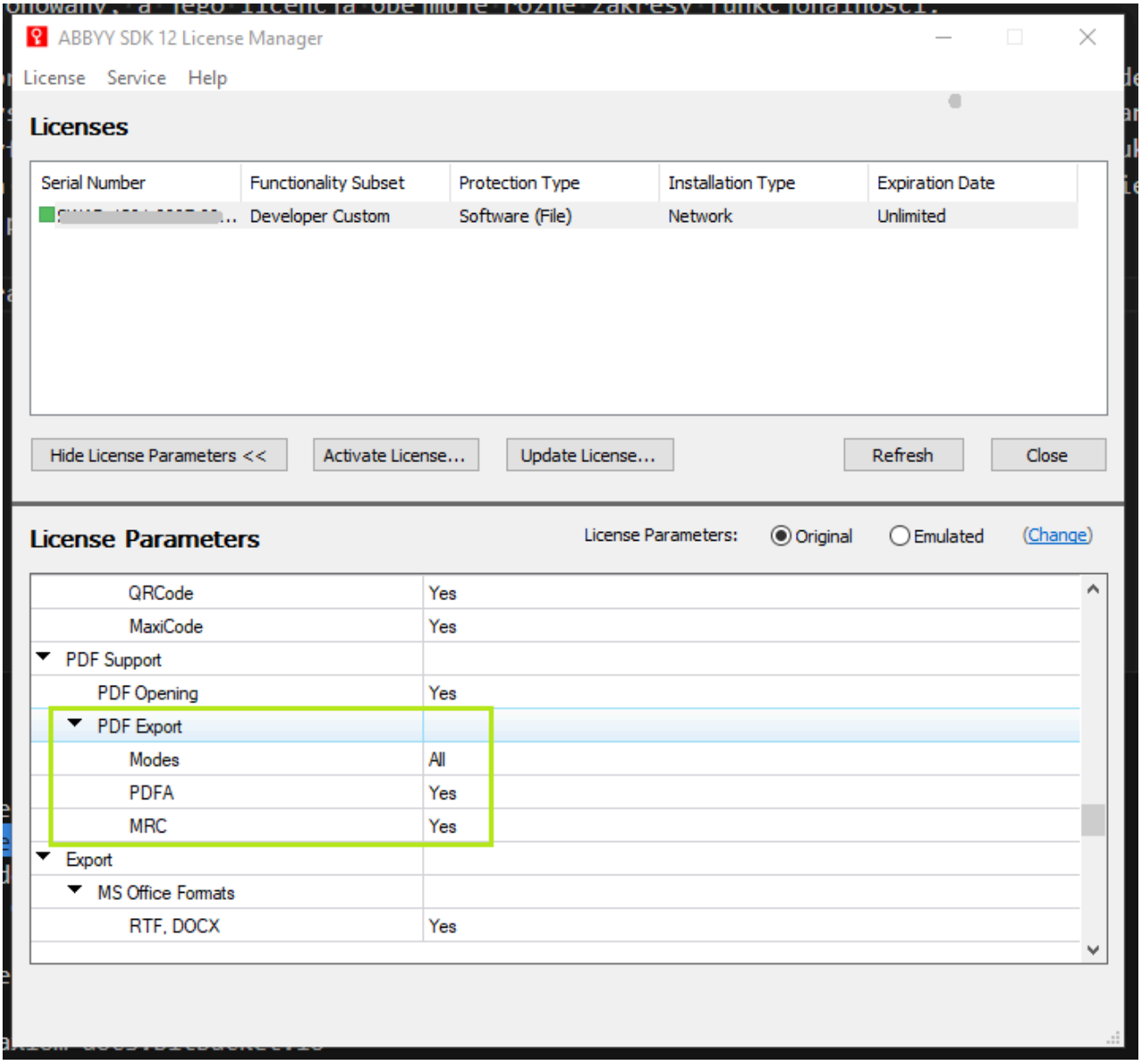
Licencje nAxiom *OCR* i *OCR_Plus* wymagają zainstalowania produktu ABBYY FineReader Engine w wersji 12. Ten produkt jest licencjonowany, a jego licencja obejmuje różne zakresy funkcjonalności.

Dla poprawnego działania akcji Utwórz PDF z warstwą tekstową licencja ABBYY FineReader Engine 12 musi obejmować moduł **PDF Export** ze wszystkimi jego składnikami. Brak tego modułu lub brak niektórych składników ograniczy możliwości nAxiom wyłącznie do zidentyfikowania warstwy tekstowej analizowanego pliku i dodania go do indeksu wyszukiwania pełnotekstowego. Nie będzie natomiast możliwa modyfikacja pliku PDF przez dodanie warstwy tekstowej, a tym samym nie będzie można przeszukiwać zawartości pliku PDF w sekcji podglądu ani po pobraniu go na dysk.

WSKAZÓWKA

Podczas instalacji konieczne jest podanie adresu IP lub nazwy serwera licencji, rodzaju protokołu do połączeń z tym serwerem oraz podanie parametru `Customer Project Id`. Te informacje można uzyskać od opiekuna technicznego platformy nAxiom w danej organizacji.

Aby sprawdzić zakres licencji ABBYY FineReader Engine 12, należy uruchomić program License Manager (`C:\Program Files\ABBYY SDK\12\FineReader Engine\Bin64\LicenseManager.exe`). Szczegółowe informacje dotyczące instalowania licencji i zarządzania nimi zawiera pomoc do tego programu.



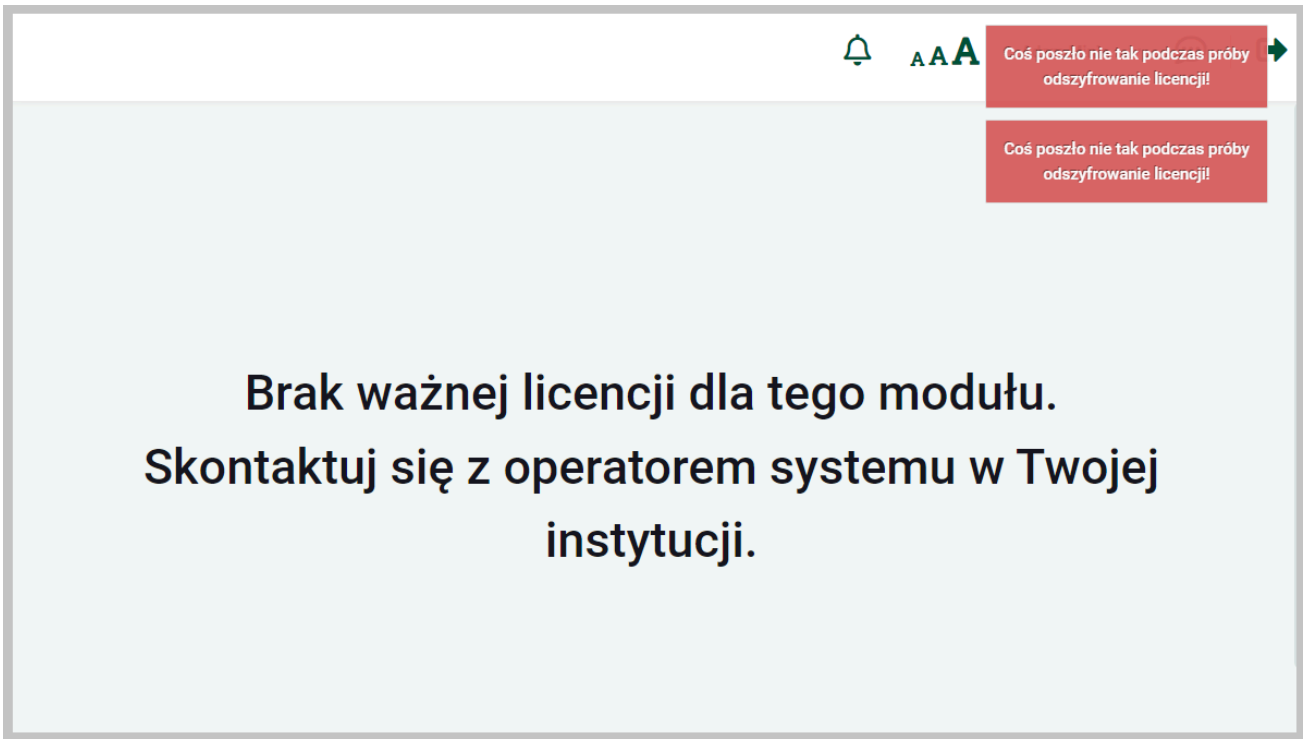
ABBY FineReader Engine License Manager

8.6.1. Import pliku licencji

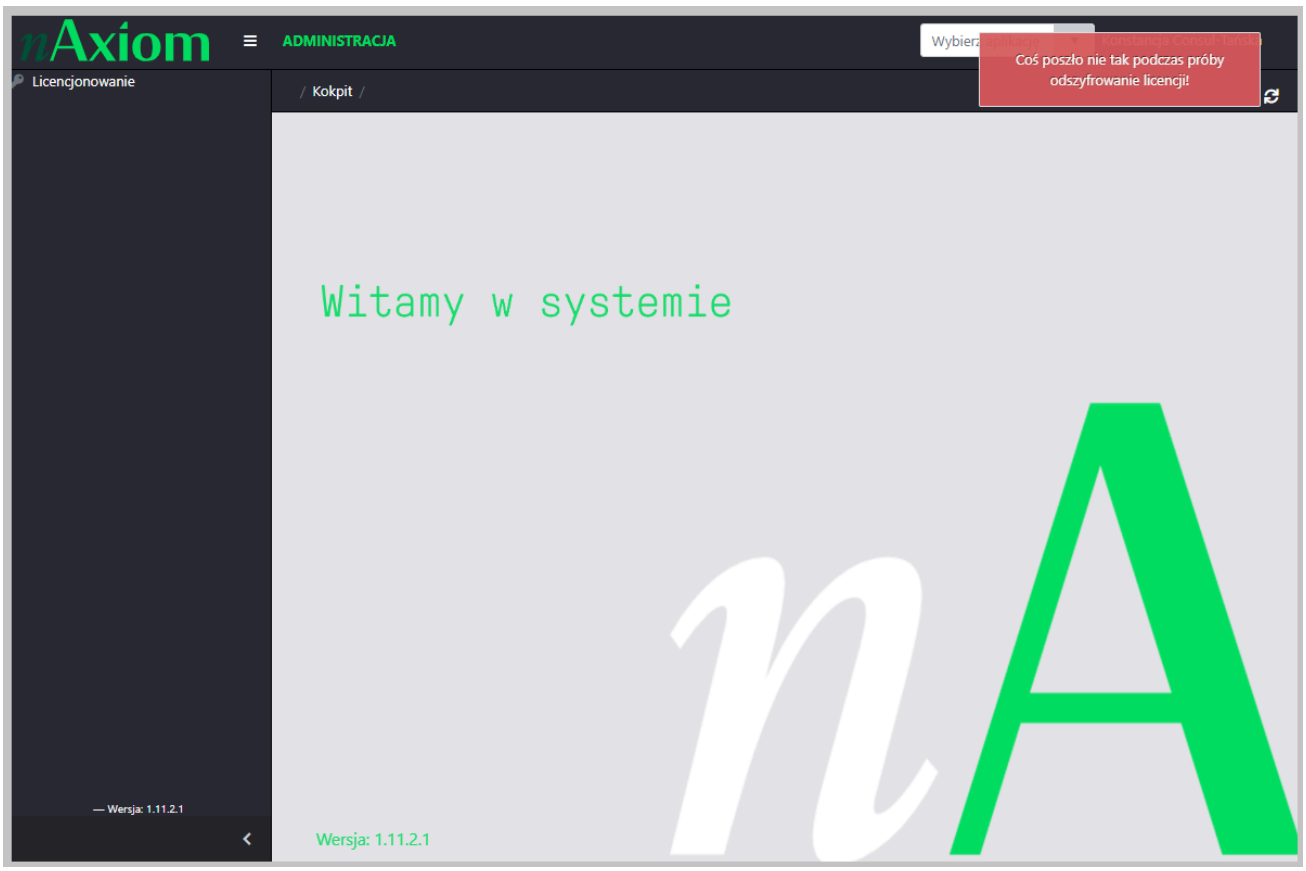
Podsumowanie:
Plik licencji to zaszyfrowany plik z rozszerzeniem `nlic` umożliwiający korzystanie z podstawowych i rozszerzonych funkcjonalności platformy nAxiom.

☛ Menu *AdminSPA* > `ADMINISTRACJA` > `Licencjonowanie` > `Importuj plik licencji`

Jeśli podczas instalacji plik instalacji nie był dostępny, albo jeśli dla danego tenanta administrator tenantów zmienił nazwę klienta (klucz licencyjny) lub wreszcie w przypadku kiedy licencja wygaśnie, użytkownicy logujący się do witryny naxiom zobaczą w przeglądarce strony, jak na ilustracjach poniżej.



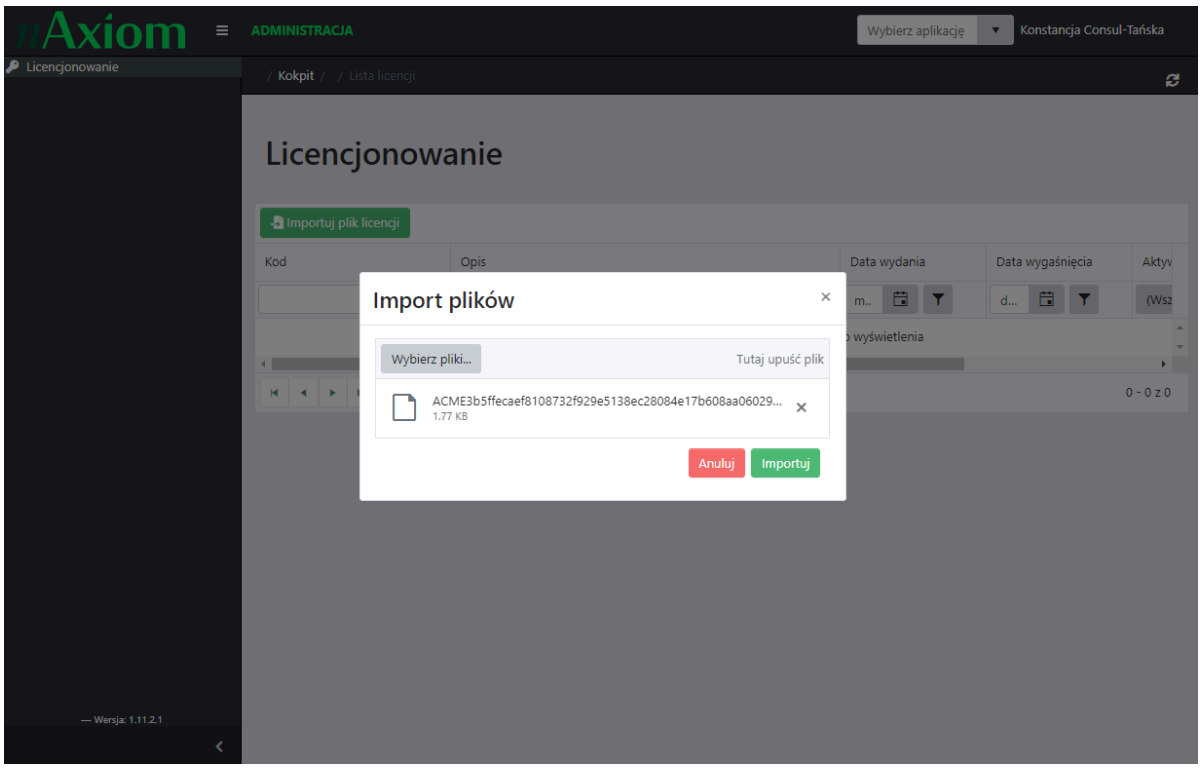
FrontSPA, brak licencji



AdminSPA, brak licencji

W menu *AdminSPA* jest tylko jedno dostępne polecenie, które umożliwia zaimportowanie pliku licencji. Aby zaimportować plik licencji, wykonaj następujące czynności:

1. Kliknij pozycję `Licencjonowanie` w sekcji `ADMINISTRACJA` menu głównego.
2. Kliknij przycisk `Importuj plik licencji`.
3. W wyświetlonym oknie dialogowym kliknij przycisk `Wybierz pliki` i wskaż plik w oknie wyboru pliku albo upuść plik licencji na opisany obszar.
4. Kliknij przycisk `Importuj`.



Import pliku licencji

WAŻNE
Plik licencji musi być zgodny z nazwą klienta podaną podczas instalacji witryny nAxiom (lub tworzenia nowego tenanta). W przeciwnym razie import takiego pliku nie powiedzie się.

8.7. MIGRATOR APLIKACJI

Podsumowanie:
Migrator aplikacji umożliwia przenoszenie aplikacji między środowiskami nAxiom. Obejmuje to zarówno import nowej aplikacji do środowiska docelowego, jak i aktualizację aplikacji w środowisku docelowym. Niezbędne wymagania dla pomyślnego zmigrowania aplikacji do innego środowiska to:
- takie same wersje nAxiom w obu środowiskach; a dokładniej takie same wersje bazy danych, patrz Import aplikacji,
- takie same schematy obiektów bazy danych w obu środowiskach.

W ogólności migracja polega na przeniesieniu danych definiujących aplikację z bazy danych środowiska źródłowego do bazy w środowisku docelowym. Jednak szczegóły tej operacji zależą od rodzaju obiektów.

☛ *AdminSPA* > `ADMINISTRACJA` > `Migrator aplikacji`

W TYM TEMACIE

- Wstęp
- Budowanie aplikacji pod kątem migracji
- Czynności do wykonania przed eksportem aplikacji
- Konfiguracje eksportu
- Porównywanie konfiguracji eksportu
- Podgląd danych w paczce
- Import aplikacji
- Czynności wykonywane po imporcie
- Szczególne przypadki aktualizacji aplikacji

WSTĘP

Migracja to proces, który odbywa sie w dwóch krokach:

- w środowisku źródłowym: wybór jednej lub kilku aplikacji i wyeksportowanie definicji obiektów przypisanych do tych aplikacji do pliku *.nax*.
- w środowisku docelowym wczytanie pliku *.nax* i zaimportowanie obiektów aplikacji do bazy danych.

Jeśli w danej aplikacji używane są obiekty przypisane do innej aplikacji, w definicji migrowanej aplikacji należy wskazać moduły macierzyste tych obiektów jako moduły powiązane (*AdminSPA* > `Lista aplikacji` > `Edytuj` > `Moduły` > `Dodaj powiązany moduł`).

Migrator identyfikuje aplikacje w pliku importu i w środowisku docelowym na podstawie atrybutu *RowGuid* definicji aplikacji. Dzięki temu możliwa jest aktualizacja aplikacji w środowisku docelowym także wtedy, kiedy jej nazwa (kod) i nazwa (kod) aplikacji importowanej są różne.

WSKAZÓWKA

Jeśli aplikacja *BaseApp* nie zawiera obiektów utworzonych przez użytkownika, jej migracja nie jest konieczna. Ta aplikacja zawsze istnieje w środowisku docelowym migracji.

Obiekty składające się na aplikację to:

- model danych (tabela): powiązanie z aplikacją następuje podczas generowania elementów aplikacji; możliwe jest także dowolne powiązanie innych tabel z modułem aplikacji na karcie `Moduły` w Kreatorze modeli danych.

> **WSKAZÓWKA**
> Migrator przenosi tylko definicję tabel użytkownika; dane nie są migrowane. Patrz także Szczególne przypadki aktualizacji aplikacji

- generowane automatycznie elementy aplikacji:
  - definicja dokumentu biznesowego, w tym typy dokumentów biznesowych, schemat autonumeracji, definicje statusów i diagram procesu
  - formularz
  - widok listy
- kategorie załączników
- definicje akcji, w tym zarówno globalne, jak i powiązane z formularzami
- definicje źródeł danych i słowników
- widżety, kontrolki, sekcje i podsekcje menu, szablony (Word, Excel, raporty, e-mail)
- monitory e-mail, zadania cykliczne
- tłumaczenia
- role biznesowe

W przypadku kategorii załączników, jeśli importowana kategoria ma przypisane jedno z repozytoriów systemowych, to przypisanie zawsze nadpisze przypisanie w aplikacji docelowej. Jeśli natomiast importowana kategoria ma przypisane repozytorium niestandardowe, przypisanie w aplikacji docelowej nie zmieni się. W przypadku importu nowej kategorii z repozytorium niestandardowym zostanie utworzona kategoria z przypisanym domyślnym repozytorium załączników.

> **OSTRZEŻENIE**
> Migracja nie obejmuje stylów ani zestawów rozpoznawanych danych i szablonów OCR. Ponadto nie są migrowane repozytoria załączników.

Ponadto migrator przenosi powiązane z tabelami danych biznesowych obiekty bazodanowe wymagane do poprawnego działania aplikacji w nowym środowisku:

- *Constraints* — w nAxiom definiowane na karcie `Relacje` dla wybranej tabeli w kreatorze modeli danych,
- *Indexes* — definiowane w bazie danych, niedostępne w nAxiom,
- *Triggers* — definiowane w nAxiom podczas definiowania audytów na poszczególnych kolumnach,

- *Views* — funkcje tabelaryczne tworzone zarówno w nAxiom jak i w bazie danych.

BUDOWANIE APLIKACJI POD KĄTEM MIGRACJI

Należy zwrócić uwagę na odwołania do obiektów aplikacji w oparciu o ustalone wartości identyfikatora. W przypadku tabel z kolumną klucza głównego o wartościach generowanych automatycznie (najczęściej kolumna o nazwie *Id* i typie *int*), po migracji do innego środowiska wartości w tej kolumnie mogą być inne. Spowoduje, to że wszelkie odwołania do obiektów reprezentowanych przez te rekordy za pomocą konkretnej wartości *Id* przestaną działać.

Aby zapobiec takim problemom, należy konstruować odwołania np. do *kodu* danego obiektu lub jego identyfikatora typu *uniqueidentifier*. W niektórych przypadkach można także użyć funkcjonalności *smart numbers* opisanej w temacie Smart numbers .

CZYNNOŚCI DO WYKONANIA PRZED EKSPORTEM APLIKACJI

Aby zapewnić powodzenie migracji aplikacji, zaleca się sprawdzenie kilku podstawowych kwestii.

1. Czy zamiast odwołań do identyfikatorów obiektów (np. statusów, typów biznesowych, słowników) używane są referencje *smart numbers* lub identyfikatory innego typu. Dotyczy to w szczególności akcji, zależności na listach, konfiguracjach list, zależności kontrolek.
2. Czy wszystkie tabele, które mają być migrowane, są przypisane do modułów aplikacji (`Kreator modeli danych`, karta `Moduły`); dotyczy to przede wszystkim tabel pomocniczych oraz tych tabel modeli standardowych i uproszczonych, których dokumenty biznesowe nie są migrowane.
3. Czy definicje dokumentów biznesowych, formularze i listy są powiązane z odpowiednimi aplikacjami i modułami. Migrator uwzględni tylko obiekty powiązane z modułami włączonymi do konfiguracji eksportu. W przypadku korzystania z obiektów powiązanych z innymi aplikacjami, należy powiązać moduł macierzysty takiego obiektu z modułem migrowanej aplikacji, zgodnie z opisem powyżej .

KONFIGURACJE EKSPORTU

Aby wyeksportować aplikacje z bieżącego środowiska nAxiom, wykonaj poniższe kroki:

1. Wyświetl stronę `Migrator aplikacji` (*AdminSPA* > `ADMINISTRACJA` > `Migrator aplikacji`).
2. Kliknij przycisk `Eksportuj` i w wyświetlonym oknie wpisz nazwę eksportowanej

konfiguracji, a następnie zaznacz pola wyboru obok aplikacji, które chcesz eksportować. Podana nazwa zostanie użyta w nazwie pliku konfiguracji. Dla każdej aplikacji można włączyć przełącznik w kolumnie `Eksportuj uprawnienia PBA`. Spowoduje to, że wraz z rolami przypisanymi do danej aplikacji zostaną wyeksportowane uprawnienia PBA przypisane tej roli we FrontSPA oraz w AdminSPA.

3. Kliknij przycisk `Eksportuj`. Zdefiniowana konfiguracja eksportu zostanie wyświetlona na liście.

Dla każdej konfiguracji eksportu w ostatniej kolumnie tabeli są dostępne przyciski akcji:

- `Importuj do platformy`: uruchamia kreatora importu aplikacji z danej konfiguracji; patrz sekcja Import aplikacji
- `Właściwości`: wyświetla listę aplikacji w konfiguracji
- `Pobierz`: pobiera konfigurację w postaci pliku archiwum z rozszerzeniem *nax* do domyślnego folderu pobierania; nazwa pliku ma postać *nAxiom-NazwaKonfiguracji-data-godzina*
- `Usuń`: usuwa konfigurację z listy

PORÓWNYWANIE KONFIGURACJI EKSPORTU

Na stronie `Migrator aplikacji` dostępne są przyciski umożliwiające porównanie dwóch konfiguracji eksportu oraz porównanie wybranej konfiguracji z bieżącym środowiskiem.

Aby porównać dwie konfiguracje, zaznacz je na liście i kliknij przycisk `Porównaj dwie konfiguracje`. Zostanie wyświetlona lista z nazwami tabel systemowych, w których zapisywane sa elementy aplikacji, oraz w kolejnych kolumnach liczby nowych wpisów, modyfikacji i usunięć. Każdą tabelę można rozwinąć, aby wyświetlić listę obiektów, które zostały zmienione, a te z kolei można rozwinąć do poziomu atrybutów. Dla szczegółów zmiany wyświetlany jest rodzaj modyfikacji (`DELETE`, `INSERT`, `UPDATE`) oraz dane oryginalne i dane importowane.

Z prawej strony nagłówka listy zmian są dostępne dwa przyciski:

- `Schowaj różnice`/`Pokaż różnice`: ukrywa/wyświetla listę zmian
- `Pokaż szczegóły`/`Ukryj szczegóły`: rozwija/zwija wszystkie pozycje na liście zmian

Aby porównać wybraną konfigurację z bieżącym środowiskiem, zaznacz konfigurację na liście i kliknij przycisk `Porównaj z obecną konfiguracją`. Zostanie wyświetlona lista zmian identyczna z opisaną powyżej. W tym przypadku konfiguracja bieżąca jest traktowana

jako pierwotna (bez daty/godziny), a konfiguracja zaznaczona na liście jako nowa konfiguracja.

Porównywanie konfiguracji odbywa się także podczas importu aplikacji, ale możne je wówczas pominąć.

PODGLĄD DANYCH W PACZCE

Aby wyświetlić wykaz obiektów w danej konfiguracji, należy ją zaznaczyć na liście i kliknąć przycisk `Podgląd danych w paczce`. Kliknięcie dowolnej pozycji w kolumnie `Nazwa` powoduje wyświetlenie listy obiektów danej kategorii i ich atrybutów.

IMPORT APLIKACJI

OSTRZEŻENIE
Przed importem aplikacji należy wykonać kopię zapasową bazy danych na środowisku docelowym, aby niepowodzenie operacji nie doprowadziło do utraty danych.

Aby zapewnić powodzenie importu, należy wziąć pod uwagę następujące kwestie:

- Migrator nie tworzy w docelowej bazie danych schematów bazy danych. Jeśli importowana aplikacja korzysta z niestandardowych schematów bazy danych, przed importem konsultant musi utworzyć je w docelowej bazie danych. W przeciwnym razie import nie powiedzie się.
- Środowisko źródłowe importowanej aplikacji musi być zgodne ze środowiskiem docelowym na poziomie bazy danych. Zgodne bazy danych to takie, w których wykonywane są te same migracje Entity Framework, czyli ostatni rekord w tabeli *core.__EFMigrationsHistory* ma tę samą wartość w kolumnie *MigrationId* w obu bazach. W szczególnych przypadkach może być możliwe migrowanie aplikacji między różnymi wersjami nAxiom. Odpowiednie informacje są publikowane w Informacjach o wersji.

Podczas importu konfiguracji dokonywanych jest szereg zmian na bazie danych. Zaleca się, aby w tym czasie ze środowiska docelowego nie korzystali inni użytkownicy, ani w aplikacji *FrontSPA*, ani w aplikacji *AdminSPA*.

Aby zaimportować konfigurację do bieżącego środowiska, kliknij przycisk `Importuj do platformy` w wierszu tej konfiguracji na ekranie `Migrator aplikacji`. Zostanie wyświetlone okno dialogowe z pytaniem o porównanie konfiguracji importowanej aplikacji z konfiguracją aplikacji w docelowej bazie danych.

W przypadku bardziej złożonych aplikacji ta operacja może zajmować dużo czasu, dlatego można ją pominąć, klikając przycisk `Rozpocznij bez porównania`. Po kliknięciu przycisku `Porównaj` zostanie wyświetlona lista tabel systemowych, w których w związku z importem zostaną zmienione rekordy. Rozwijając poszczególne pozycje, można szczegółowo analizować zmiany w wyniku importu. Kliknij przycisk `Rozpocznij import` u dołu. Przycisk `Anuluj` spowoduje przerwanie procesu.

Operacja importu jest realizowana w trybie asynchronicznym, to jest po rozpoczęciu importu użytkownik może przejść do innych zadań lub wylogować się.

UWAGA

1. W trakcie operacji importu zainicjowanej z poziomu AdminSPA nie można uruchomić kolejnego importu ani wyeksportować konfiguracji.
2. Aby odświeżyć status importu, należy kliknąć przycisk odświeżania listy widoczny na ekranie migratora.
3. W przypadku importowania aplikacji przez Public API, można zlecić kilka operacji importu, które zostaną dodane do kolejki.

W razie wystąpienia błędów operacja importu jest przerywana i żadne zmiany nie są dokonywane w bazie danych. Aby zaimportować konfigurację pomimo błędów importu, włącz przełącznik `Pomiń błędy importu` w ustawieniach środowiska (`ADMINISTRACJA` > `Ustawienia systemu` > `Ogólne`).

WAŻNE

W przypadku decyzji o imporcie z pominięciem błędów konieczne jest wyświetlenie dziennika błędów (`ADMINISTRACJA` > `Logi środowiska i bieżącego tenanta`) i rozwiązanie wszystkich problemów napotkanych podczas importu.

WSKAZÓWKA

Po zaimportowaniu aplikacji następuje automatyczne wyczyszczenie pamięci podręcznej dla wszystkich tenantów.

CZYNNOŚCI WYKONYWANE PO IMPORCIE

1. Ustawić ścieżki w repozytoriach załączników. W przeciwnym razie utracone zostaną informacje o kategoriach załączników, zgodnie z uwagą powyżej; `DANE` > `Repozytoria załączników`.
2. Zmigrować dane biznesowe, których atrybuty uległy zmianie na skutek importu nowej definicji aplikacji; szczegóły opisano ponizej . Ta czynność dotyczy także tabel

pomocniczych.

UWAGA

- Migrator nie przenosi ustawień e-mail. W celu korzystania z wysyłki poczty może być konieczne skonfigurowanie ustawień e-mail (`ADMINISTRACJA` > `Ustawienia systemu` > `Email`). Następnie należy zrestartować komponent *taskservice* (Menedżer usług informacyjnych > `Pule aplikacji` > *naxiom.nazwa_witryny.taskservice* > `Odtwarzanie`).
- Uprawnienia ról do menu podlegają migracji tylko w obrębie migrowanej aplikacji. Uprawnienia roli do menu z innych aplikacji zostaną zmigrowane tylko wtedy, gdy w witrynie docelowej będą te aplikacje.

SZCZEGÓLNE PRZYPADKI AKTUALIZACJI APLIKACJI

Migrator nAxiom obsługuje trzy rodzaje zmian w importowanej aplikacji, które w przypadku aktualizacji aplikacji wpływają na dane biznesowe w środowisku docelowym.

Pierwsza z nich dotyczy zmiany definicji kolumny w tabeli dokumentu biznesowego. Jeśli w środowisku docelowym ta kolumna zawiera dane, migrator zmieni jej nazwę na *StaraNazwa_backup<timestamp>*, a następnie utworzy nową kolumnę zgodną z nową specyfikacją. Po zakończeniu importu konieczne będzie przeniesienie danych z oryginalnej kolumny do nowej i usunięcie starej kolumny.

UWAGA
Jeśli w ramach aktualizacji aplikacji została usunięta kolumna w tabeli użytkownika, ta kolumna nie zostanie usunięta z aplikacji w środowisku docelowym, bez względu na to czy zawiera dane.

Druga funkcja ma na celu zabezpieczenie danych biznesowych w sytuacji, kiedy w migrowanej aplikacji usunięto status, typ biznesowy lub przejście na diagramie procesów, ale w docelowej bazie danych istnieją dokumenty, które mają ustawione te atrybuty. Na potrzeby obsługi takiego przypadku utworzono specjalną definicję dokumentu biznesowego — *Dokument biznesowy usunięty przez migrator*, z którą powiązano specjalny status i typ dokumentu biznesowego:

- *Status biznesowy usunięty przez migrator* (kod: *ErasedByAppMigration*)
- *Typ biznesowy usunięty przez migrator* (kod: *ErasedByAppMigration*)

Te wartości są używane do zastąpienia usuniętego statusu i/lub typu biznesowego w instancjach dokumentów w środowisku docelowym. Po migracji konsultant musi je zastąpić odpowiednimi wartościami.

Trzecia funkcja obsługuje przypadek usunięcia przejścia na diagramie procesu w importowanej aplikacji. Powoduje ona zastąpienie wszystkich wystąpień identyfikatora tego przejścia w kolumnie *Transition* w tabeli dokumentu biznesowego wartością *NULL*.

Powiązane tematy:

- Smart numbers

9. Budowanie i debugowanie aplikacji nAxiom

W tej sekcji opisano różne funkcjonalności, które łączą funkcje dostępne w poleceniach z różnych sekcji menu aplikacji *AdminSPA*.

9.1. WYSZUKIWANIE PEŁNOTEKSTOWE W NAXIOM

Podsumowanie:
W nAxiom dostępne są podstawowe funkcjonalności umożliwiające korzystanie z funkcji wyszukiwania pełnotekstowego w plikach załączników, w polach dokumentów, w komentarzach do dokumentów i w plikach poddanych przetwarzaniu OCR. W przypadku pól dokumentów konieczne ustawienie atrybutu wyszukiwania pełnotekstowego we właściwościach kolumny w kreatorze modeli danych.

nAxiom używa silnika wyszukiwania Elasticsearch.

W TYM TEMACIE

- Wdrożenie Elasticsearch
- Konfiguracja w pliku appsettings.json
- Konfiguracja w TenantsAdminSPA
- Korzystanie z wyszukiwarki

WDROŻENIE ELASTICSEARCH

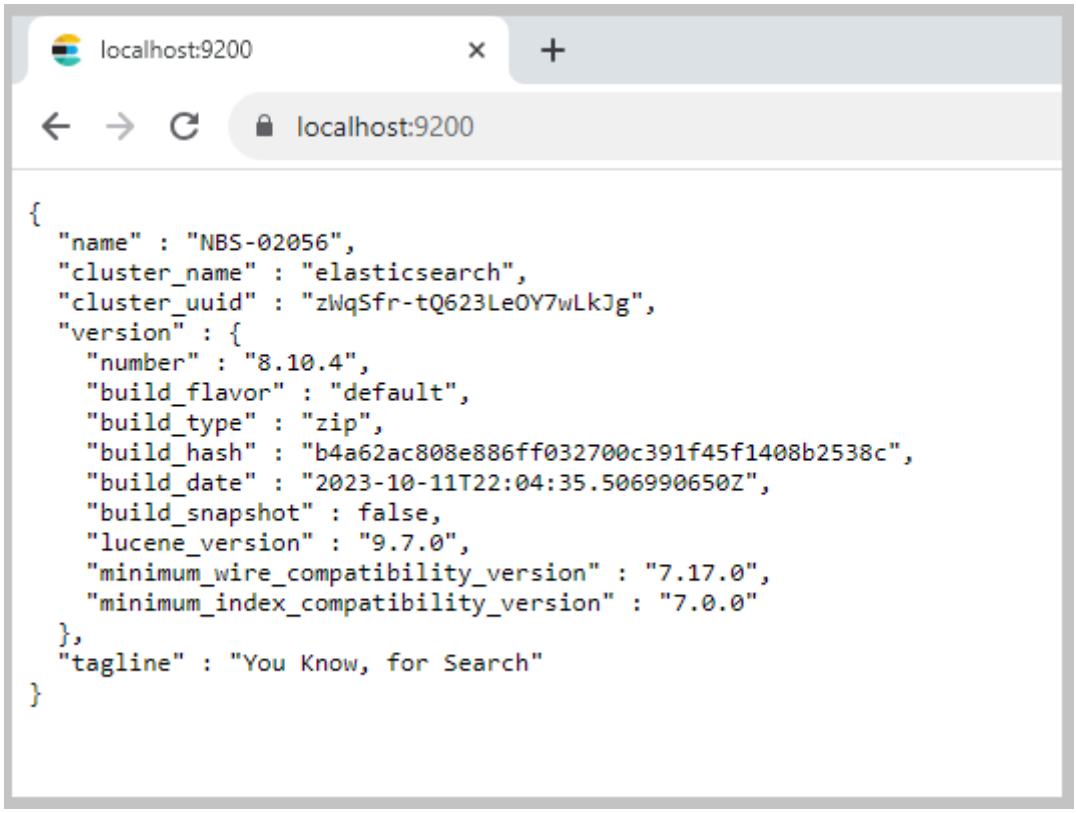
Witrynę nAxiom można podłączyć do lokalnego lub zdalnego wdrożenia ElasticSearch. Nie jest obsługiwana współpraca z wdrożeniem chmurowym Elasticsearch Service.

Na potrzeby testów można uruchomić lokalne wdrożenie Elastic, pobierając pakiet ZIP ze strony producenta https://www.elastic.co/downloads/elasticsearch, wypakowując folder produktu i uruchamiając plik *elasticsearch.bat* z podkatalogu *bin*.

Od wersji 8 silnik Elasticsearch domyślnie uruchamia się z włączonymi zabezpieczeniami, umożliwiając szyfrowaną komunikację z aplikacją kliencką. Podczas pierwszego uruchomienia oprogramowanie generuje certyfikat CA (podkatalog *config/certs*) oraz wyświetla hasło dla domyślnego użytkownika *elastic*. Certyfikat należy dodać do magazynu *Zaufane główne urzędy certyfikacji*, a nazwę użytkownika i hasło należy zapisać i podać w konfiguracji nAxiom.

> **WSKAZÓWKA**
> Do pierwszego uruchomienia programu *elasticsearch.bat* nie należy używać wiersza poleceń z podniesionymi uprawnieniami (`Uruchom jako administrator`), ponieważ spowoduje to konflikt uprawnień do podfolderów tworzonych podczas pierwszego uruchomienia.

W przypadku wdrożenia lokalnego silnik Elasticsearch jest dostępny pod adresem *https://localhost:9200*. Po wpisaniu tego adresu w przeglądarce wyświetlane jest okno logowania, a po podaniu poświadczeń, dane w formacie JSON z podstawowymi informacjami o działającej instancji silnika Elasticsearch.



KONFIGURACJA W PLIKU APPSETTINGS.JSON

Jednym z warunków włączenia wyszukiwania pełnotekstowego w witrynie nAxiom jest skonfigurowanie parametrów połączenia z instancją silnika Elasticsearch, która ma być używana. Wymaga to edycji pliku *appsettings.json* dla serwisu *api*. Odpowiedni fragment wygląda następująco:

```
"Modules": {
  "FullTextSearch": {
    "Enabled": true,
    "MaxQueryResults": 50,
    "ElasticSearch": {
      "Url": "https://localhost:9200",
      "UserName": "elastic",
      "Password": "jZEId2R6lR0r336uYbgb",
      "IndexPrefix": "acme"
    }
  },
```

W celu skonfigurowania wyszukiwania należy określić następujące parametry:

- *Enabled*: (*true/false*) włącza lub wyłącza funkcjonalność.
- *MaxQueryResults*: maksymalna liczba zwracanych wyników wyszukiwania.
- *Url*: adres URL używanej instancji silnika Elasticsearch.
- *UserName*: nazwa użytkownika używana w komunikacji z silnikiem ES (domyślnie *elastic*).
- *Password*: hasło używane w komunikacji z silnikiem ES (generowane automatycznie podczas pierwszego uruchomienia ES).
- *IndexPrefix*: przedrostek dodawany do indeksów tworzonych w silniku ES dla danej witryny nAxiom, oprócz niego dodawany jest także przedrostek tenanta (kod tenanta); każda witryna nAxiom tworzy dwa indeksy w silniku ES: *IndexPrefix-TenantCode-document-attachmens* oraz *IndexPrefix-TenantCode-document-fields*

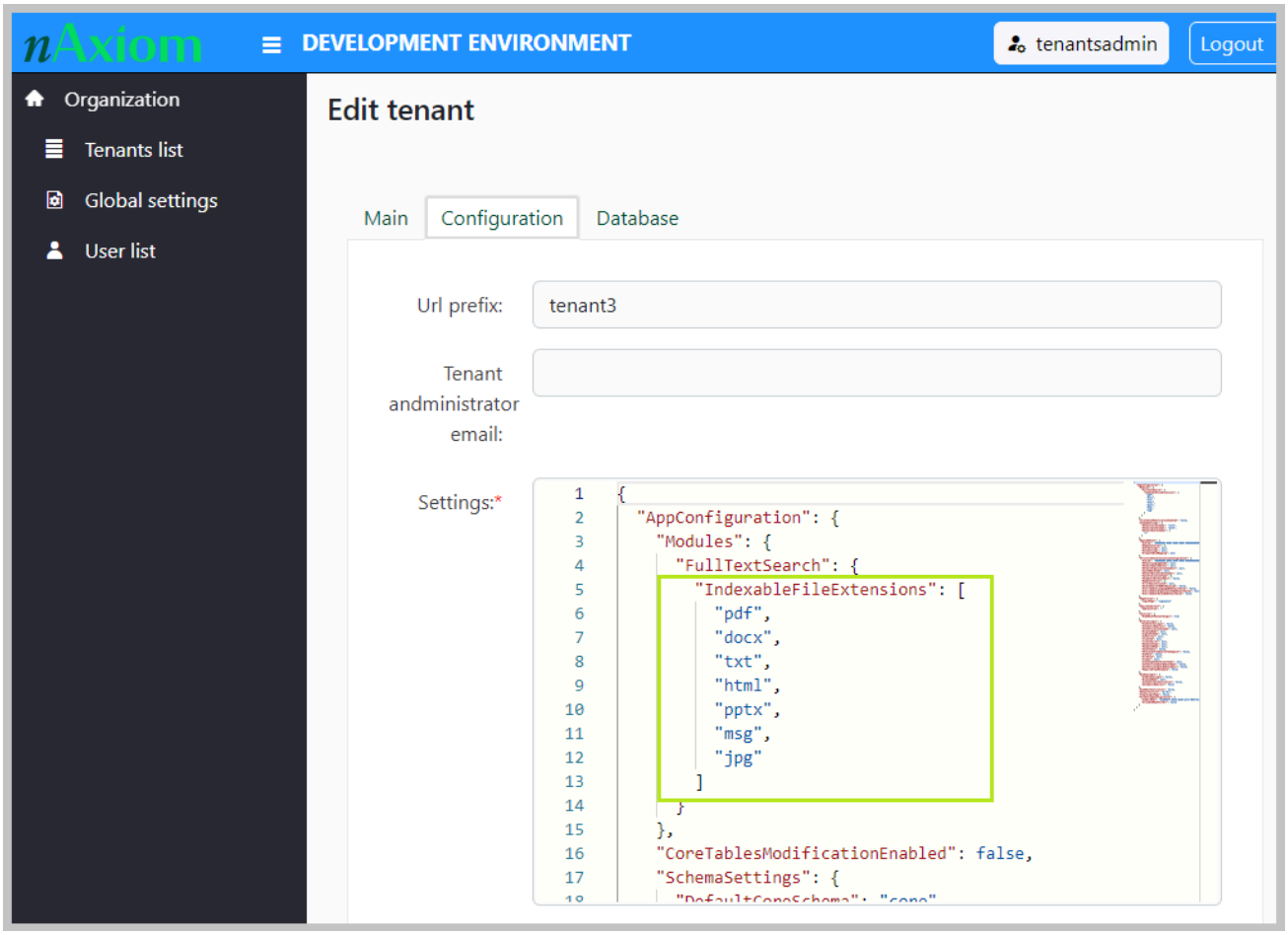
Indeksowanie może także obejmować dokumenty i załączniki dodawane przy użyciu żądań *Public API*, pod warunkiem dodania identycznych wpisów w pliku *appsettings.json* tego serwisu.

KONFIGURACJA W TENANTSADMINSPA

Dodatkowe kroki należy wykonać w konfiguracji poszczególnych tenantów. Obejmują one:

- włączenie funkcji wyszukiwania w ustawieniach systemu (*AdminSPA* > `Ustawienia systemu` > `Ogólne` > `Włącz wyszukiwarkę`),
- określenie rozszerzeń nazw plików załączników, których treść ma być indeksowana.

Drugi krok wykonuje się w aplikacji *TenantsAdminSPA* na karcie `Configuration` w ustawieniach tenanta. Rozszerzenia określa się jako elementy tablicy *IndexableFileExtensions* w pliku JSON w polu `Settings`.



Indeksowane pliki załączników

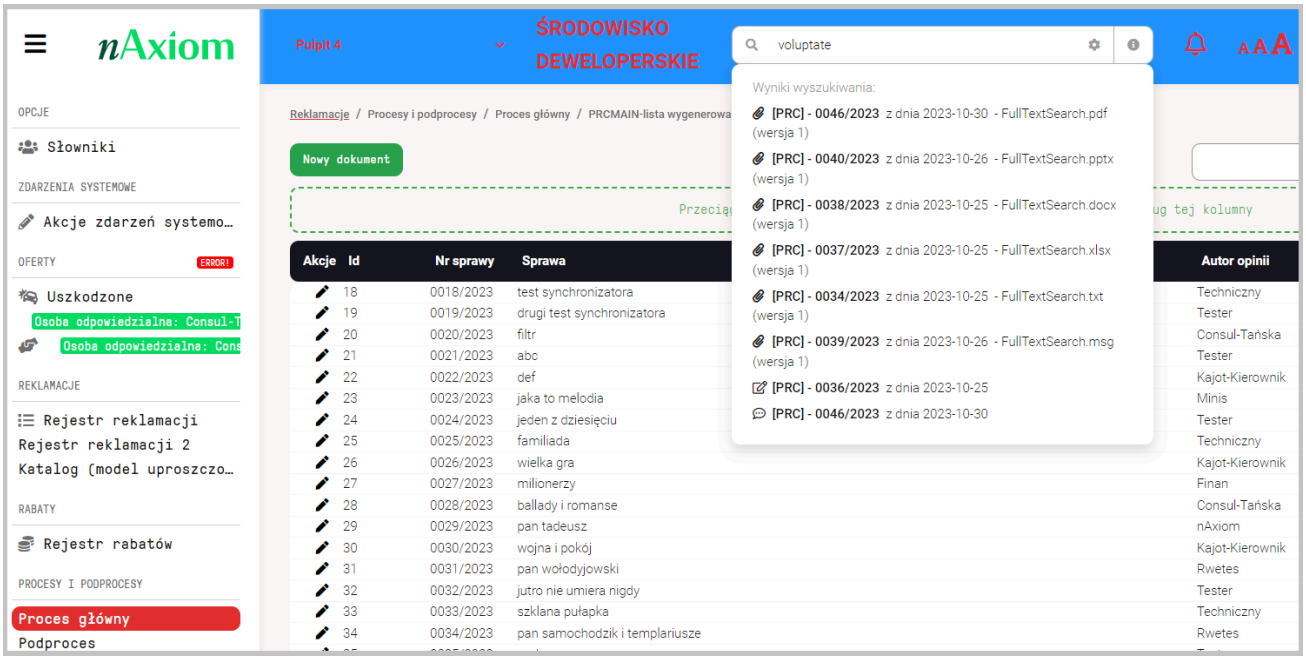
Silnik Elasticsearch do analizy zawartości plików wykorzystuje pakiet Apache Tika, który zawiera parsery umożliwiające wyodrębnienie treści i metadanych z wielu rodzajów dokumentów (patrz https://tika.apache.org/2.9.1/formats.html). Najważniejsze obsługiwane formaty to:

- Microsoft Office (zarówno starsze OLE2, jak i nowsze OOXML),
- OpenDocument Format (ODF, pakiet OpenOffice.org),
- PDF,
- RTF,
- TXT (dowolne kodowanie),

- HTML,
- XML,
- różne formaty wiadomości e-mail, między innymi MS Outlook PST i MSG.

W przypadku formatów opartych na języku znaczników, jak HTML czy XML, znaczniki nie są indeksowane. Ponadto pakiet Tika pozwala także wyodrębniać metadane z plików audio, wideo i obrazów — te funkcjonalności nie zostały na razie zaimplementowane w nAxiom.

Konfigurując rozszerzenia, należy pamiętać, że muszą one zostać dodane jako dozwolone rozszerzenia załączników w ustawieniach systemu lub w definicji kategorii załączników.



Wyniki wyszukiwania

KONFIGURACJA W TENANTSADMINSPA

Po włączeniu wyszukiwania w aplikacji *FrontSPA* na pasku nagłówka jest wyświetlane pole wyszukiwania. Domyślnie wyszukiwane są ścisłe dopasowania do dowolnego członu frazy wyszukiwania. Można używać znaków wieloznacznych: *?* zamiast dowolnego jednego znaku i * zamiast dowolnego ciągu znaków. Aby wyszukać wystąpienie całej frazy wyszukiwania, należy ująć ją w cudzysłowy (“”).

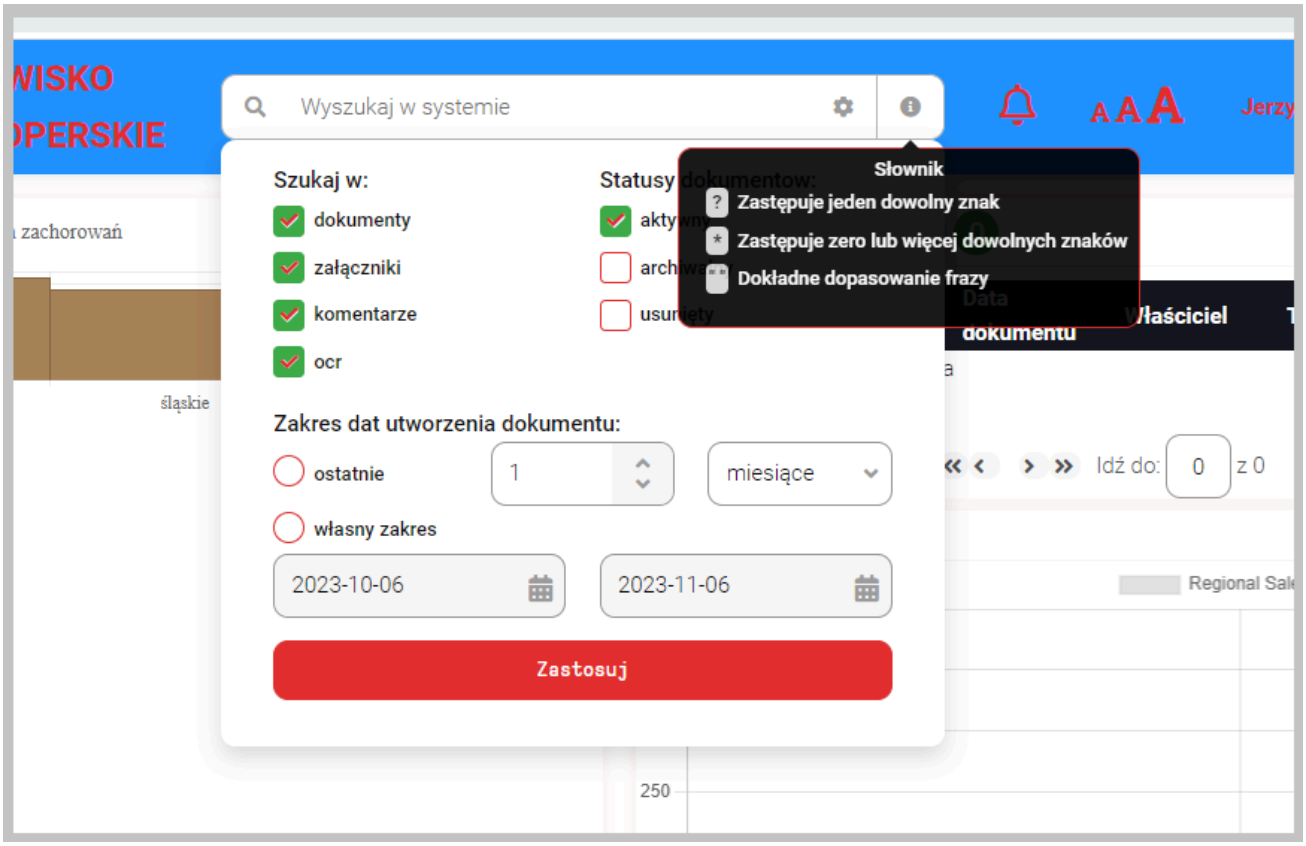
Po kliknięciu ikony ustawień wyszukiwania można określić zakres wyszukiwania pod względem rodzaju dokumentów:

- dokumenty (wartości w indeksowanych polach)
- załączniki
- komentarze
- rozpoznane pliki OCR

oraz statusu dokumentów:

- aktywne,
- archiwalne,
- usunięte.

Ponadto można określić zakres dat utworzenia przeszukiwanych dokumentów.



Ustawienia wyszukiwania

Powiązane tematy:

- Kreator modeli danych, okno Nowe pole modelu danych
- Ustawienia systemu, sekcja Ogólne (Włącz wyszukiwarkę)
- Ustawienia systemu, sekcja Załączniki
- Kategorie załączników

9.2. SMART NUMBERS

Podsumowanie:
Referencja typu *smart numbers* zwraca identyfikator obiektu na podstawie kodu tego obiektu. Funkcja jest dostępna w edytorze SQL i C# (akcja algorytm C#), a także w niektórych polach tekstowych, hybrydowych tekst/SQL, a nawet numerycznych (informacja o obsłudze jest wyświetlana *placeholder*). Funkcja działa w ten sposób, że po wpisaniu ciągu *{&* podpowiada kolejno kod aplikacji, oznaczenie obiektu aplikacji, kod modułu i kod obiektu, np.
`{&AppName.ObjectType.ModuleName.ObjectCode}`
W przypadku zmiennych aplikacji referencja zwraca wartość zmiennej, a w przypadku słowników zwykłych zwracana jest wartość pozycji słownikowej o kodzie wskazanym w referencji.

W polach tekstowych, hybrydowych tekst/SQL w trybie tekstowym oraz numerycznych można jedynie wkleić wyrażenie przygotowane wcześniej, np. w edytorze SQL.

Użycie składni smart numbers eliminuje odwołania do identyfikatorów, co umożliwia eksport obiektów aplikacji do innych środowisk, w których generowane automatycznie wartości identyfikatorów mogą być inne.

Składnia smart numbers obsługuje następujące rodzaje obiektów aplikacji nAxiom: aplikacje, akcje, zmienne aplikacji, role (biznesowe i systemowe), definicje dokumentów, statusy biznesowe, typy biznesowe, źródła danych, słowniki zwykłe i hierarchiczne, wartości ze słowników zwykłych oraz grupy interesariuszy.

W edytorze SQL (a także w edytorze kodu C# w definicji akcji C#) jest dostępna asysta przy tworzeniu referencji smart numbers. Aby z niej skorzystać, wpisz nawias klamrowy otwierający (*{*) i symbol ampersand (*&*) co spowoduje wyświetlenie listy aplikacji zdefiniowanych w systemie. W niektórych przypadkach jest dostępna pozycja *PARENT*, która oznacza aplikację macierzystą dla bieżącego formularza.

WSKAZÓWKA
Referencja w postaci *{&KodAplikacji}* zwraca identyfikator (*Id*) definicji aplikacji, którego można użyć np. w akcji Przełącz aplikację .

Asysta *smart numbers*; lista aplikacji

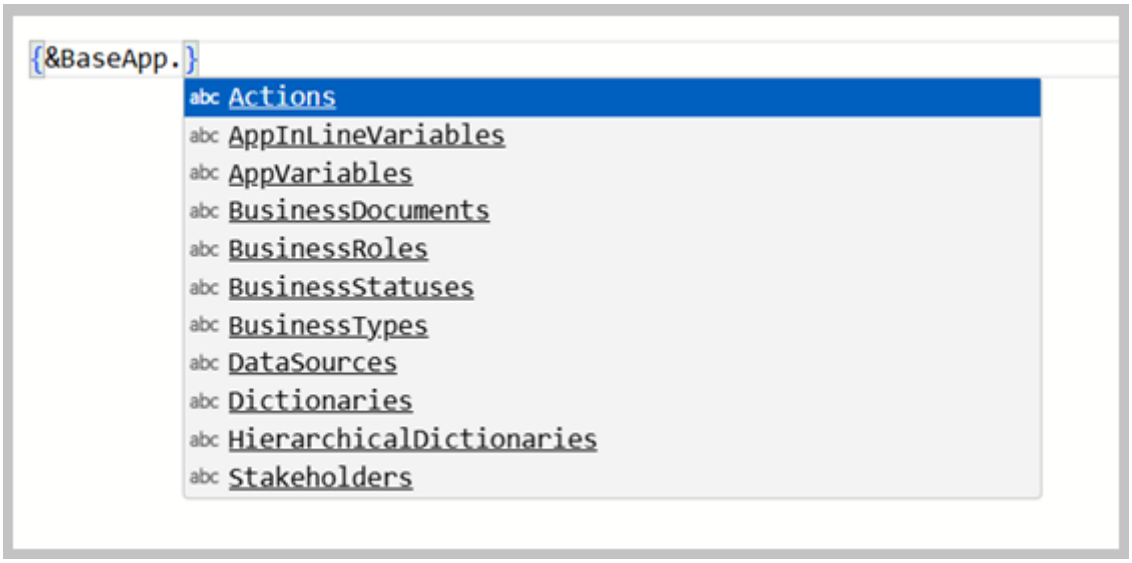
Po wybraniu aplikacji wpisz kropkę, aby wyświetlić listę obsługiwanych obiektów. Obecnie są to:

- akcje (*Actions*): następnie wybierz kod modułu i kod akcji
- zmienne środowiskowe (*AppVariables*, *AppInLineVariables*): następnie wybierz kod zmiennej

> **WAŻNE**
>
> - ***AppVariables***: odwołanie przeznaczone do użycia w zapytaniach SQL (np. w klauzuli WHERE jako wartość filtrująca); przekazuje wartość zmiennej jako parametr zgodnie z zadeklarowanym typem, dla typu tekstowego wartość zmiennej w zapytaniu jest renderowana w apostrofach
> - ***AppInlineVariables***: zwraca zadeklarowaną wartość zmiennej (jako tekst); wywoływane w ten sposób zmienne środowiskowe mogą na przykład być używane do podstawiania różnych nazw tabel w klauzuli *FROM* lub nazw kolumn w klauzuli *WHERE* w zapytaniach SQL, a także w akcjach C#, gdzie wartości typu tekstowego należy ująć w podwójne apostrofy (“).

- kategorie załączników (*AttachmentCategories*); następnie wybierz kod definicji dokumentu i kod kategorii
- definicje dokumentów (*BusinessDocuments*); następnie wybierz kod modułu i kod definicji dokumentu
- role biznesowe (*BusinessRoles*): następnie wybierz kod modułu i kod roli
- statusy dokumentów (*BusinessStatuses*): następnie wybierz kod definicji dokumentu i kod statusu
- typy biznesowe (*BusinessTypes*): następnie wybierz kod definicji dokumentu i kod typu biznesowego
- źródła danych (*DataSources*): następnie wybierz kod modułu i kod źródła danych

- słowniki zwykłe (*Dictionaries*): następnie wybierz kod słownika
- wartości pozycji w słownikach zwykłych (*Dictionaries*): następnie wybierz kod słownika i kod wartości słownika
- słowniki hierarchiczne (*HierarchicalDictionaries*): następnie wybierz kod słownika
- grupy interesariuszy (*Stakeholders*): następnie wybierz kod definicji dokumentu i kod grupy interesariuszy



Asysta *smart numbers*; dostępne obiekty aplikacji

Przykładowa składnia reprezentująca wartość identyfikatora dla statusu *Odrzucono* to:

```
{&Reklamacje34.BusinessStatuses.rklRjs.rklRjs_odrzucono}
```

W momencie wywołania zapytania dynamicznie obliczana jest wartość identyfikatora wybranego obiektu lub pobierana wartość zmiennej.

> **WAŻNE**
> Aby składnia smart numbers działała poprawnie, kod w definicji obiektu nie może zawierać znaków diakrytycznych.

Od wersji nAxiom 1.15, w związku z integracją z brokerem komunikatów RabbitMQ, dodanie nowego elementu (np. roli, kategorii interesariuszy, pozycji słownika itp.) nie wymaga przelogowania konsultanta ani wyczyszczenia pamięci podręcznej w celu aktualizacji listy obiektów dostępnych w składni *smart numbers*. W wersjach wcześniejszych wymagane jest wykonanie jednej z tych czynności.

Informacja o dodaniu nowego elementu (np. roli, kategorii interesariuszy, pozycji słownika itp.) obsługiwanego przez funkcjonalność smart numbers jest propagowana przez broker komunikatów RabbitMQ do innych serwisów co powoduje natychmiastowe przeładowanie pamięci podręcznej.

Aby aplikacje nAxiom importowane do innej witryny działały poprawnie, konieczne jest zastąpienie wszystkich odwołań do konkretnej wartości identyfikatora składnią smart numbers. Poniżej wymieniono miejsca w aplikacji AdminSPA, w których można używać składni smart numbers. Ręczne wpisanie wyrażenia smart numbers w innych miejscach nie zostanie poprawnie przetworzone.

GDZIE UŻYWAĆ *SMART NUMBERS*

Składni smart numbers można używać:

- w każdym edytorze SQL
- w akcji algorytm C# (pole `Algorytm C#`)
- w akcji zapytania do webserwisów (pole `Ścieżka żądania`)
- w poniższych polach definicji źródła danych:
  - `Adres serwera`
  - `Open api url`
  - `Serwer autoryzacji`
  - `Nazwa bazy danych`
  - `Nazwa instancji`
  - `Port`
- w polach tekstowych, hybrydowych tekst/SQL i numerycznych, w których jest wyświetlana informacja o obsłudze smart numbers (jako „placeholder text")
- w definicji zmiennych aplikacji w polach wartości zmiennej

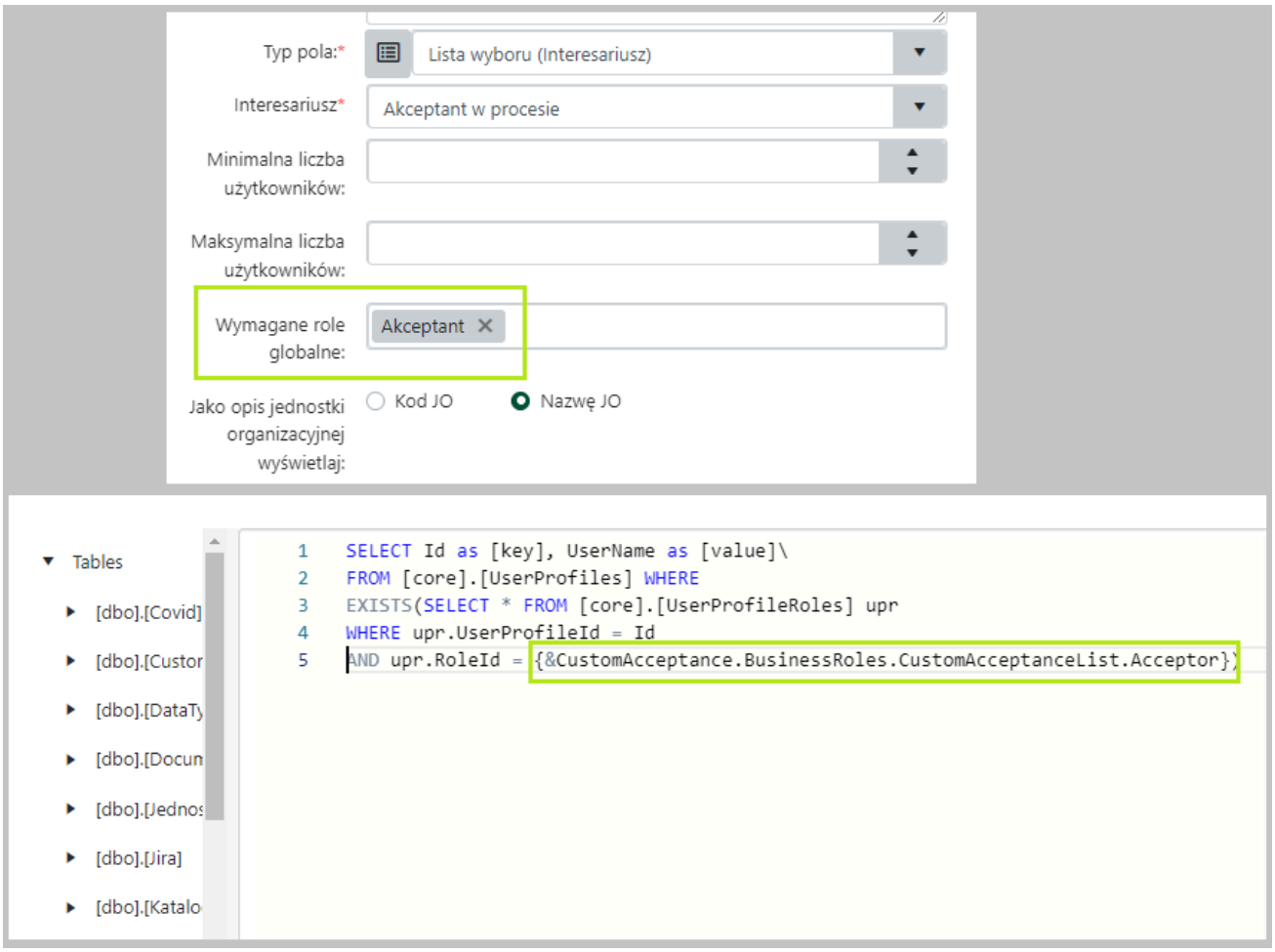
> **WSKAZÓWKA**
> W definicji zmiennych można również odwoływać się do innych zmiennych, używając selektora *AppInLineVariables*.

Jednak tylko w dwóch pierwszych przypadkach implementacja smart numbers interaktywnie wyświetla podpowiedzi. W pozostałych przypadkach można wyrażenie smart numbers wpisać ręcznie (lub wkleić) i będzie ono prawidłowo interpretowane podczas wykonywania aplikacji po stronie FrontSPA. W tych przypadkach zastosowanie mają przede wszystkim zmienne aplikacji używane z selektorem *AppInLineVariables*.

AUTOMATYCZNE UŻYCIE SMART NUMBERS

Podczas konfigurowania listy wyboru interesariusza na formularzu można zawęzić użytkowników wyświetlanych na liście do określonej roli biznesowej. Wybrana rola zostanie

automatycznie dodana w klauzuli *WHERE* zapytania SQL w konfiguracji listy z użyciem składni smart numbers. Pokazano to na ilustracji poniżej.



Użycie smart numbers w konfiguracji listy wyboru interesariuszy

Powiązane tematy:

- Ustawienia systemu, sekcja cache

9.3. PROFILE UŻYTKOWNIKÓW

Aby zalogować się do witryny nAxiom, należy mieć zarejestrowany profil (konto) użytkownika. Profilami może zarządzać użytkownik z predefiniowaną rolą biznesową *Administrator* lub odpowiednimi uprawnieniami PBA. Profile można tworzyć indywidualnie lub wsadowo, importując z arkusza Excel, a także poprzez synchronizację z usługami Active Directory.

W przypadku pierwszej instalacji w witrynie nAxiom są dostępne trzy „fabryczne" profile użytkownika

- *superuser*: wyznacza administratorów uprawnień PBA,
- *admin*: zarządza profilami użytkowników,

- *consultant*: buduje aplikacje biznesowe.

Jednocześnie użytkownicy *admin* i *consultant* są domyślnie administratorami uprawnień PBA i mogą nadawać użytkownikom uprawnienia zgodnie z ich rolą w witrynie.

Indywidualni użytkownicy z rolą systemową *Klient* mogą zmienić w swoim profilu domyślny język aplikacji oraz hasło. Dostęp do własnego profilu również wymaga posiadania odpowiednich uprawnień PBA.

Powiązane tematy:

- Model uprawnień PBA - Wprowadzenie

9.3.1. Listy użytkowników

Podsumowanie:
Widoki `Lista użytkowników` i `Drzewo listy użytkowników` to dostępne w aplikacji *FrontSPA* listy z wykazem zdefiniowanych profili użytkowników dla środowiska bieżącego tenanta. Drzewo listy użytkowników przedstawia listę użytkowników w postaci hierarchicznej z uwzględnieniem zapisanych w profilu użytkownika informacji o przełożonym.

☛ *FrontSPA* > `Użytkownicy` > `Lista użytkowników`
FrontSPA > `Użytkownicy` > `Drzewo listy użytkowników`

1. Zaloguj się do FrontSPA jako użytkownik z rolą `Administrator` lub uprawnieniem do sekcji menu `Użytkownicy` i odpowiednimi uprawnieniami PBA.
2. Kliknij polecenie `Lista użytkowników` w menu głównym.
   Zostanie wyświetlone okno `Lista użytkowników`.

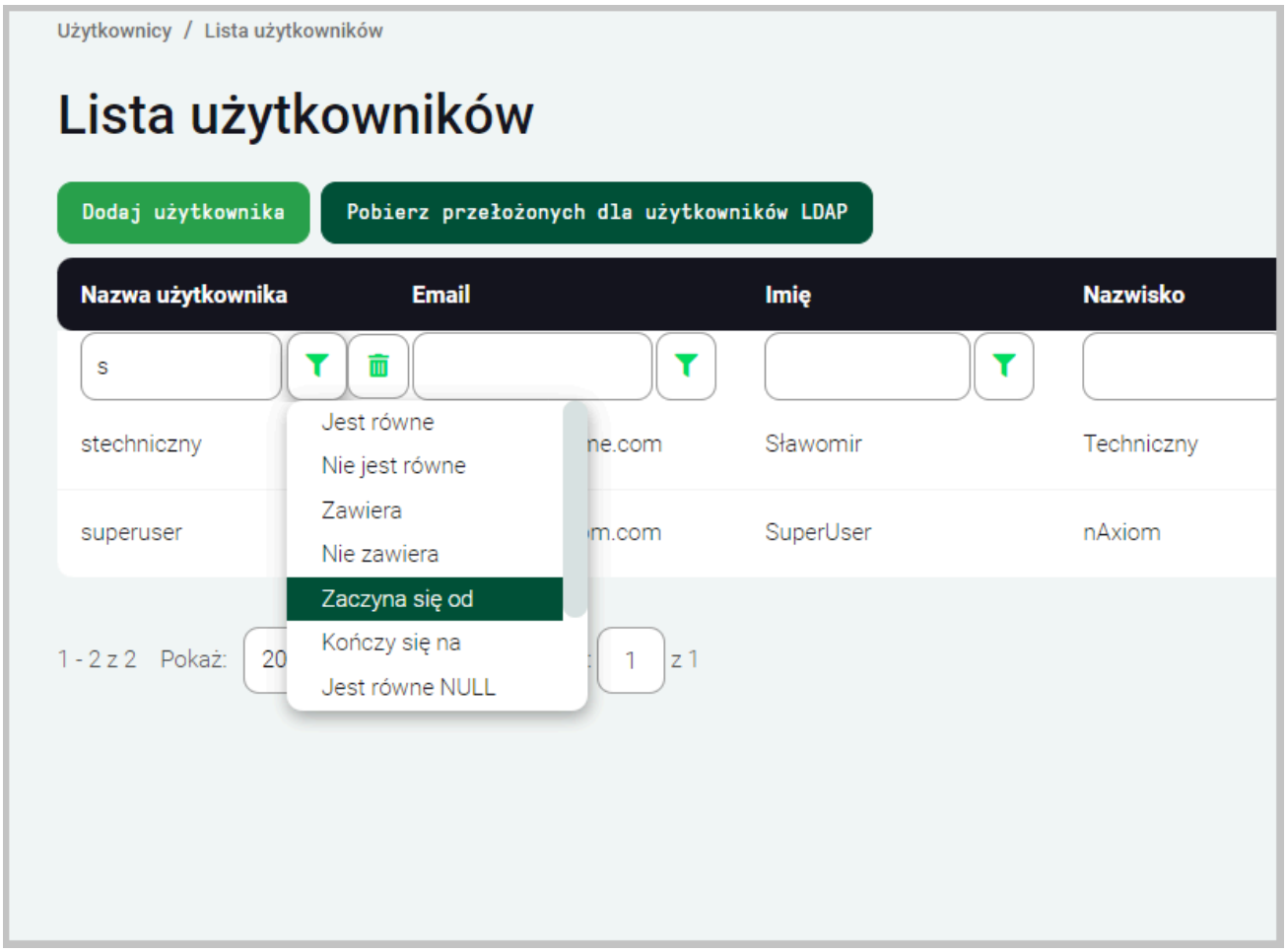
WSKAZÓWKA
Do zarządzania profilami użytkowników potrzebne są odpowiednie uprawnienia PBA.

W tym oknie są dostępne następujące przyciski:

- `Dodaj użytkownika`: wyświetla okno, w którym można wpisać dane nowego konta użytkownika.
- `Pobierz przełożonych dla użytkowników LDAP`: pobiera informacje o przełożonych użytkowników, których dane zostały zaimportowane z zewnętrznego systemu usług katalogowych.

- `Eksportuj listę do pliku MS Excel` (ikona): eksportuje listę użytkowników do pliku w formacie MS Excel, z uwzględnieniem ustawień filtrowania i sortowania; eksportowane są tylko dane z kolumn wyświetlanych w widoku listy.
- `Edytuj`: wyświetla okno, w którym można edytować dane użytkownika.
- `Zablokuj/Odblokuj`: umożliwia zablokowanie/odblokowanie konta użytkownika; użytkownik z zablokowanym kontem nie może się zalogować do witryny nAxiom. (Zablokowanie konta może być skutkiem przekroczenia maksymalnej liczby nieudanych prób zalogowania się). Informacje o blokadzie są zapisywane w tabeli *auth.AspNetUsers* w kolumnie *LockoutEnd*. Ten przycisk jest dostępny pod warunkiem posiadania odpowiedniego uprawnienia PBA (*Blokowanie konta*). Domyślnie to uprawnienia mają przypisani użytkownicy *admin* i *superuser*.

Dla każdej kolumny na liście pod nagłówkiem dostępne jest pole filtra i przycisk filtrowania ( ). Kliknięcie przycisku filtra rozwija listę predefiniowanych wyrażeń filtrowania odpowiednich do typu danych w kolumnie. Na przykład dla kolumny z danymi tekstowymi można wybrać wyrażenie *Zawiera* i wpisać w pole obok ciąg znaków, które mają zawierać odfiltrowane wartości w tej kolumnie.



Filtrowanie wartości na liście użytkowników

Ponadto rekordy listy można sortować, klikając nagłówek kolumny. Pierwsze kliknięcie sortuje w porządku rosnącym, drugie — malejącym, trzecie — wyłącza sortowanie.

Powiązane tematy:

- Obszar Zarządzanie strukturą organizacyjną i użytkownikami

9.3.2. Tworzenie i edycja profilu użytkownika

Podsumowanie:
Profil (konto) użytkownika to zbiór podstawowych danych użytkownika witryny nAxiom, w tym poświadczeń umożliwiających zalogowanie się do witryny nAxiom, oraz informacji o jego rolach w organizacji i powiązaniu ze strukturą organizacyjną.

☛ *FrontSPA* > `Użytkownicy` > `Lista użytkowników` > `Dodaj użytkownika` / `Edytuj`

Aby utworzyć profil użytkownika, wykonaj następujące czynności:

1. Zaloguj się do FrontSPA jako użytkownik z rolą `Administrator` lub odpowiednimi uprawnieniami PBA do edycji profilu użytkowników — patrz informacje powiązane.
2. Kliknij polecenie `Lista użytkowników` w menu głównym.

   WSKAZÓWKA
   Alternatywnie zarządzanie profilami użytkowników możliwe jest również z poziomu listy wyświetlanej przy użyciu polecenia `Drzewo listy użytkowników`.

3. W wyświetlonym oknie `Lista użytkowników` kliknij jasnozielony przycisk `Dodaj użytkownika` z lewej strony u góry listy.
   Zostanie wyświetlone okno `Dodanie profilu użytkownika`.
4. Wypełnij pola zgodnie z opisem w pod ilustracją.
5. Kliknij przycisk `Zapisz`.
   Nowy użytkownik zostanie dodany do listy użytkowników.

W oknie `Dodanie profilu użytkownika` znajdują się następujące pola, listy rozwijane i przełączniki:

- `Nazwa użytkownika`: nazwa używana przez użytkownika do logowania; dozwolone są znaki alfanumeryczne oraz znaki - (minus), . (kropka), _ (podkreślenie), @ (at), +

(plus); patrz artykuł w witrynie Microsoft; nazwa użytkownika może mieć od 1 do 256 znaków.

- `Email`: adres e-mail użytkownika.
- `Imię`: imię użytkownika.
- `Nazwisko`: nazwisko użytkownika.

- `Aktywny`: przełącznik, którym można dezaktywować konto użytkownika; w nAxiom nie ma funkcjonalności usuwania profili użytkowników, dezaktywacja profilu to jedyny sposób zablokowania użytkownikowi posiadającemu profil dostępu do witryny.

- `Aktywny`: przełącznik, którym można dezaktywować konto użytkownika; w nAxiom nie ma funkcjonalności usuwania profili użytkowników, dezaktywacja profilu to jedyny sposób zablokowania użytkownikowi posiadającemu profil dostępu do witryny. Dezaktywacja profilu może wystąpić jako skutek synchronizacji z katalogiem LDAP lub w wyniku żądania do Public API. We wszystkich przypadkach dezaktywacja powoduje natychmiastowe wylogowanie użytkownika z nAxiom.

WAŻNE
System porównuje liczbę licencji z liczbą ***aktywnych*** profili użytkowników. Dezaktywowanie profilu powoduje zwolnienie przypisanej mu licencji.

- `Dostęp do Internal API`: określa, czy użytkownik ma dostęp do wewnętrznego interfejsu API platformy nAxiom.
- `Dostęp do Public API`: określa, czy użytkownik ma dostęp do publicznego interfejsu API platformy nAxiom.

- `Dostęp do Mobile API`: określa, czy użytkownik ma dostęp do interfejsu API platformy nAxiom dedykowanego do obsługi aplikacji mobilnej; jeśli ten przełącznik będzie wyłączony, użytkownik nie będzie mógł zalogować się do aplikacji mobilnej.

- `Dostępna weryfikacja biometryczna`: określa czy użytkownik może korzystać w aplikacji mobilnej z biometrycznych metod uwierzytelniania (odcisk palca, identyfikacja rysów twarzy) zdefiniowanych w urządzeniu mobilnym.

- `Bezterminowe hasło`: (ustawienie dostępne tylko po włączeniu polityki okresu ważności haseł) pozwala wyłączyć dany profil z obowiązku okresowej zmiany hasła zgodnie z polityką bezpieczeństwa.

- `Data ważności`: data ważności profilu użytkownika na platformie nAxiom.
- `Przełożony`: przełożony użytkownika w strukturze organizacyjnej.

- `Rola systemowa`: jedna z trzech ról systemowych (*Klient*, *Klient uproszczony*, *Konsultant*), które można przypisać użytkownikom nAxiom; procedurę dodawania nowego konsultanta opisano w następnym temacie .

- `Rola systemowa`: zwykle *Klient* lub *Klient uproszczony*, to role użytkowników, którzy pracują w aplikacjach biznesowych dostępnych we Front SPA. Użytkownik, który ma mieć dostęp do modułu Admin SPA i móc tworzyć aplikacje może mieć jednocześnie rolę *Konsultant* i jedną z ról klienckich; procedurę dodawania nowego konsultanta opisano w następnym temacie .

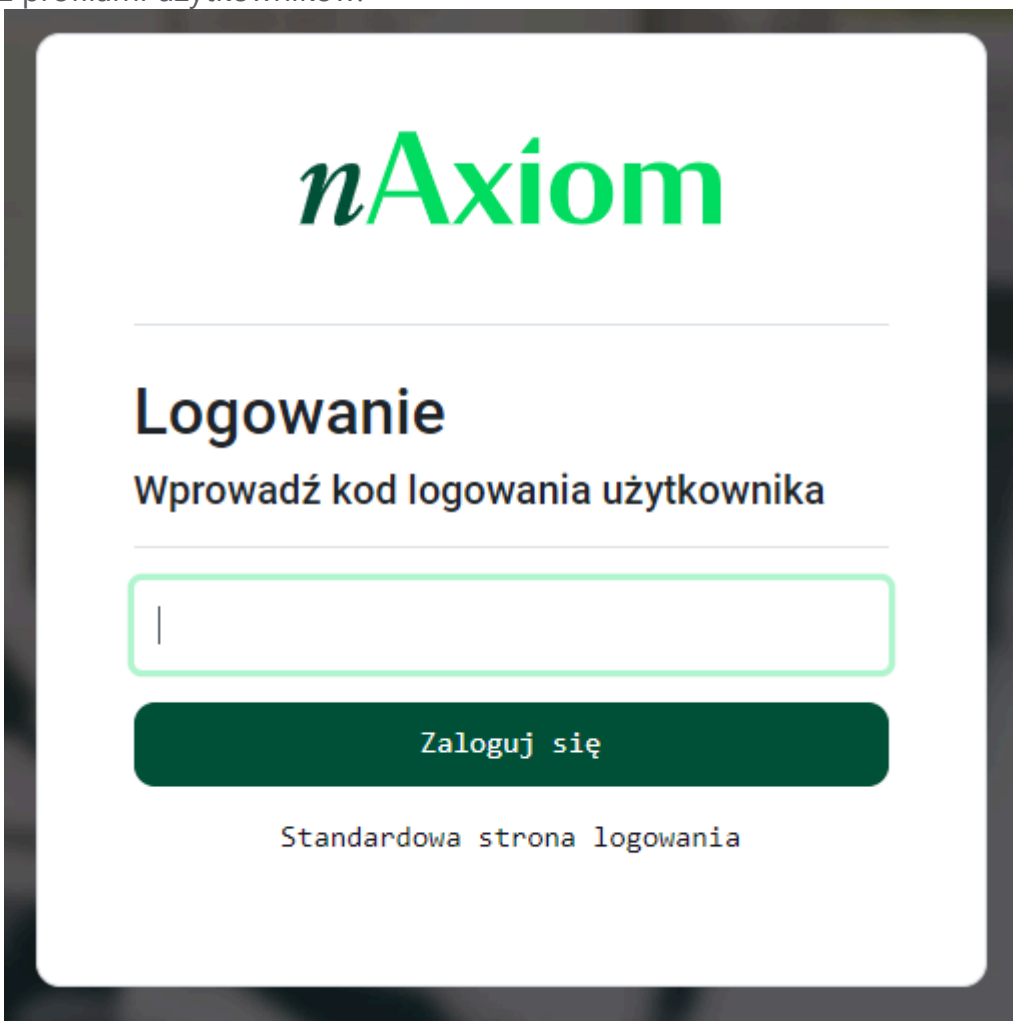
- `Globalne role biznesowe`: jedna lub kilka ról biznesowych zdefiniowanych w nAxiom; informacje o rolach użytkownika są zapisywane w tabeli *core.UserProfileRoles*.
- `Role biznesowe w jednostkach organizacyjnych`: role biznesowe przypisywane użytkownikowi w jednostkach organizacyjnych, do których został dodany; w przypadku nowego użytkownika należy najpierw utworzyć profil, następnie dodać do jednostek organizacyjnych, a potem przypisać mu role w tych jednostkach; przypisanie roli w jednostce wymaga wypełnienia pola `Rola biznesowa` dla odpowiedniego pola `Jednostka organizacyjne`. W tym polu może być wyświetlana ścieżka nazw lub kodów jednostek organizacyjnych w strukturze, zależnie od ustawienia przełącznika w sekcji `Struktura organizacyjna` w ustawieniach systemu w AdminSPA; użytkownikowi można przypisać tylko jedną rolę w jednostce organizacyjnej; informacje o rolach w jednostkach są zapisywane w tabeli *core.OUInstanceUserRoles*.
- `Język`: wersja językowa, w jakiej będzie wyświetlany interfejs nAxiom.

- `Użytkownik LDAP`: przełącznik umożliwiający wskazanie użytkownika, którego dane logowania są importowane z zewnętrznego serwera usług katalogowych.

- `Hasło`, `Potwierdź hasło`: pola służące do ustawienia hasła, przy użyciu którego użytkownik będzie logował się do nAxiom; te pola są wyświetlane tylko w przypadku profili użytkowników definiowanych lokalnie (przełącznik `Użytkownik LDAP` wyłączony).

> **UWAGA**
> Jeśli pola hasła pozostaną puste, podczas zapisu profilu zostanie wykonana akcja wysyłki wiadomości e-mail z linkiem do resetowania hasła; ta akcja musi zostać wskazana w ustawieniach systemu (`ADMINISTRACJA` > `Ustawienia systemu` > `Polityka restowania hasła` > `Wskazanie akcji email do powiadomienia o resecie hasła`). Jeśli taka akcja nie została skonfigurowana lub nie można jej wykonać, zostanie wyświetlony komunikat o błędzie i zapisanie profiu nie będzie możliwe.

- `Uproszczone logowanie`: dowolny ciąg znaków, który będzie pełnił rolę kodu uproszczonego logowania; po włączeniu opcji uproszczonego logowania w ustawieniach konfiguracyjnych (plik *appsettings.json* w podfolderze *AUTH* folderu instalacyjnego, ustawienie `"SimpleLoginPage": true`) zamiast standardowego okna logowania wyświetlane jest okno z jednym polem, w którym należy wpisać ten kod, aby się zalogować; w przypadku przypisania tego samego kodu kilku użytkownikom, będzie się na niego logował użytkownik pierwszy w kolejności w tabeli z profilami użytkowników.



Uproszczone logowanie

- `Drukarka`: nazwa drukarki domyślnej, z której będzie korzystał użytkownik podczas

pracy w nAxiom.

- `Domyślna aplikacja użytkownika`: aplikacja, według której filtrowane będą sekcje menu wyświetlane we FrontSPA dla danego użytkownika w trybie pojedynczej aplikacji. Na liście dostępne są tylko aplikacje, do których użytkownik ma dostęp na podstawie przypisanej roli. Dla użytkowników, dla których żadna aplikacja nie zostanie wybrana, w trybie pojedynczej aplikacji wyświetlane będą sekcje menu powiązane z aplikacją wskazaną w ustawieniach systemu.

- `Po zalogowaniu wyświetlaj pulpit`: dostępne ustawienia:
  - `Pulpit bieżącej aplikacji`: dotyczy trybu pojedynczej aplikacji; wyświetla domyślny pulpit aplikacji domyślnej; patrz opis pulpitów aplikacji .
  - `Mój pulpit domyślny`: pulpit ustawiony przez użytkownika jako domyślny; patrz opis pulpitów użytkownika .

Powyższe dane, oprócz organizacji, ról globalnych i ról w jednostkach, są zapisywane w tabeli *core.UserProfiles*.

Powiązane tematy:

- Obszar Zarządzanie strukturą organizacyjną i użytkownikami
- Model uprawnień PBA - Wprowadzenie
- Struktura organizacyjna
- Tryb pojedynczej aplikacji
- Ustawienia systemu, Tryb pojedynczej aplikacji

9.3.3. Tworzenie nowego konsultanta

Podsumowanie:
Aby utworzyć kolejny profil konsultanta, należy przypisać użytkownikowi rolę systemową *Konsultant* oraz nadać mu uprawnienia PBA do konfiguracji systemu i budowania aplikacji. Dodatkowo, o ile to konieczne, inny konsultant z uprawnieniami PBA do definicji aplikacji musi nadać mu uprawnienia do poszczególnych aplikacji w AdminSPA.

WSKAZÓWKA
W poniższym opisie zakłada się, że czynności administracyjne wykonuje użytkownik z profilem *admin*, ale może to również być inny profil z uprawnieniami administratora.

1. Użytkownik z profilem *admin* tworzy nowy profil użytkownika i przypisuje mu rolę systemową *Konsultant*; patrz Tworzenie i edycja profilu użytkownika .

2. Użytkownik z profilem *superuser* włącza dla profilu *admin* uprawnienia `Dostęp do gałęzi budowania aplikacji oraz do uprawnień aplikacji` w menu `Administracja uprawnieniami`; patrz Wyznaczanie administratora uprawnień PBA .

   > **WSKAZÓWKA**
   > Domyślnie profil *admin* nie ma włączonego dostępu do tych uprawnień.

3. *admin* przypisuje nowemu konsultantowi uprawnienia PBA z gałęzi `Budowanie aplikacji` i `Konfiguracja systemu`; patrz PBA: konfiguracja systemu i PBA: budowanie aplikacji .
4. Konsultant, który ma uprawnienia PBA do definicji istniejących aplikacji przypisuje nowemu konsultantowi odpowiednie uprawnienia do indywidualnych aplikacji; patrz Uprawnienia do definicji aplikacji .

Po wykonaniu powyższych kroków nowy konsultant może przystąpić do pracy.

> **WSKAZÓWKA**
> Zależnie od polityki danej organizacji, po utworzeniu nowego konsultanta, użytkownik z profilem *superuser* może wyłączyć użytkownikowi z profilem *admin* uprawnienia `Dostęp do gałęzi budowania aplikacji oraz do uprawnień aplikacji`.

9.3.4. Edytowanie własnego profilu użytkownika

> **Podsumowanie:**
> Każdy użytkownik nAxiom może w swoim profilu zmienić zmienić kilka ustawień dotyczących witryny nAxiom oraz hasło (pod warunkiem posiadania odpowiednich uprawnień PBA).

> ☛ *FrontSPA* > *imię i nazwisko użytkownika* > `Mój profil` > `Zmień hasło`

Aby zmienić ustawienia w swoim profilu:

1. Zaloguj się do witryny nAxiom; użytkownicy z rolą *Konsultant* muszą przełączyć się do *front office* (zmieniając tekst *admin* na *front* w adresie witryny na pasku adresu w przeglądarce).
2. Kliknij imię i nazwisko wyświetlane w prawym górnym rogu okna przeglądarki.
3. Kliknij polecenie `Mój profil` w rozwiniętym menu.
4. Zostanie wyświetlone okno dialogowe `Edycja profilu użytkownika` z następującymi ustawieniami:

    - `Język`: wersja językowa witryny nAxiom
    - `Procentowy rozmiar panelu notyfikacji`: rozmiar panelu powiadomień systemowych (wyświetlanego po kliknięciu ikony) względem rozmiaru okna przeglądarki; wartość domyślna 0 oznacza rozmiar domyślny, dozwolone wartości od 30 do 80. Zmiana wymaga przelogowania użytkownika.

- `Domyślna aplikacja użytkownika`: aplikacja, według której filtrowane będą sekcje menu wyświetlane we FrontSPA dla danego użytkownika w trybie pojedynczej aplikacji. Na liście dostępne są tylko aplikacje, do których użytkownik ma dostęp na podstawie przypisanej roli. Dla użytkowników, dla których żadna aplikacja nie zostanie wybrana, w trybie pojedynczej aplikacji wyświetlane będą sekcje menu powiązane z aplikacją wskazaną w ustawieniach systemu.

- `Po zalogowaniu wyświetlaj pulpit`: dostępne ustawienia:
    - `Pulpit bieżącej aplikacji`: dotyczy trybu pojedynczej aplikacji; wyświetla domyślny pulpit aplikacji domyślnej; patrz opis pulpitów aplikacji .
    - `Mój pulpit domyślny`: pulpit ustawiony przez użytkownika jako domyślny; patrz opis pulpitów użytkownika .
- `Zmień hasło`: wyświetla okno dialogowe zmiany hasła; wymaga podania bieżącego hasła, nowego hasła oraz potwierdzenia nowego hasła. Hasło musi spełniać wymagania określone w ustawieniach systemowych .
    1. Po dokonaniu zmian w profilu kliknij przycisk `Zapisz`.

Powiązane tematy:
- Obszar Zarządzanie strukturą organizacyjną i użytkownikami
- Tryb pojedynczej aplikacji
- Ustawienia systemu, Tryb pojedynczej aplikacji

9.3.5. Logowanie za pomocą hasła

Podsumowanie:
Konsultant może skonfigurować tryb logowania, w którym użytkownik otrzymuje (osobno) link do strony logowania i hasło. Po kliknięciu linku użytkownik podaje hasło/kod i loguje się do systemu.
Funkcjonalność może być przydatna w scenariuszach, w których użytkownicy uzyskują tymczasowy dostęp do systemu, np. kandydaci w procesie rekrutacji.

- Włącz tryb logowania za pomocą hasła w w ustawieniach tenanta na karcie

`Configuration` w TenantsadminSPA, patrz opis ustawień konfiguracyjnych tenanta .

- Przygotuj teksty w oknie dialogowym (także tłumaczenia) i obraz wyświetlany w tle w ustawieniach tenanta na karcie `Guest login` w TenantsadminSPA, patrz opis tej karty . Wymagana jest konfiguracja dla co najmniej jednego języka.
- Utwórz profilu użytkownika.
- Wyślij użytkownikowi link w formacie:

  ```
  https://<url-witryny-naxiom>/Guest?uuid=<user-guid>&lang=<lang-code>
  ```

  np. https://naxiom.localtest.me/Guest?uuid=454115ec-a4bb-493e-a54a-dc0985c4c8fc&lang=pl.

- Osobnym kanałem wyślij użytkownikowi hasło do jego profilu

Po kliknięciu w link użytkownik zobaczy stronę logowania z oknem, w którym będzie mógł wpisać przekazane hasło. Sam proces uwierzytelnienia użytkownika przebiega tak samo jak dla logowania w trybie zwykłym i jest rejestrowany w audycie.

> **WSKAZÓWKA**
> Kod języka jest opcjonalny, jeśli nie zostanie podany, zostanie użyty język ustawiony w przeglądarce (nagłówek *Accept-Language*). Jeśli dla podanego lub przyjętego z przeglądarki kodu języka nie zostanie znalezione tłumaczenie, okno logowania będzie zawierało teksty w pierwszym zdefiniowanym języku.

9.4. SYNCHRONIZACJA SQL

> **Podsumowanie:**
> Synchronizacja SQL to mechanizm cyklicznego importu danych organizacyjnych (użytkowników i jednostek organizacyjnych) z systemu źródłowego do nAxiom i ich aktualizacji. Umożliwia zsynchronizowanie struktury organizacyjnej w nAxiom z dowolnym systemem, z którego dane można wyeksportować do bazy danych SQL nAxiom. Proces obejmuje dwa etapy: import danych do tabel przejściowych oraz automatyczną aktualizację struktury docelowej.
>
> Informacje o przebiegu synchronizacji są dostępne poprzez monitorowanie statusu synchronizacji tabel *core.SyncUserProfiles* i *core.SyncOUInstances* w tabeli *core.SqlTableStatuses*.

9.5. CZYM JEST SYNCHRONIZACJA SQL

Synchronizacja SQL jest uniwersalnym mechanizmem importu danych organizacyjnych z systemu źródłowego do nAxiom. W odróżnieniu od synchronizacji wbudowanej , która komunikuje się bezpośrednio z źródłem danych (AD, OpenLDAP), synchronizacja SQL polega na:

1. **Imporcie danych** — wyeksportowanie informacji o użytkownikach i jednostkach organizacyjnych z systemu źródłowego do tabel przejściowych w bazie danych nAxiom
2. **Automatycznej synchronizacji** — przetworzenie danych z tabel przejściowych i aktualizacja struktury organizacyjnej w nAxiom

Takie rozdzielenie pozwala na:

- Elastyczne integracje z dowolnymi systemami źródłowymi
- Przygotowanie i walidację danych przed synchronizacją
- Zmianę trybu synchronizacji bez wpływu na proces importu
- Łatwe debugowanie — dane z tabel przejściowych są dostępne dla administracji

9.6. WYMAGANIA I PRZYGOTOWANIE

9.6.1. Dane niezbędne do synchronizacji

Synchronizacja wymaga następujących informacji dla każdego użytkownika:

- Identyfikator użytkownika (login, ID z systemu źródłowego)
- Imię i nazwisko (opcjonalnie rozdzielone na imię i nazwisko)
- Adres e-mail
- Przypisanie do jednostki organizacyjnej
- Status (aktywny/nieaktywny)
- Data wygaśnięcia profilu

Dla jednostek organizacyjnych:

- Identyfikator OU (kod, ID z systemu źródłowego)
- Identyfikator OU nadrzędnej (relacja hierarchiczna)
- Kod
- Nazwa
- Status (aktywna/nieaktywna)
- Data początkowa aktywności

- Data końcowa aktywności

9.6.2. Przygotowanie danych

Dane z systemu źródłowego muszą być wyeksportowane w formacie, który można załadować do tabel synchronizacyjnych nAxiom. Typowym podejściem jest:

- Eksport z systemu źródłowego do pliku CSV lub formatu tekstowego
- Transformacja danych (jeśli wymagana) za pomocą skryptu lub narzędzia ETL
- Załadowanie do tabel przejściowych (`SyncUserProfiles`, `SyncOUInstances`, `SyncOUInstanceUsers`) za pośrednictwem procedury SQL lub skryptu

9.7. KONFIGURACJA

Synchronizacja SQL jest sterowana przez **konfigurację tenanta**. W konfiguracji tenanta określa się między innymi:

- tryb synchronizacji (`SynchronizationType`)
- częstotliwość sprawdzania statusów (`RepeatInterval`)
- domyślne wartości używane podczas synchronizacji, np. język, role i adres e-mail
- dodatkowe opcje wpływające na sposób aktualizacji użytkowników i jednostek organizacyjnych

TaskService odczytuje te ustawienia z konfiguracji tenanta i na ich podstawie wykonuje cykliczną synchronizację.

9.7.1. Tabele synchronizacyjne

Dane przejściowe są przechowywane w czterech tabelach:

core.SyncUserProfiles — profile użytkowników

- `Id` — unikalny identyfikator rekordu (auto-increment)
- `UserId` (nvarchar, UNIQUE) — identyfikator użytkownika z systemu źródłowego (login, GUID itp.)
- `ParentId` — FK do UserId (dla hierarchii, np. manager użytkownika)
- `UserName` — nazwa użytkownika (login wyświetlany)
- `FirstName`, `LastName` — imię i nazwisko
- `Email` — adres e-mail
- `IsActive` (bit) — czy użytkownik jest aktywny (1 = tak, 0 = nie)
- `ExpireDate` (datetime2) — data wygaśnięcia dostępu

core.SyncOUInstances — jednostki organizacyjne

- `Id` — unikalny identyfikator rekordu (auto-increment)
- `OrgUnitId` (nvarchar, UNIQUE) — identyfikator OU z systemu źródłowego (kod OU, GUID itp.)
- `ParentId` — FK do OrgUnitId (OU nadrzędna dla hierarchii)
- `Code` — kod organizacyjny OU (np. “IT”, “HR”, “SALES”)
- `Name` — nazwa jednostki organizacyjnej
- `IsActive` (bit) — czy OU jest aktywna
- `ActiveFrom` (datetime2) — data aktywacji
- `ActiveUntil` (datetime2) — data deaktywacji (NULL jeśli bez ograniczenia)

core.SyncOUInstanceUsers — przypisania użytkowników do OU

- `OrgUnitId` (nvarchar, FK) — referencja do `SyncOUInstances.OrgUnitId`
- `UserId` (nvarchar, FK) — referencja do `SyncUserProfiles.UserId`
- Tabela asocjacyjna bez własnego ID — określa powiązania użytkowników z jednostkami

core.SqlTableStatuses — statusy synchronizacji

- `Id` — unikalny identyfikator
- `Name` (nvarchar) — nazwa tabeli (`SyncUserProfiles` lub `SyncOUInstances`)
- `Status` (int) — kod statusu (0=ReadyForData, 1=DataInput, 2=ReadyForSync, 3=InSync, 4=Error.)

WAŻNE

Statusy synchronizacji są przechowywane w *core.SqlTableStatuses* na poziomie tabeli, nie na poziomie poszczególnych rekordów. Stan wskazuje na postęp całej partii danych dla danej tabeli. **Tabela** `SyncOUInstanceUsers` **nie ma własnego statusu w** `SqlTableStatuses` — jej synchronizacja jest zintegrowana z przetwarzaniem `SyncOUInstances`. Przypisania użytkowników do OU (zawarte w `SyncOUInstanceUsers`) są automatycznie przetwarzane w ramach synchronizacji jednostek organizacyjnych, bez osobnej kontroli statusu. Oznacza to, że:

- Gdy załadujesz dane do `SyncOUInstanceUsers`, nie zmienią się statusy w `SqlTableStatuses`
- Synchronizacja przypisań następuje automatycznie gdy `SyncOUInstances` ma status `InSync`
- Ewentualne błędy w przypisaniach będą zaraportowane w logach TaskService podczas przetwarzania `SyncOUInstances`

9.7.2. Przygotowanie do pierwszej synchronizacji

Przed załadowaniem danych do tabel synchronizacyjnych:

1. **Przygotuj dane** w systemie źródłowym i wyeksportuj je w wybranym formacie
2. **Zweryfikuj dane** przed załadowaniem:
   - Czy identyfikatory użytkowników (`UserId`) są unikalne
   - Czy identyfikatory OU (`OrgUnitId`) są unikalne
   - Czy hierarchia OU jest spójna (wszystkie `ParentId` wskazują na istniejące `OrgUnitId`)
   - Czy przypisania użytkowników do OU są prawidłowe (`SyncOUInstanceUsers` zawiera tylko istniejące `UserId` i `OrgUnitId`)
3. **Załaduj dane** do tabel synchronizacyjnych
   - `SyncUserProfiles` — profili użytkowników
   - `SyncOUInstances` — jednostek organizacyjnych
   - `SyncOUInstanceUsers` — przypisań użytkowników do OU
4. **Upewnij się, że status = ReadyForSync** — jeśli załadujesz przez API, status ustawia się automatycznie; jeśli bezpośrednio SQL, musisz ustawić ręcznie
5. **Synchronizacja uruchomi się automatycznie** — TaskService ma wbudowany timer, który zgodnie z konfiguracją tenanta co `RepeatInterval` sekund wyzwala synchronizację (patrz sekcja *Jak działa synchronizacja*)

9.8. PRZEBIEG SYNCHRONIZACJI

9.8.1. Fazy synchronizacji

1. Import danych

Administrator lub skrypt ETL przygotowuje dane w tabelach przejściowych. Dane mogą być załadowane poprzez:

- Procedury SQL (`INSERT INTO SyncUserProfiles ...`)
- Skrypty ETL
- Public API nAxiom (które obsługuje walidację i zmianę statusu)

Tabele są zasilane danymi z systemu źródłowego:

- `SyncUserProfiles` — lista użytkowników
- `SyncOUInstances` — struktura organizacyjna
- `SyncOUInstanceUsers` — przypisania użytkowników do OU

9.8.2. Endpointy PublicAPI do zasilania tabel przejściowych

Jeśli dane są ładowane przez PublicAPI (zamiast bezpośrednio SQL), można użyć endpointów kontrolera `StructureSynchronization`.

Ogólny wzorzec ścieżki w PublicAPI to:

- `/{Controller}/{Action}`

W praktyce dla synchronizacji struktury używane są:

- `POST /StructureSynchronization/UserProfileList`
  - Zasila tabelę `core.SyncUserProfiles`
  - Body: obiekt z listą `UserProfiles` (dla każdego elementu m.in. `UserId`, `ParentId`, `UserName`, `Email`, `FirstName`, `LastName`, `IsActive`, `ExpireDate`)
- `POST /StructureSynchronization/OUInstanceList`
  - Zasila tabelę `core.SyncOUInstances`
  - Body: obiekt z listą `OUInstances` (dla każdego elementu m.in. `OrgUnitId`, `ParentId`, `Code`, `Name`, `IsActive`, `ActiveFrom`, `ActiveUntil`)
- `POST /StructureSynchronization/OUInstanceUsersList`
  - Zasila tabelę `core.SyncOUInstanceUsers`

- Body: obiekt z listą `OUInstanceUsers` (dla każdego elementu: `OrgUnitId`, `UserId`)

> **WAŻNE**
> Endpointy PublicAPI do importu danych wymagają poprawnej autoryzacji PublicAPI i odpowiednich uprawnień użytkownika technicznego. Przy imporcie przez API status tabel (`core.SqlTableStatuses`) jest obsługiwany automatycznie przez aplikację (w tym przejście do `ReadyForSync`).

Uwaga: endpoint `POST /StructureSynchronization/SynchronizeStructure` nie służy do zasilania tabel przejściowych. Służy do uruchomienia synchronizacji danych już załadowanych do tabel `Sync*`.

9.8.3. Przykładowe payloady JSON (PublicAPI)

Przykład dla `POST /StructureSynchronization/UserProfileList`:

```
{
  "userProfiles": [
    {
      "userProfile": {
        "id": 0,
        "userId": "u-001",
        "parentId": null,
        "userName": "jan.kowalski",
        "email": "jan.kowalski@acme.pl",
        "firstName": "Jan",
        "lastName": "Kowalski",
        "isActive": true,
        "expireDate": "2026-12-31T00:00:00"
      }
    }
  ]
}
```

Przykład dla `POST /StructureSynchronization/OUInstanceList`:

```
{
  "ouInstances": [
    {
      "ouInstance": {
```

```
        "id": 0,
        "orgUnitId": "OU-IT",
        "parentId": null,
        "code": "IT",
        "name": "Dzial IT",
        "isActive": true,
        "activeFrom": "2026-01-01T00:00:00",
        "activeUntil": null
      }
    }
  ]
}
```

Przykład dla `POST /StructureSynchronization/OUInstanceUsersList`:

```
{
  "ouInstanceUsers": [
    {
      "ouInstanceUser": {
        "orgUnitId": "OU-IT",
        "userId": "u-001"
      }
    }
  ]
}
```

> **WAŻNE**
> Nazwy właściwości JSON podawaj zgodnie z kontraktem API (jak w przykładach powyżej). W praktyce integracyjnej najbezpieczniej najpierw wykonać test na pojedynczym rekordzie, a dopiero potem wysyłać pełne paczki danych.

2. Automatyczne wyzwolenie synchronizacji (nie ma ręcznego “uruchamiania”)

TaskService ma wbudowany timer (`StructureSynchronizationConnector`), który:

- Co `RepeatInterval` sekund (z konfiguracji) sprawdza status tabel w `SqlTableStatuses`
- Szuka tabel ze statusem `ReadyForSync` (kod 2)
- Jeśli znaleziona, zmienia status na `InSync` (kod 3) i uruchamia synchronizację

Wymagane kroki:

1. Załaduj dane do tabel przejściowych
2. Upewnij się, że status w `SqlTableStatuses` dla obu tabel (`SyncUserProfiles`, `SyncOUInstances`) wynosi `ReadyForSync` (= 2)
3. Czekaj na timer TaskService — synchronizacja uruchomi się automatycznie w następnym cyklu

Jeśli dane są załadowane przez Public API, zmiana statusu na `ReadyForSync` następuje automatycznie. Jeśli załadujesz dane bezpośrednio SQL, musisz ręcznie ustawić status:

```
UPDATE core.SqlTableStatuses SET Status = 2 WHERE Name = 'SyncUserProfiles'
UPDATE core.SqlTableStatuses SET Status = 2 WHERE Name = 'SyncOUInstances'
```

3. Przetwarzanie i aktualizacja struktury

TaskService:

- Odczytuje dane z tabel przejściowych (`SyncUserProfiles`, `SyncOUInstances`, `SyncOUInstanceUsers`)
- Przetwarza je zgodnie z wybranym trybem (Full, Update, Differential)
- Aktualizuje strukturę w tabelach docelowych (`UserProfiles`, `OUInstances`, `OUInstanceUsers`)
- **Przypisania użytkowników** z `SyncOUInstanceUsers` są przetwarzane automatycznie w ramach synchronizacji `SyncOUInstances`

4. Monitorowanie i weryfikacja

Administrator monitoruje statusy synchronizacji:

- Statusy w `SqlTableStatuses` wskazują postęp (`ReadyForData` → `DataInput` [przy imporcie przez API] → `ReadyForSync` → `InSync` → `ReadyForData` lub `Error`)
- Błędy są zaraportowane w logach TaskService
- W razie problemów dane w tabelach przejściowych pozostają dostępne do analizy

9.8.4. Schemat czasowy

Synchronizacja jest **w pełni automatyczna**:

- TaskService ma timer, który co `RepeatInterval` sekund (typowo 60-300 sekund, zależy od konfiguracji tenanta) sprawdza status
- Synchronizacja jest wyzwalana automatycznie gdy status = `ReadyForSync`

- Administrator nie ma GUI ani polecenia do ręcznego uruchamiania — synchronizacja działa w tle
- Jedynym sposobem kontroli jest:
    - Zmiana `RepeatInterval` w konfiguracji tenanta (aby zmieniła się częstotliwość polingu)
    - Ustawienie statusu na `ReadyForSync` (aby wyzwolić synchronizację w następnym cyklu timera)
    - Monitorowanie logów — aby zobaczyć czy synchronizacja się powiodła

9.9. TRYBY SYNCHRONIZACJI

W konfiguracji tenanta ustawienie 'SynchrozationType' określa tryb synchronizacji czyli, jak dane z tabel przejściowych są stosowane do struktury docelowej. Do dyspozycji są trzy główne tryby:

Tryb	Dodawanie nowych rekordów	Aktualizacja istniejących	Deaktywacja brakujących	Typowe zastosowanie
Full	Tak	Tak	Tak	Pełne odświeżenie struktury na podstawie kompletnego źródła danych
Update	Nie	Tak	Tak	Aktualizacja rekordów już istniejących w nAxiom, bez zakładania nowych
Differential	Tak	Tak	Tak	Synchronizacja oparta o dane różnicowe; w praktyce bardzo zbliżona do trybu Full

WAŻNE
Tryb `Update` nie tworzy nowych użytkowników ani nowych jednostek organizacyjnych. Jeżeli podczas synchronizacji system nie znajdzie odpowiednika danego rekordu w strukturze docelowej, rekord zostanie pominięty. Jednocześnie tryb ten nadal może deaktywować obiekty brakujące w danych źródłowych.

9.9.1. Tryb Full (Pełna synchronizacja)

Tryb ten implementuje najpełniejszą synchronizację danych:

- **Dodaje** nowych użytkowników i OU
- **Aktualizuje** istniejące rekordy (dane, status)

- **Deaktywuje** użytkowników i OU brakujące w danych źródłowych (użytkownik istnieje w nAxiom, ale nie ma go w tabelach przejściowych)

Zastosowanie: Idealne do pełnego odświeżenia struktury, gdzie dane źródłowe zawsze odzwierciedlają aktualną rzeczywistość. Po synchronizacji struktura w nAxiom będzie dokładnym odzwierciedleniem danych źródłowych.

Uwaga: Deaktywowanie brakujących użytkowników może wpłynąć na dostęp i uprawnienia. Należy upewnić się, że dane źródłowe zawierają wszystkich użytkowników, którzy powinni być aktywni.

9.9.2. Tryb Update (Aktualizacja)

Tryb ten jest bardziej konserwatywny:

- **Aktualizuje** istniejące użytkowników i OU (dane i status)
- **Nie dodaje** nowych rekordów
- **Deaktywuje** użytkowników i OU brakujące (tak jak w trybie Full)

Zastosowanie: Gdy struktura bazowa istnieje już w nAxiom, a chcemy aktualizować tylko dane istniejących rekordów. Zapobiega przypadkowemu dodaniu duplikatów, ale nadal stosuje deaktywowanie brakujących rekordów.

Uwaga: Semantycznie tryb "Update" zawiera również akcję deaktywowania, co może być nieintuicyjne. Umownie, deaktywowanie brakujących jest powszechnym zachowaniem synchronizacji.

9.9.3. Tryb Differential (Synchronizacja różnicowa)

Praktycznie zachowuje się podobnie do trybu Full, ale logika wewnętrzna jest zoptymalizowana do wskaźników różnic między danymi:

- Dodaje nowe rekordy
- Aktualizuje istniejące
- Deaktywuje brakujące

Zastosowanie: Gdy chcemy zachować logikę Full, ale dane źródłowe są dostępne jako zbiory delta (wpisy nowe, zmienione, usunięte) zamiast całej listy.

9.9.4. Wybór trybu

Decyzja o wyborze trybu zależy od:

- **Dostępności kompletnych danych** — jeśli dane źródłowe zawsze zawierają pełną listę, stosuj tryb Full
- **Bezpieczeństwa** — jeśli chcesz uniknąć przypadkowej deaktywacji, rozważ

ograniczenia na poziomie bazy danych
- **Nowych vs. istniejących danych** — tryb Update jest bezpieczniejszy dla struktur, które już zawierają dane lokalne

9.10. JAK DZIAŁA SYNCHRONIZACJA — TECHNICZNY PRZEBIEG

9.10.1. Timer TaskService (StructureSynchronizationConnector)

TaskService zawiera wbudowany usługę `SynchronizeStructureTimedHostedService`, która:

Inicjowanie:

- Startuje automatycznie gdy tenant jest aktywowany
- Konfiguracja pobierana z `StructureSynchronizationConfiguration.RepeatInterval` (czas w sekundach)

Cykliczny monitoring:

- Co `RepeatInterval` sekund timer sprawdza tabelę `core.SqlTableStatuses`
- Szuka wpisów dla `SyncUserProfiles` i `SyncOUInstances`
- Sprawdza czy `Status` = `2` (ReadyForSync)

Wyzwolenie synchronizacji:

```
Jeśli Status == ReadyForSync (2):
  1. Zmiana Status → InSync (3)
  2. Uruchomienie SynchronizeStructureFromSyncTableHandler
  3. Odczyt danych z tabel przejściowych
  4. Przetworzenie zgodnie z trybem
  5. Aktualizacja tabel docelowych
  6. Zmiana Status → ReadyForData (0) [sukces] lub Error (4) [błąd]
```

Parametry:

- `RepeatInterval`: domyślnie 60-300 sekund (zależy od instalacji)
- Parametry pochodzą z konfiguracji tenanta, w szczególności z sekcji `StructureSynchronizationConfiguration`; wartość 0 wyłącza synchronizację
- **UWAGA:** `RepeatInterval` nie jest ustawieniem tabel synchronizacyjnych ani bazy danych. Jest elementem konfiguracji tenanta odczytywanej przez TaskService.

9.10.2. Ustawienie statusu ReadyForSync

Status `ReadyForSync` (kod 2) powinien być ustawiony na dwa sposoby:

1. API (zalecane):

- Załadowanie danych przez API automatycznie ustawia `Status = ReadyForSync` po wkładzie danych
- Przykład: `SaveSyncTableUserProfileListHandler` po INSERT danych zmienia status na 2

2. Ręcznie SQL (dla zaawansowanych):

- Jeśli załadujesz dane bezpośrednio INSERT, musisz ręcznie ustawić status:

```
UPDATE core.SqlTableStatuses SET Status = 2 WHERE Name = 'SyncUserProfiles'
UPDATE core.SqlTableStatuses SET Status = 2 WHERE Name = 'SyncOUInstances'
```

- Po tym kroku timer TaskService w następnym cyklu wyzwoli synchronizację

9.11. STATUSY I MONITOROWANIE

9.11.1. Cykl statusów synchronizacji

Statusy synchronizacji są przechowywane w tabeli `core.SqlTableStatuses` na poziomie całej tabeli (nie na poziomie poszczególnych rekordów). Dla tabel przejściowych `SyncUserProfiles` i `SyncOUInstances` istnieje rekord w `SqlTableStatuses` z jego aktualnym stanem:

Status	Znaczenie	Opis
0 — ReadyForData	Gotów do przyjęcia danych	Tabela oczekuje na załadowanie danych z systemu źródłowego
1 — DataInput	Dane są ładowane	Trwa wczytywanie danych do tabeli
2 — ReadyForSync	Gotów do synchronizacji	Dane są kompletne i gotowe do przetworzenia
3 — InSync	Synchronizacja w trakcie	Trwa przetwarzanie danych i aktualizacja tabel systemowych
4 — Error	Błąd	Podczas synchronizacji wystąpił błąd; szczegóły są dostępne w logach TaskService

Uwaga: Tabela `SyncOUInstanceUsers` nie ma własnego statusu w `SqlTableStatuses`. Jej synchronizacja jest automatyczną częścią przetwarzania `SyncOUInstances`. To oznacza:

- Przypisania użytkowników do OU są przetwarzane razem z OU na podstawie statusu `SyncOUInstances`
- Gdy `SyncOUInstances` ma status `InSync` (3), również dane z `SyncOUInstanceUsers` są synchronizowane w tym samym cyklu
- Brakuje osobnej kontroli statusu dla przypisań — są one zawsze przetwarzane razem z jednostkami

9.11.2. Monitorowanie synchronizacji

Aby monitorować przebieg synchronizacji:

1. **Sprawdź statusy tabeli** — `SELECT * FROM core.SqlTableStatuses WHERE Name IN ('SyncUserProfiles', 'SyncOUInstances')`
    - **Uwaga:** Brak statusu dla `SyncOUInstanceUsers` — jest ona przetwarzana razem z `SyncOUInstances`
2. **Obserwuj zawartość tabel** — czy rekordy w `SyncUserProfiles`, `SyncOUInstances` i `SyncOUInstanceUsers` zawierają oczekiwane dane
3. **Przeanalizuj błędy** — jeśli status to `4 (Error)`, przejrzyj logi TaskService w celu znalezienia szczegółów
4. **Porównaj dane** — porównaj zawartość tabel przejściowych z tabelami docelowymi (`UserProfiles`, `OUInstances`)

9.11.3. Logi synchronizacji

TaskService loguje każde uruchomienie synchronizacji, w tym:

- Datę i godzinę
- Tryb synchronizacji
- Liczbę przetworzonych rekordów
- Błędy i ostrzeżenia
- Czas trwania

Logi pomagają w diagnostyce problemów i analizie historii synchronizacji.

WAŻNE

Status nie zawsze odzwierciedla rzeczywisty stan synchronizacji

TaskService uruchamia cykl synchronizacji co `RepeatInterval` i po zakończeniu cyklu może ustawić status na `ReadyForData` (0), nawet jeśli w poprzednim cyklu wystąpił `Error` (4). W praktyce można zaobserwować sekwencję: `2 (ReadyForSync)` -> `3 (InSync)` -> `4 (Error)` -> po kolejnym odświeżeniu/cyklu `0 (ReadyForData)`.

Oznacza to, że **status=ReadyForData nie gwarantuje, że poprzednia próba synchronizacji przebiegła bez błędów**. Status informuje o bieżącej gotowości tabeli do przyjęcia kolejnych danych, a nie o jakości poprzedniego przebiegu.

Administrator MUSI regularnie sprawdzać logi TaskService, aby upewnić się, że synchronizacja faktycznie się powiodła. Błędy (np. duplikaty, naruszenia FK, konflikty danych) mogą być zaraportowane tylko w logach, a status będzie wyglądać na "OK".

Praktyka: Po każdej synchronizacji:

1. Sprawdź status w `SqlTableStatuses` — czy zmienił się na `0 (ReadyForData)`
2. Przejrzyj logi TaskService z ostatnich minut — szukaj błędów, ostrzeżeń, liczby przetworzonych rekordów
3. Opcjonalnie: Porównaj liczbę rekordów w tabelach przejściowych z liczbą zsynchronizowanych rekordów (może się nie zgadzać jeśli były błędy)

9.12. ROZWIĄZYWANIE PROBLEMÓW

9.12.1. Synchronizacja nie zaczyna się

Przyczyny:

- Timer TaskService nie jest uruchomiony (serwis nie działa itp.)
- Tabele przejściowe są puste (brak danych w *SyncUserProfiles* lub *SyncOUInstances*)
- Status w `SqlTableStatuses` nie jest `ReadyForSync` (2) — synchronizacja czeka tylko na ten status
- `RepeatInterval` jest bardzo duży — timer sprawdza rzadko, czekaj dłużej

Rozwiązanie:

- Sprawdź, czy TaskService jest uruchomiony (pula aplikacji lub kontener)
- Weryfikuj, czy dane zostały załadowane do tabel przejściowych:

```
SELECT COUNT(*) FROM core.SyncUserProfiles;
SELECT COUNT(*) FROM core.SyncOUInstances;
```

- Sprawdź statusy i `RepeatInterval`:

```
SELECT * FROM core.SqlTableStatuses WHERE Name IN ('SyncUserProfiles', 'SyncOUInstances');
```

  - Sprawdź `RepeatInterval` w konfiguracji tenanta, w sekcji `StructureSynchronizationConfiguration`. Jeśli `RepeatInterval` jest równe 0 lub brak tego ustawienia, timer się nie uruchamia lub sprawdza status zbyt rzadko.
- Jeśli status to `0 (ReadyForData)` a chcesz wyzwolić synchronizację, ustaw go na `2`:

```
UPDATE core.SqlTableStatuses SET Status = 2 WHERE Name IN ('SyncUserProfiles', 'SyncOUInstances')
```

- Czekaj `RepeatInterval` sekund na uruchomienie timera (np. jeśli RepeatInterval=60, czekaj minutę)

9.12.2. Użytkownicy lub OU nie są synchronizowani

Przyczyny:

- Brakujące dane w tabelach przejściowych
- Hierarchia OU jest uszkodzona (OU nadrzędna nie istnieje)
- Przypisania użytkowników (`SyncOUInstanceUsers`) są niekompletne
- Identyfikatory użytkowników nie zgadzają się między tabelami

Rozwiązanie:

- Sprawdź dane w `SyncUserProfiles` i `SyncOUInstances` — czy wszystkie wymagane pola są wypełnione
- Zweryfikuj hierarchię OU — wszystkie OU nadrzędne muszą być zdefiniowane
- Sprawdź przypisania w `SyncOUInstanceUsers` — czy użytkownicy są przypisani do OU
- Porównaj wartości identyfikatorów między tabelami

9.12.3. Błędy synchronizacji — status Error

Przyczyny:

- Naruszenie integralności danych (duplikaty UserId lub OrgUnitId, wartości NULL w wymaganych polach)

- Uszkodzona hierarchia (ParentId wskazuje na nieistniejący rekord)
- Konflikt z istniejącymi danymi w tabelach docelowych
- Problem z uprawnieniami bazy danych dla TaskService

Rozwiązanie:

- Odczytaj komunikat błędu z logów TaskService
- Wyczyść dane tabel przejściowych i spróbuj ponownie załadować dane (bezpiecznie dla relacji FK): `DELETE FROM core.SyncOUInstanceUsers; DELETE FROM core.SyncUserProfiles; DELETE FROM core.SyncOUInstances;`
- Zweryfikuj integrację danych przed załadowaniem:
  - Czy `UserId` są unikalne w `SyncUserProfiles`?
  - Czy `OrgUnitId` są unikalne w `SyncOUInstances`?
  - Czy wszystkie `ParentId` w `SyncOUInstances` wskazują na istniejące `OrgUnitId`?
  - Czy wszyscy użytkownicy w `SyncOUInstanceUsers` istnieją w `SyncUserProfiles`?
- Jeśli problem utrzymuje się, zweryfikuj konta bazy danych — czy TaskService ma uprawnienia do odczytu i zapisu

9.12.4. Deaktywowani użytkownicy utracili dostęp

Przyczyna:

- Uruchomiono synchronizację w trybie Full lub Update, a użytkowników nie ma w danych źródłowych

Rozwiązanie:

- Sprawdź, czy użytkownicy faktycznie powinni być deaktywowani
- Jeśli deaktywacja była błędem, przywróć dane z kopii zapasowej
- Zmień tryb synchronizacji na bardziej konserwatywny (np. Update zamiast Full)
- Jeśli to możliwe, skonfiguruj filtrowanie na poziomie TaskService, aby nie deaktywować wybranych użytkowników

9.12.5. Przypisania użytkowników do OU nie są synchronizowane

Przyczyny:

- Dane w `SyncOUInstanceUsers` są załadowane, ale status `SyncOUInstances` jest `0 (ReadyForData)` — synchronizacja nie została uruchomiona
- Uszkodzone referencje FK — `UserId` lub `OrgUnitId` w `SyncOUInstanceUsers`

wskazują na nieistniejące rekordy
- Luka w danych — niektórych użytkowników lub OU nie ma w `SyncUserProfiles` lub `SyncOUInstances`

Rozwiązanie:

- Upewnij się, że status `SyncOUInstances` to `2 (ReadyForSync)` — przypisania są przetwarzane razem z OU
- Sprawdź integrość danych (patrz skrypt walidacji FK w sekcji *Przydatne skrypty SQL*)
- Jeśli wszystkie dane są prawidłowe, ale przypisania nie zostały zsynchronizowane, poczekaj aż synchronizacja przejdzie do statusu `3 (InSync)`
- Sprawdź logi TaskService

Przyczyna (w wersji wcześniejszej):

- W starszych wersjach nAxiom przed wersją 1.15.6.4 synchronizacja agresywnie czyszczała przypisania użytkowników do OU

Rozwiązanie:

- Upewnij się, że korzystasz z wersji 1.15.6.4 lub nowszej, która zawiera poprawkę
- Przy aktualizacji z wersji starszej przywróć przypisania z kopii zapasowej lub przesynchronizuj dane

9.12.6. Synchronizacja wygląda na udaną, ale dane nie są aktualizowane

Przyczyna:

- Status zmienił się na `ReadyForData` (0), ale w logach TaskService są błędy
- Część danych została zsynchronizowana, część odrzucona z powodu błędów
- Status nie wskazuje na błędy — zmienia się na `ReadyForData` niezależnie od powodzenia

Rozwiązanie:

- **Zawsze sprawdzaj logi TaskService** — status `ReadyForData` nie gwarantuje powodzenia
- Przejrzyj logi z czasu synchronizacji — szukaj:
    - Błędów: “Error”, “Exception”, “Failed”
    - Ostrzeżeń: “Warning”, “skipped”, “ignored”
    - Liczby przetworzonych vs. załadowanych rekordów — czy się zgadzają?
- Porównaj dane w tabelach docelowych (`UserProfiles`, `OUInstances`) z liczbą w tabelach przejściowych — jeśli liczby się nie zgadzają, część danych została

odrzucona
- Sprawdź szczegóły błędów w logach — mogą wskazywać na:
    - Duplikaty klucza w danych źródłowych
    - Naruszenia integralności danych (NULL w wymaganych polach)
    - Konflikty z istniejącymi danymi

Praktyka bezpieczna:

1. Po każdej synchronizacji odczekaj kilka minut
2. Sprawdzaj status — czy zmienił się na `0 (ReadyForData)`
3. **Przejrzyj logi TaskService z ostatniego okresu** (co najmniej 10 minut wstecz)
4. Sprawdź liczbę rekordów:

```sql
-- Liczba w tabelach przejściowych
SELECT COUNT(*) FROM core.SyncUserProfiles;
SELECT COUNT(*) FROM core.SyncOUInstances;

-- Liczba w tabelach docelowych
SELECT COUNT(*) FROM core.UserProfiles;
SELECT COUNT(*) FROM core.OUInstances;
```

5. Jeśli liczby się nie zgadzają lub są błędy w logach — przeprowadź diagnostykę

9.12.7. Timer TaskService nie wyzwala synchronizacji mimo ustawienia ReadyForSync

Przyczyny:

- Timer nie jest uruchomiony (serwis nie startuje, zawiesił się)
- Konfiguracja `RepeatInterval` jest ustawiona na 0 lub bardzo dużą wartość
- Serwis TaskService ma błędy w logach, które uniemożliwiają działanie timera

Rozwiązanie:

- Sprawdź, czy TaskService jest aktywny — czy serwis uruchamia się, czy ma statusu "running"
- Przeczytaj logi TaskService z ostatnich minut — szukaj błędów wokół "StructureSynchronizationConnector", "SynchronizeStructure"
- Sprawdź konfigurację `RepeatInterval` w konfiguracji tenanta, w sekcji `StructureSynchronizationConfiguration`. Jeśli `RepeatInterval` wynosi 0 lub brak tego ustawienia, timer się nie uruchamia. Zmień wartość na rozsądną (60-300 sekund) i zrestartuj TaskService.

- Jeśli timer startuje ale nie reaguje, spróbuj zrestartować serwis TaskService
- Jeśli problem utrzymuje się, sprawdź czy nie ma deadlocka w bazie — czy inne procesy blokują `SqlTableStatuses`

9.13. PRZYDATNE SKRYPTY SQL

9.13.1. Sprawdzenie statusów synchronizacji

```
-- Pokaż aktualny status każdej tabeli synchronizacyjnej
SELECT Name, Status,
  CASE Status
    WHEN 0 THEN 'ReadyForData'
    WHEN 1 THEN 'DataInput'
    WHEN 2 THEN 'ReadyForSync'
    WHEN 3 THEN 'InSync'
    WHEN 4 THEN 'Error'
    ELSE 'Unknown'
  END AS StatusName
FROM core.SqlTableStatuses
WHERE Name IN ('SyncUserProfiles', 'SyncOUInstances')
ORDER BY Name;
```

9.13.2. Liczba załadowanych danych

```
-- Liczba rekordów w tabelach synchronizacyjnych
SELECT
  'SyncUserProfiles' AS TableName, COUNT(*) AS RecordCount FROM core.SyncUserProfiles
UNION ALL
SELECT 'SyncOUInstances', COUNT(*) FROM core.SyncOUInstances
UNION ALL
SELECT 'SyncOUInstanceUsers', COUNT(*) FROM core.SyncOUInstanceUsers;
```

9.13.3. Walidacja integracji danych

```
-- Sprawdzenie duplikatów UserId
SELECT UserId, COUNT(*) AS DupCount
FROM core.SyncUserProfiles
GROUP BY UserId
HAVING COUNT(*) > 1;

-- Sprawdzenie duplikatów OrgUnitId
```

```sql
SELECT OrgUnitId, COUNT(*) AS DupCount
FROM core.SyncOUInstances
GROUP BY OrgUnitId
HAVING COUNT(*) > 1;

-- Sprawdzenie, czy ParentId w OU wskazują na istniejące OrgUnitId
SELECT DISTINCT ParentId
FROM core.SyncOUInstances
WHERE ParentId IS NOT NULL
  AND ParentId NOT IN (SELECT OrgUnitId FROM core.SyncOUInstances);

-- Sprawdzenie, czy wszyscy użytkownicy w przypisaniach istnieją
SELECT DISTINCT u.UserId
FROM core.SyncOUInstanceUsers u
WHERE u.UserId NOT IN (SELECT UserId FROM core.SyncUserProfiles);

-- Sprawdzenie, czy wszystkie OU w przypisaniach istnieją
SELECT DISTINCT o.OrgUnitId
FROM core.SyncOUInstanceUsers o
WHERE o.OrgUnitId NOT IN (SELECT OrgUnitId FROM core.SyncOUInstances);
```

9.13.4. Resetowanie statusu synchronizacji

```sql
-- Ustaw status na ReadyForSync (2) aby wyzwolić synchronizację w następnym cyklu timera
UPDATE core.SqlTableStatuses SET Status = 2 WHERE Name = 'SyncUserProfiles'
UPDATE core.SqlTableStatuses SET Status = 2 WHERE Name = 'SyncOUInstances'
```

9.13.5. Monitoring timera TaskService i konfiguracji

Aby sprawdzić `RepeatInterval` w konfiguracji tenanta:

- Szukaj sekcji `StructureSynchronizationConfiguration`
- Parametr: `RepeatInterval`
- Wartość w sekundach — jak często timer sprawdza `SqlTableStatuses`
- Po zmianie `RepeatInterval`: zrestartuj TaskService

9.13.6. Wyzwolenie synchronizacji (manual trigger)

```sql
-- 1. Załaduj dane (przykład)
DELETE FROM [core].[SyncOUInstanceUsers];
```

```sql
DELETE FROM [core].[SyncUserProfiles];
DELETE FROM [core].[SyncOUInstances];

INSERT INTO [core].[SyncUserProfiles] (UserId, UserName, FirstName, LastName, Email, IsActive, ExpireDate)
VALUES ('user001', 'john.doe', 'John', 'Doe', 'john@company.com', 1, '2026-12-31');

INSERT INTO [core].[SyncOUInstances] (OrgUnitId, Code, Name, IsActive, ActiveFrom, ActiveUntil, ParentId)
VALUES ('IT001', 'IT', 'Information Technology', 1, '2026-01-01', NULL, NULL);

-- 2. Ustaw status na ReadyForSync (2)
UPDATE [core].[SqlTableStatuses] SET Status = 2 WHERE Name = 'SyncUserProfiles';
UPDATE [core].[SqlTableStatuses] SET Status = 2 WHERE Name = 'SyncOUInstances';

-- 3. Czekaj RepeatInterval sekund — timer TaskService wyzwoli synchronizację
-- 4. Sprawdzaj status:
SELECT Name, Status,
  CASE Status
    WHEN 0 THEN 'ReadyForData (gotowe)'
    WHEN 1 THEN 'DataInput (ładowanie)'
    WHEN 2 THEN 'ReadyForSync (czeka na timer)'
    WHEN 3 THEN 'InSync (synchronizacja w trakcie)'
    WHEN 4 THEN 'Error (błąd)'
  END AS StatusText
FROM [core].[SqlTableStatuses]
WHERE Name IN ('SyncUserProfiles', 'SyncOUInstances');
```

9.13.7. Czyszczenie tabel (UWAGA: nieodwracalne!)

```sql
-- Usuń wszystkie dane z tabel synchronizacyjnych
-- UWAGA: Najpierw wykonaj kopię zapasową!
DELETE FROM core.SyncOUInstanceUsers;
DELETE FROM core.SyncUserProfiles;
DELETE FROM core.SyncOUInstances;
```

BEZPIECZNE PRAKTYKI

- **Zawsze wykonaj kopię zapasową** przed uruchomieniem synchronizacji w trybie Full
- **Testuj na środowisku testowym** przed wdrożeniem na produkcji
- **Monitoruj logi TaskService** — status `ReadyForData` nie gwarantuje powodzenia synchronizacji; zawsze sprawdzaj logi pod kątem błędów
- **Sprawdzaj liczę rekordów** po synchronizacji — porównaj dane w tabelach przejściowych z tabelami docelowymi
- **Dokumentuj** zmiany — kiedy przeprowadzono synchronizację, w jakim trybie, jakie były wyniki (szczególnie czy były błędy w logach)
- **Waliduj dane źródłowe** przed załadowaniem do tabel przejściowych — brakujące lub nieprawidłowe dane mogą prowadzić do niespójności struktury
- **Regularne przeglądy logów** — ustaw proces monitorowania logów TaskService (np. codzienne przeglądy, alerting na błędy)

Powiązane tematy:

- Użytkownicy
- Jednostki organizacyjne
- Zarządzanie danymi

9.14. MODEL UPRAWNIEŃ PBA

Podsumowanie:
Model uprawnień PBA obejmuje uprawnienia do wykonywania określonych czynności w systemie przypisywane rolom biznesowym i indywidualnym użytkownikom. Te uprawnienia są przypisywane zarówno w aplikacji *FrontSPA*, jak i w aplikacji *AdminSPA*.

☛
FrontSPA > `ZARZĄDZANIE UPRAWNIENIAMI` > `Administracja`
FrontSPA > `ZARZĄDZANIE UPRAWNIENIAMI` > `Systemowe` (użytkownik *superuser*)

AdminSPA > `ADMINISTRACJA` > `Lista aplikacji` >

Model uprawnień *Permissions Based Authorization* (autoryzacja oparta na uprawnieniach) pozwala kontrolować dostęp użytkowników do wykonywania określonych czynności w systemie na podstawie posiadanych przez nich uprawnień. Proces przyznawania uprawnień

odbywa się zarówno w aplikacji *FrontSPA*, jak i w aplikacji *AdminSPA*, gdzie są nadawane uprawnienia do budowania poszczególnych aplikacji. Te ostatnie opisano w rozdziale poświęconym aplikacjom.

- Uprawnienia mogą być przyznawane rolom biznesowym oraz indywidualnym użytkownikom.
- Uprawnienia wynikowe każdego użytkownika stanowią logiczną sumę uprawnień przypisanych w ramach każdej jego roli oraz indywidualnie jego profilowi; to znaczy, że aby pozbawić użytkownika danego uprawnienia, należy je wyłączyć zarówno w jego profilu, jak i we wszystkich jego rolach.
- Weryfikacja uprawnień odbywa się podczas logowania użytkownika (zmiana uprawnień zaczyna obowiązywać po ponownym zalogowaniu użytkownika).
- Zarządzanie uprawnieniami w modelu PBA odbywa się na dwóch poziomach:
    - Poziom administracji uprawnieniami: wyznaczanie użytkowników (ról), którzy będą mogli definiować uprawnienia innych użytkowników (administratorzy uprawnień)
    - Poziom przypisywania uprawnień: przypisywanie rolom i użytkownikom uprawnień z predefiniowanych obszarów uprawnień w nAxiom, np. uprawnień powiązanych z zarządzaniem użytkownikami, uprawnień do korzystania z aplikacji, do rozbudowy aplikacji itp.

Powiązane tematy:

- Uprawnienia PBA do budowania aplikacji

9.14.1. Inicjowanie modelu uprawnień PBA

Podsumowanie:
Po zainstalowaniu witryny nAxiom w nowym środowisku konieczne jest zainicjowanie modelu uprawnień PBA. Początkowa konfiguracja obejmuje profil użytkownika *superuser*, który ma tylko uprawnienie do wyznaczania administratorów uprawnień. Ponadto dostępne są dwa predefiniowane profile użytkowników „systemowych”: *admin* i *consultant*, które mają przypisane uprawnienia umożliwiające im pracę w standardowym zakresie.

W celu zainicjowania modelu PBA należy skorzystać z predefiniowanego profilu użytkownika *superuser* (hasło jest dostarczane z pakietem instalacyjnym). Ten profil odgrywa rolę *superadministratora uprawnień*, to znaczy może wyznaczać użytkowników (w tym siebie) jako administratorów uprawnień. Temu użytkownikowi nie można odebrać uprawnień do administracji uprawnieniami.

W szkicowaniu scenariuszy inicjowania modelu uprawnień PBA można wyróżnić dwa podstawowe przypadki:

- środowisko nAxiom z pustą bazą danych, tj. brak profili użytkowników innych niż systemowe
- środowisko nAxiom z bazą danych zawierającą profile użytkowników i role biznesowe

Pierwszy przypadek jest bardziej złożony, dlatego zostanie opisany poniżej. W drugim przypadku wystarczy pominąć kroki związane z definiowaniem ról użytkownika i profili użytkowników.

> **UWAGA**
> Uprawnienia PBA są weryfikowane podczas logowania użytkownika, dlatego każda zmiana uprawnień zalogowanego użytkownika zacznie obowiązywać dopiero po jego wylogowaniu i ponownym zalogowaniu.

1. Użytkownik *consultant* loguje się do back-office, gdzie definiuje co najmniej jedną aplikację i zestaw ról biznesowych; ewentualnie może być konieczne włączenie dla określonych ról biznesowych systemowych pozycji menu front-office, np. `Użytkownicy`, `Uprawnienia` itp.
2. Użytkownik *admin* loguje się do front-office i definiuje profile użytkowników.
3. W razie potrzeby *superuser* może wyznaczyć kolejnych administratorów uprawnień.
4. Administratorzy uprawnień przypisują rolom biznesowym i użytkownikom role z obszarów uprawnień.

WAŻNE

- Bez uprawnienia `Używanie wszystkich aplikacji` (`Uprawnienia` > `Użytkowanie aplikacji`) inne posiadane przez użytkownika uprawnienia będą nieskuteczne, z wyjątkiem uprawnienia `Zmiana własnego hasła`.
- Użytkownicy, którzy mają zarządzać profilami użytkowników, muszą mieć przypisaną rolę systemową *Administrator* albo rolę biznesową z dostępem do polecenia `Lista użytkowników` (ewentualnie `Drzewo listy użytkowników`).
- Analogicznie, administratorzy uprawnień będą mieli dostęp do obszarów uprawnień po przypisaniu im ról biznesowych z włączonym dostępem do menu `Zarządzanie uprawnieniami` > `Administracja` / `Uprawnienia`; domyślnie to menu jest dostępne tylko dla posiadaczy roli systemowej `Administrator`.
- Dostęp do back-office (zarządzanie definicjami aplikacji, rozbudowa aplikacji) wymaga posiadania roli systemowej *Konsultant*.
- Uprawnieniem podstawowym, wymaganym przez prawie wszystkie uprawnienia z obszaru `Budowanie aplikacji` i `Konfiguracja systemu` jest uprawnienie `Przegląd aplikacji w module AdminSPA` (`Uprawnienia` > `Budowanie aplikacji`); te obszary uprawnień są przeznaczone dla użytkowników z przypisaną rolą systemową *Konsultant*.
- Jeśli podczas instalowania witryny nie zostanie wskazana licencja, konieczne jest zaimportowanie licencji po zalogowaniu na profil użytkownika systemowego *consultant*. Dopiero wtedy można będzie zalogować się na konto *superuser* i zarządzać uprawnieniami.
- Użytkownik *superuser* ma na stałe przypisane uprawnienie do wyznaczania administratorów uprawnień (pole wyboru `Dostęp do administracji uprawnieniami`) i nie można tego zmienić. Po zainicjowaniu uprawnień PBA ten profil można dezaktywować.

Powiązane tematy:

- Listy użytkowników
-

9.14.2. Wyznaczanie administratora uprawnień PBA

Podsumowanie:
Administrator uprawnień to użytkownik, który może nadawać uprawnienia innym użytkownikom. W modelu PBA można wyróżnić dwie kategorie administratorów uprawnień:

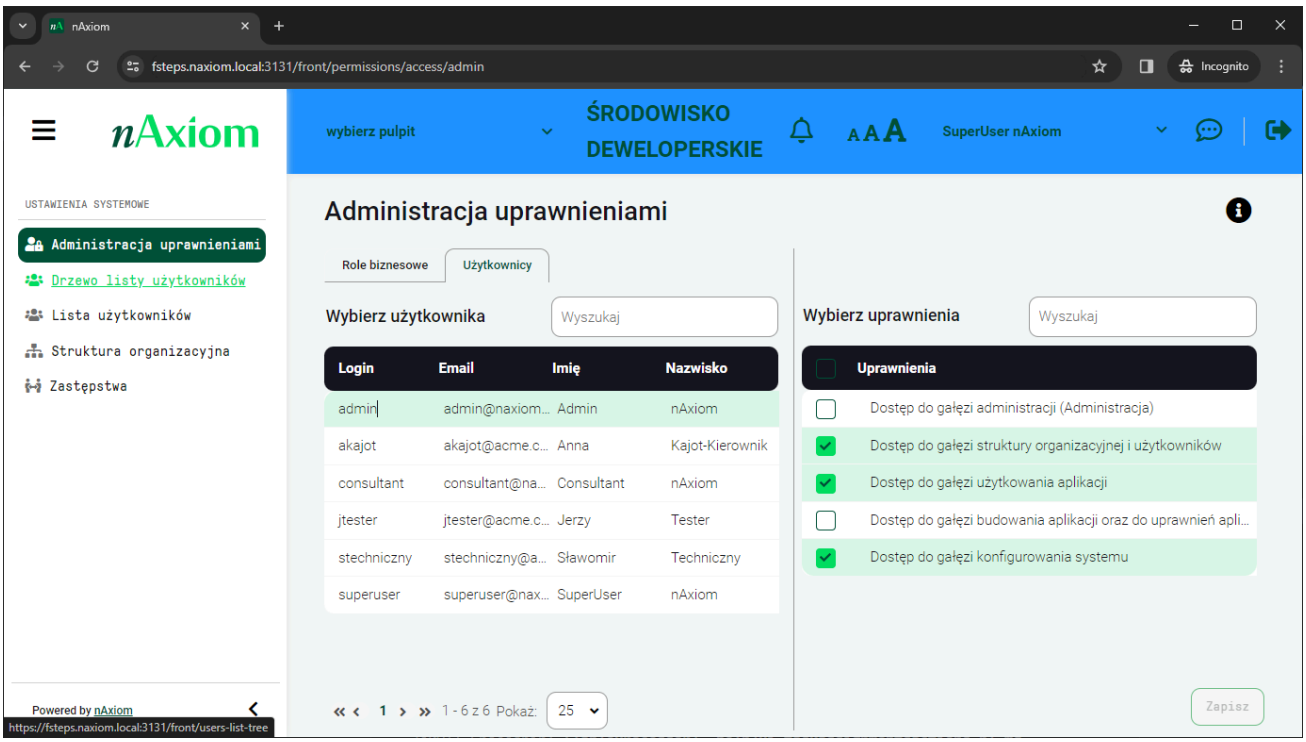
- superadministrator uprawnień: może wyznaczać innych administratorów uprawnień
- administrator uprawnień w danym obszarze: może przypisywać innym użytkownikom uprawnienia z tego obszaru

☛ *FrontSPA* (*superuser*) > `ZARZĄDZANIE UPRAWNIENIAMI` > `Administracja`

Aby wyznaczyć administratora uprawnień:

1. Zaloguj się, korzystając z profilu *superuser* lub z profilu użytkownika, który ma uprawnienia do administrowania uprawnieniami.
2. Kliknij polecenie `Administracja` w sekcji `ZARZĄDZANIE UPRAWNIENIAMI` w menu po lewej stronie. Zostanie wyświetlony ekran `Administracja uprawnieniami`.
3. Kliknij pozycję na karcie `Role biznesowe` lub kliknij kartę `Użytkownicy` i kliknij profil użytkownika, któremu chcesz nadać uprawnienia. Z prawej strony zostanie wyświetlona lista dostępnych uprawnień.
4. Zaznacz pola wyboru obok odpowiednich uprawnień i kliknij przycisk `Zapisz`.

UWAGA
Zmiany uprawnień należy zapisywać indywidualnie dla każdej roli/użytkownika.



9.14.3. Uprawnienia administracyjne

Po kliknięciu polecenia `Administracja` w sekcji `ZARZĄDZANIE UPRAWNIENIAMI` i wybraniu roli biznesowej lub użytkownika zostają wyświetlone pola wyboru uprawnień administracyjnych:

- `Dostęp do gałęzi administracji (Administracja)`: *superadministrator uprawnień*, może wyznaczać innych superadministratorów uprawnień oraz administratorów w obszarach uprawnień.
- `Dostęp do gałęzi struktury organizacyjnej i użytkowników`: administrator uprawnień w obszarze `Zarządzanie strukturą organizacyjną i użytkownikami`.
- `Dostęp do gałęzi użytkowania aplikacji`: administrator uprawnień w obszarze `Użytkowanie aplikacji`.
- `Dostęp do gałęzi budowania aplikacji oraz do uprawnień aplikacji`: administrator uprawnień w obszarze `Budowanie aplikacji`, administrator uprawnień do indywidualnych aplikacji w *back-office*.
- `Dostęp do gałęzi konfigurowania systemu`: administrator uprawnień w obszarze `Konfiguracja systemu`.

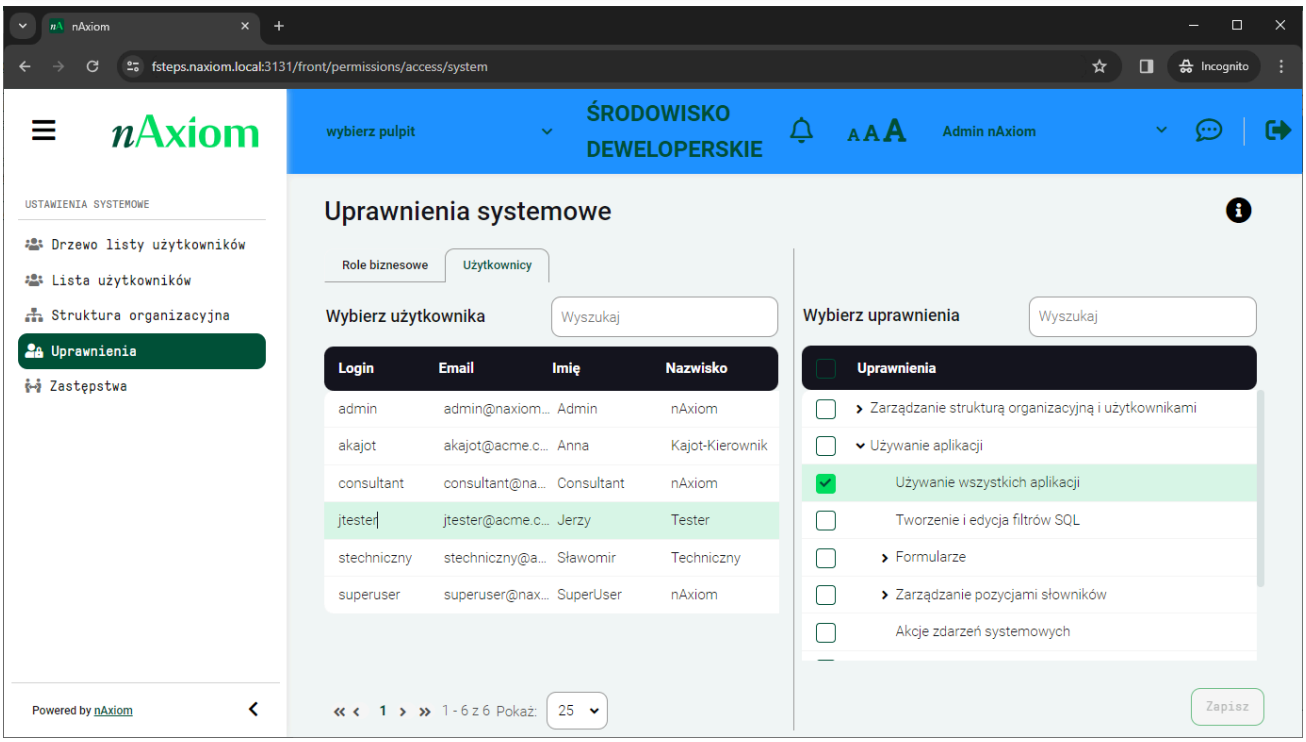
9.14.4. Ustawianie uprawnień ról/użytkowników

Podsumowanie:
Zarządzanie uprawnieniami PBA użytkowników wymaga posiadania uprawnień administracyjnych: `Dostęp do gałęzi struktury organizacyjnej i użytkowników`, `Dostęp do gałęzi użytkowania aplikacji` oraz (opcjonalnie) `Dostęp do gałęzi konfigurowania systemu`.

Aby ustawić uprawnienia roli/użytkownika:

1. Zaloguj się, korzystając z profilu użytkownika, który jest administratorem uprawnień w co najmniej jednym z obszarów uprawnień.
2. Kliknij polecenie `Uprawnienia` w sekcji `ZARZĄDZANIE UPRAWNIENIAMI` w menu po lewej stronie. Zostanie wyświetlony ekran `Uprawnienia systemowe`.
3. Kliknij pozycję na karcie `Role biznesowe` lub kliknij kartę `Użytkownicy` i kliknij profil użytkownika, któremu chcesz nadać uprawnienia. Z prawej strony zostanie wyświetlona lista dostępnych uprawnień.
4. Zaznacz pola wyboru obok odpowiednich uprawnień i kliknij przycisk `Zapisz`.

UWAGA
Zmiany uprawnień należy zapisywać indywidualnie dla każdej roli/użytkownika.



9.14.5. Obszar Zarządzanie strukturą organizacyjną i użytkownikami

Podsumowanie:
`Zarządzanie strukturą organizacyjną i użytkownikami` to obszar grupujący uprawnienia do zarządzania jednostkami struktury organizacyjnej i profilami użytkowników; uprawnienia z tego obszaru mają zastosowanie tylko do użytkowników, którzy mają dostęp do poleceń `Lista użytkowników` i/lub `Drzewo listy użytkowników` oraz do struktury organizacyjnej (we FrontSPA i/lub w AdminSPA).

Obszar obejmuje następujące uprawnienia i grupy uprawnień:

Grupa `Zarządzanie strukturą organizacyjną`

Podgrupa uprawnień/ uprawnienie	Komentarz	Wymagane uprawnienia
`Tworzenie jednostek organizacyjnych`	*front-office/back-office*	`Podgląd jednostek organizacyjnych`, `Przegląd aplikacji w module AdminSPA` (*Konsultant*), `Używanie wszystkich aplikacji` (*Klient*)
`Podgląd jednostek organizacyjnych`	*front-office/back-office*	`Przegląd aplikacji w module AdminSPA` (*Konsultant*), `Używanie wszystkich aplikacji` (*Klient*)
`Aktualizacja jednostek organizacyjnych`	*front-office/back-office*	`Podgląd jednostek organizacyjnych`, `Przegląd aplikacji w module AdminSPA` (*Konsultant*), `Używanie wszystkich aplikacji` (*Klient*)
`Usuwanie jednostek organizacyjnych`	*front-office/back-office*	`Podgląd jednostek organizacyjnych`, `Przegląd aplikacji w module AdminSPA` (*Konsultant*), `Używanie wszystkich aplikacji` (*Klient*)

Grupa `Zarządzanie użytkownikami`

UWAGA

Uprawnienia w tej grupie określają dostęp zarówno do natywnych funkcjonalności zarządzania profilami użytkowników (polecenia `Lista użytkowników` i `Drzewo listy użytkowników`) dostępnych we front-office, jak i do końcowych API udostępniających funkcje związane z profilami użytkowników.

Podgrupa uprawnień/ uprawnienie	Komentarz	Wymagane uprawnienia
`Przeglądanie listy użytkowników`	dostęp do polecenia `Lista użytkowników` we *front-office*	
`Przeglądanie drzewa użytkowników`	dostęp do polecenia `Drzewo listy użytkowników` we *front-office*	
`Tworzenie nowego użytkownika`	dostęp do przycisku `Dodaj użytkownika` na liście użytkowników; do wyświetlenia okna dialogowego `Dodanie profilu użytkownika` wymagane jest uprawnienie `Podgląd szczegółów dowolnego użytkownika`; posiadanie tylko tych dwóch uprawnień pozwala tworzyć nowe profile użytkownika bez przypisanej roli biznesowej i hasła; możliwe jest utworzenie profilu użytkownika bez hasła, ale na taki profil nie można się zalogować (do wykorzystania w scenariuszach, kiedy inna osoba tworzy profil, a inna nadaje hasło)	`Używanie wszystkich aplikacji`
`Zmiana ról biznesowych użytkownika`	dostęp do listy `Rola biznesowa` na formularzu danych użytkownika	`Podgląd szczegółów dowolnego użytkownika`, `Edycja profilu dowolnego użytkownika`, `Używanie wszystkich aplikacji`
`Podgląd danych użytkowników`		
• `Podgląd szczegółów dowolnego użytkownika`	umożliwia wysyłanie zapytań API o dane użytkowników; pomocniczo wymagane przez uprawnienia `Tworzenie nowego użytkownika`	`Używanie wszystkich aplikacji`

Podgrupa uprawnień/ uprawnienie	Komentarz	Wymagane uprawnienia
• `Podgląd własnego profilu`	umożliwia wysyłanie zapytań API o własne dane; daje dostęp do polecenia `Mój profil` w menu rozwijanym po kliknięciu nazwy użytkownika w aplikacji FrontSPA.	
`Edycja danych użytkowników`		
• `Edycja profilu dowolnego użytkownika`	pozwala modyfikować dane w profilu użytkownika; daje dostęp do przycisków `Edytuj` w ostatniej kolumnie listy użytkowników	`Podgląd szczegółów dowolnego użytkownika`, `Używanie wszystkich aplikacji`
• `Edycja własnego profilu`	pozwala modyfikować dane w swoim profilu użytkownika; daje dostęp do przycisku `Edytuj` w wierszu swojego profilu na liście użytkowników; daje dostęp do polecenia `Mój profil` wyświetlanego po kliknięciu w nazwę logowania	`Używanie wszystkich aplikacji`, `Podgląd własnego profilu`
`Zmiana hasła użytkowników`		
• `Zmiana hasła dowolnego użytkownika`	pozwala zmieniać hasło profilu użytkownika; bez tego uprawnienia pola tekstowe do definiowania hasła są ukryte	`Edycja profilu dowolnego użytkownika`, `Podgląd szczegółów dowolnego użytkownika`, `Używanie wszystkich aplikacji`
• `Zmiana własnego hasła`	pozwala zmieniać własne hasło użytkownika; bez tego uprawnienia przycisk `Zmień hasło` w oknie dialogowym polecenia `Mój profil` jest ukryty	`Podgląd własnego profilu`

Podgrupa uprawnień/ uprawnienie	Komentarz	Wymagane uprawnienia
`Blokowanie konta`	daje dostęp do przycisku blokowania/ odblokowania konta w widokach `Lista użytkowników` i `Drzewo listy użytkowników`	`Przeglądanie listy użytkowników`
`Ustawianie bezterminowych haseł`	pozwala wyłączać profile użytkowników z polityki okresowej zmiany haseł	
`Edycja profilu dowolnego użytkownika`		
`Dostęp do funkcji 'Pobierz przełożonych z LDAP'`	daje dostęp do przycisku `Pobierz przełożonych dla użytkowników LDAP` w nagłówku listy użytkowników	`Przeglądanie listy użytkowników`
`Import użytkowników z pliku excel`	daje dostęp do przycisku `Importuj użytkowników z pliku Excel` w sekcji `Import użytkowników` w ustawieniach systemu w back-office (`ADMINISTRACJA` > `Ustawienia systemu`, wymagana rola systemowa *Konsultant*)	`Przegląd aplikacji w module AdminSPA`, `Dostęp do głównych ustawień systemu`
`Zastępstwa`		
`Definiowanie zastępstw za dowolnego użytkownika`	pozwala definiować zastępstwa za dowolnego użytkownika, w tym za siebie i podwładnych; kiedy to pole wyboru jest zaznaczone, ustawienia pozostałych pól w tej sekcji są ignorowane	`Używanie wszystkich aplikacji`
`Definiowanie zastępstw za siebie i podwładnych`	pozwala definiować zastępstwa za siebie, swoich podwładnych i podwładnych swoich podwładnych; kiedy to pole wyboru jest zaznaczone, ustawienie pola poniżej jest ignorowane	`Używanie wszystkich aplikacji`
`Definiowanie`	pozwala definiować zastępstwa za siebie	`Używanie`

Podgrupa uprawnień/ uprawnienie	Komentarz	Wymagane uprawnienia
`zastępstw za siebie`		`wszystkich aplikacji`

9.14.6. Obszar Użytkowanie aplikacji

`Użytkowanie aplikacji` to obszar grupujący uprawnienia do korzystania z aplikacji biznesowych, dostępu do elementów konfiguracji oraz systemowych pozycji menu i systemowych ikon na formularzu.

Podgrupa uprawnień/ uprawnienie	Komentarz	Wymagane uprawnienia
`Używanie wszystkich aplikacji`	daje dostęp do poleceń menu we *front-office*, do których użytkownik ma dostęp w związku z przypisanymi mu rolami biznesowymi; podstawowe uprawnienie wymagane do pracy w nAxiom	
`Zarządzanie pulpitami użytkownika`		
• `Dostęp do listy pulpitów`	określa dostęp do polecenia `Lista pulpitów` na rozwijanej liście pulpitów w nagłówku FrontSPA; jeśli użytkownik ma to uprawnienie i nie ma żadnych uprawnień do tworzenia, może edytować pulpity, których jest właścicielem, ale bez możliwości zmiany trybu dostępu do tych pulpitów	
• `Tworzenie pulpitów publicznych`	pozwala zapisywać pulpity z typem dostępu `Publiczny`	
• `Tworzenie pulpitów grup`	pozwala zapisywać pulpity z typem dostępu `Grupa`	
• `Tworzenie pulpitów prywatnych`	pozwala zapisywać pulpity z typem dostępu `Prywatny`	
`Tworzenie i edycja filtrów SQL`	umożliwia definiowanie i edytowanie filtrów dla list jako zapytań SQL (*FrontSPA* > *NazwaListy* > Lista rozwijana wyboru filtra > `Ustawienia` > `Dodaj nowy filtr predefiniowany`); nie jest wymagane do korzystania z filtrów tego rodzaju	`Używanie wszystkich aplikacji`
`Formularze`		
`Opcje systemowe`		

Podgrupa uprawnień/ uprawnienie	Komentarz	Wymagane uprawnienia
• `Podgląd diagramu procesu`	daje dostęp do ikony diagramu procesu* w nagłówku formularza, umożliwiając wyświetlenie diagramu procesu dla bieżącego dokumentu	`Używanie wszystkich aplikacji`
• `Podgląd modelu danych`	daje dostęp do ikony kontekstu formularza** w nagłówku formularza, umożliwiając wyświetlenie diagramu procesu dla bieżącego dokumentu	`Używanie wszystkich aplikacji`
• `Dostęp do audytu dokumentu`	daje dostęp do ikony audytu*** w nagłówku formularza, umożliwiając wyświetlenie informacji audytowych dla bieżącego dokumentu	`Używanie wszystkich aplikacji`
`Zarządzanie pozycjami słowników`		
• `Tworzenie pozycji`	pozwala dodawać nowe pozycje do słowników	`Używanie wszystkich aplikacji`
• `Podgląd listy słowników i ich pozycji`	pozwala przeglądać listę słowników i listę pozycji każdego słownika	`Używanie wszystkich aplikacji`
• `Aktualizacja pozycji`	pozwala na edycję i sterowanie ochroną pozycji słownika	`Używanie wszystkich aplikacji`
• `Usuwanie pozycji`	pozwala usuwać pozycje słownika	`Używanie wszystkich aplikacji`
`Akcje zdarzeń systemowych`	pozwala na sterowanie akcjami zdarzeń systemowych	`Używanie wszystkich aplikacji`
`Import z MS Excel - mapowanie kolumn`	pozwala na wprowadzanie konfiguracji dla importu z MS Excel	`Używanie wszystkich aplikacji`

Podgrupa uprawnień/ uprawnienie	Komentarz	Wymagane uprawnienia
`OCR`		(wymagana licencja OCR)
• `Szablony`	pozwala na zarządzanie szablonami OCR	`Używanie wszystkich aplikacji`
• `Niezidentyfikowane dokumenty`	pozwala na przeglądanie dokumentów niezidentyfikowanych przez OCR	`Używanie wszystkich aplikacji`

*

**

***

9.14.7. Obszar Konfiguracja systemu

`Konfiguracja systemu` to obszar grupujący uprawnienia do ustawień konfiguracyjnych dostępnych w *AdminSPA*. Uprawnienia te dotyczą użytkowników z rolą systemową *Konsultant*. Dodatkowo, obszar ten zawiera ustawienia dotyczące trybu serwisowego, który jest dostępny w *FrontSPA*.

> **WSKAZÓWKA**
> Domyślnie w nAxiom jest dostępny jeden profil konsultanta. Procedurę tworzenia kolejnych opisano w temacie Tworzenie nowego konsultanta .

Podgrupa uprawnień/uprawnienie	Komentarz	Wymagane uprawnienia
`Dostęp do głównych ustawień systemu`	daje dostęp do polecenia `Ustawienia systemowe` w *back-office*, wymagana rola *Konsultant*	`Przegląd aplikacji w module AdminSPA`
`Dostęp do narzędzia Kreator stylów`	daje dostęp do polecenia `Kreator stylów systemu` w *back-office*, wymagana rola *Konsultant*	`Przegląd aplikacji w module AdminSPA`
`Dostęp do logów systemu`	daje dostęp do polecenia `Logi środowiska i bieżącego tenanta` w *back-office*, wymagana rola *Konsultant*	`Przegląd aplikacji w module AdminSPA`
`Tryb serwisowy`	daje dostęp do przełącznika `Tryb serwisowy` w menu wyświetlanym po kliknięciu imienia i nazwiska użytkownika na pasku nagłówka na pulpicie	
`Tryb serwisowy – podgląd szczegółów`	daje dostęp do podglądu szczegółów informacji logowanych w logach serwisowych	
`Dostęp do definiowania repozytoriów w systemie`	daje dostęp do przycisku `Nowe repozytorium` na stronie `Repozytoria załączników` w *back-office*, wymagana rola *Konsultant*	`Przegląd aplikacji w module AdminSPA`
`Dostęp do zarządzania wtyczkami systemu przeznaczonymi do używania w obrębie innych aplikacji`	daje dostęp do polecenia `Wtyczki dla Microsoft Office` w *back-office*, wymagana rola *Konsultant*	`Przegląd aplikacji w module AdminSPA`

9.14.8. Obszar Budowanie aplikacji

`Budowanie aplikacji` to obszar grupujący globalne uprawnienia do tworzenia definicji aplikacji oraz określonych funkcjonalności menu *AdminSPA*. Uprawnienia te dotyczą użytkowników z rolą systemową *Konsultant*.

WSKAZÓWKA

Domyślnie w nAxiom jest dostępny jeden profil konsultanta. Procedurę tworzenia kolejnych opisano w temacie Tworzenie nowego konsultanta .

Podgrupa uprawnień/ uprawnienie	Komentarz	Wymagane uprawnienia
`Tworzenie definicji aplikacji`	daje dostęp do przycisku `Dodaj definicję` na stronie `Lista aplikacji` w *AdminSPA*	`Przegląd aplikacji w module AdminSPA`
`Dostęp do Kreatora modeli danych`	daje dostęp do polecenia `Kreator modeli danych` w *AdminSPA*	`Przegląd aplikacji w module AdminSPA`
`Dostęp do Designera raportów Telerik`	daje dostęp do przycisku `Designer` na stronie `Szablony raportów` (`NARZĘDZIA` > `Szablony` > `Szablony raportów`) w *AdminSPA*	`Przegląd aplikacji w module AdminSPA`
`Generowanie dokumentacji aplikacji`	daje dostęp do polecenia `Generator dokumentacji` w *AdminSPA*	`Przegląd aplikacji w module AdminSPA`
`Migrowanie aplikacji między systemami`	daje dostęp do polecenia `Migrator aplikacji` w *AdminSPA*	`Przegląd aplikacji w module AdminSPA`
`Pozostałe pozycje menu projektowania systemu`	daje dostęp do poleceń menu w *AdminSPA*	`Przegląd aplikacji w module AdminSPA`
`Przegląd aplikacji w module AdminSPA`	podstawowe uprawnienie wymagane do pracy w *AdminSPA*; pozwala jedynie przeglądać listy obiektów zdefiniowanych w *AdminSPA*; to uprawnienie jest także wymagane do konfigurowania akcji dla zdarzenia logowania użytkownika we *FrontSPA* (akcje zdarzeń systemowych)	

9.15. UPRAWNIENIA ACL DO DOKUMENTÓW

Podsumowanie:
Uprawnienia ACL określają prawo do tworzenia, odczytu, aktualizacji i usuwania dokumentów (**C** - *create*,odczytu, aktualizacji i usuwania dokumentów (**R** - *read*, **U** - *update* i **D** - *delete*) oraz prawo do administrowania tymi uprawnieniami — to jest określania uprawnień ACL dla indywidualnych dokumentów z poziomu formularza w aplikacji *FrontSPA*.

Uprawnienia ACL dla dokumentów definiuje się w formie szablonów uprawnień w oknie definicji dokumentu (indywidualnie dla każdego statusu oraz globalnie, bez względu na status). Podmiotem uprawnień (uprawnionym) może być twórca dokumentu, posiadacz roli biznesowej lub członek grupy interesariuszy.

Uprawnienia ACL dla formularzy określają dostęp do danych na poziomie kolumn tabeli dokumentu prezentowanych przez kontrolki na formularzach. Te uprawnienia opisano osobno.

Uprawnienia ACL zaimplementowane w nAxiom służą do kontroli podstawowych operacji na dokumentach (rekordach tabeli modelu danych): tworzenia, odczytu, modyfikowania i usuwania. Podstawowy sposób określania tych uprawnień to *szablony uprawnień* tworzone na poziomie definicji dokumentu. Dla każdego nowego dokumentu tworzone są wpisy uprawnień zgodne z szablonami uprawnień.

Dodatkowo w nAxiom wprowadzono uprawnienie do administrowania uprawnieniami, to jest do modyfikowania uprawnień dla indywidualnych dokumentów przez autoryzowanych użytkowników końcowych aplikacji po stronie front-office. Można wyróżnić dwie podstawowe kategorie takich szablonów: szablony globalne, które obowiązują wszystkie dokumenty w danej klasie bez względu na ich status oraz szablony statusów obowiązujące dokumentu z danej klasy we wskazanym statusie.

Podmiotami uprawnień przyznawanych w szablonie uprawnień, czyli osobami uprawnionymi, mogą być:

- twórca dokumentu,
- członek grupy *rola biznesowa*, czyli użytkownik, który ma przypisaną wskazaną rolę biznesową.
- członek grupy interesariuszy.

Korzystanie z uprawnień ACL w nAxiom wymaga włączenia ich obsługi w oknie definicji dokumentu oraz włączenia weryfikacji w ustawieniach widoku listy utworzonego dla danej definicji dokumentu. Następnie należy zdefiniować szablony uprawnień. Szablony dla twórcy

i dla ról mają charakter statyczny — obowiązują w stosunku do wszystkich dokumentów tworzonych w ramach danej definicji.

Szablony dla interesariuszy są najbardziej elastyczne, ponieważ skład grupy interesariuszy może być różny dla poszczególnych dokumentów i dodatkowo może się zmieniać podczas przetwarzania dokumentu. W tym przypadku istotnym elementem logiki biznesowej jest mechanizm określania członków grupy interesariuszy za pomocą akcji przypisywania do interesariuszy lub specjalnej kontrolki na formularzu.

Użytkownicy z uprawnieniem do administrowania mogą zmieniać uprawnienia ACL na poziomie indywidualnych dokumentów. Kliknięcie ikony kluczyka w prawym górnym rogu formularza wyświetla odpowiednie okno dialogowe.

Kolejny krok w stronę bardziej precyzyjnej i elastycznej kontroli dostępu do danych to szablony uprawnień definiowane na poziomie formularzy w oparciu o ustawienia atrybutów widoczności, dostępności i wymagalności indywidualnych sekcji i pól formularza. Szablony te definiuje się na karcie `Uprawnienia` w kreatorze formularza. Ta karta jest dostępna tylko wtedy, kiedy włączono obsługę uprawnień na poziomie dokumentu biznesowego. Podmiotem tych szablonów, podobnie jak w przypadku szablonów uprawnień dokumentu biznesowego może być twórca dokumentu, użytkownik z określoną rolą globalną lub członek grupy interesariuszy dla danej definicji dokumentu.

Informacje o uprawnionych i uprawnieniach dla poszczególnych dokumentów są zapisywane w bazie w momencie utworzenia dokumentu. Ponadto wpisy uprawnień interesariuszy są przeliczane w momencie zmiany statusu dokumentu lub w wyniku uruchomienia akcji zmiany statusu.

Powiązane tematy:

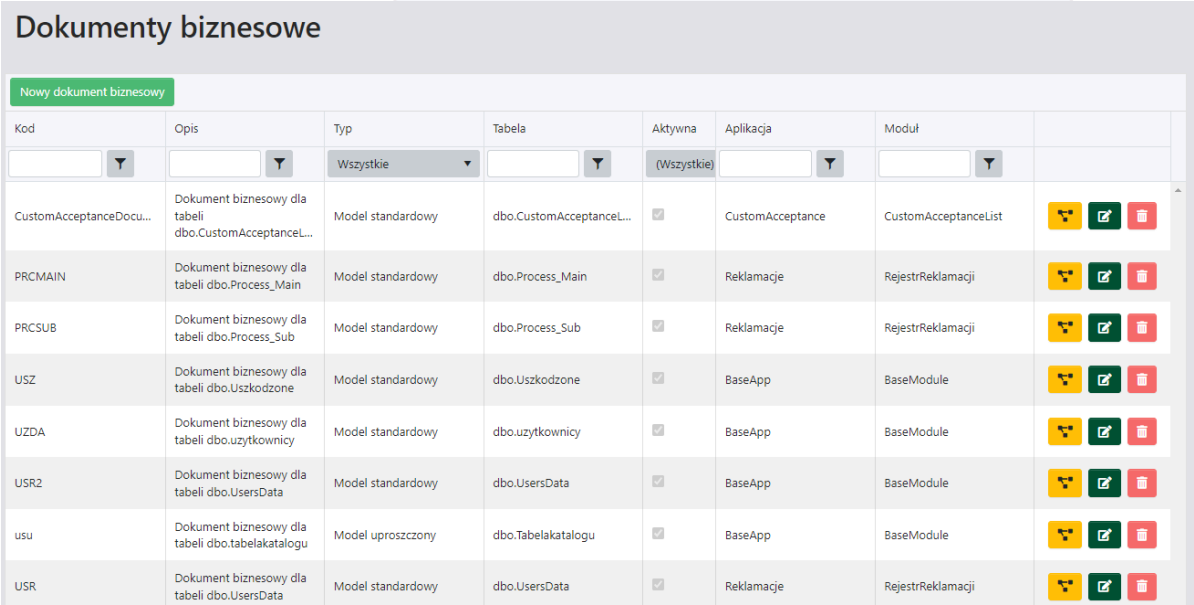
- Włączanie obsługi uprawnień
- Włączanie weryfikacji uprawnień
- Nowy szablon uprawnień do dokumentu
- Administrowanie uprawnieniami ACL
- Mechanika uprawnień ACL (dokumenty)

9.15.1. Włączanie obsługi uprawnień

Podsumowanie:
Dla nowej definicji dokumentu uprawnienia ACL są domyślnie wyłączone. Włącza się je w definicji dokumentu. Dodatkowo należy włączyć weryfikację uprawnień w definicji widoku listy powiązanego z tym dokumentem.

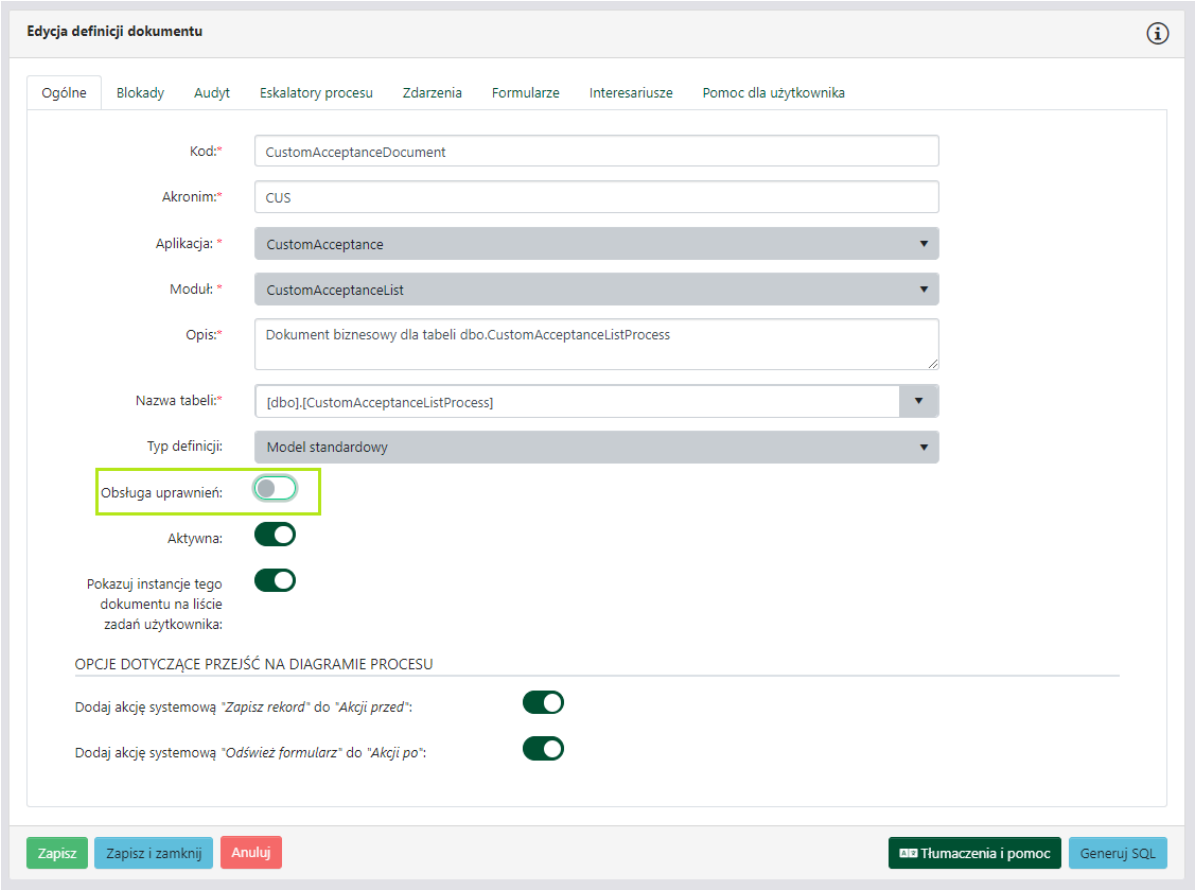
> ☛ `Procesy biznesowe` > `Dokumenty biznesowe` > `Edytuj` > `Ogólne` > `Obsługa uprawnień`

1. Kliknij pozycję `Dokumenty biznesowe` w menu `PROCESY BIZNESOWE`.
   Zostanie wyświetlone okno `Lista definicji dokumentów biznesowych`.



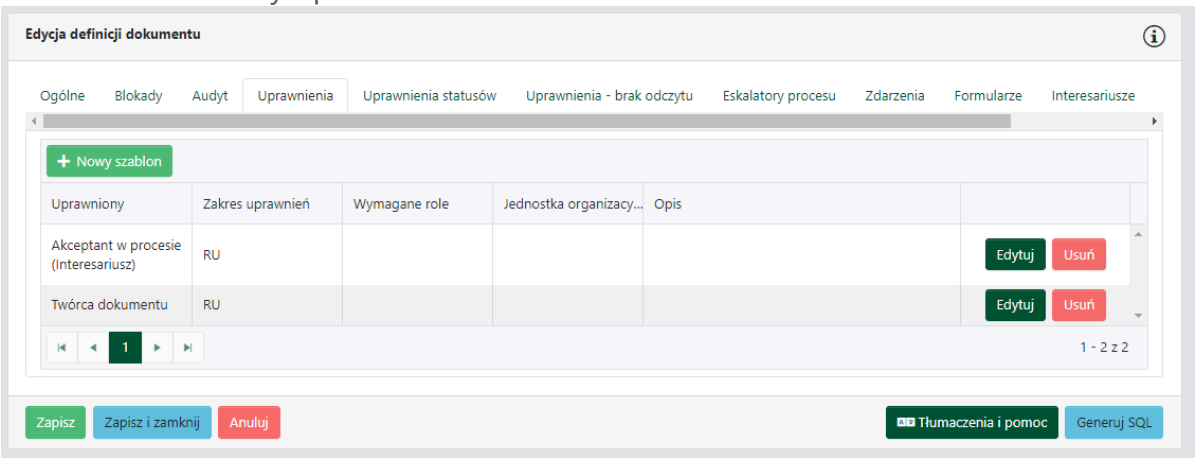
Lista definicji dokumentów

2. Kliknij ikonę (`Edytuj`) w wierszu odpowiedniej definicji dokumentu biznesowego.
   Zostanie wyświetlone okno `Edycja definicji dokumentu`.



Edycja definicji dokumentu

3. Włącz przełącznik `Obsługa uprawnień`.
   Zostaną wyświetlone dwie zakładki: `Uprawnienia` i `Uprawnienia statusów`.
   Kliknięcie zakładki powoduje wyświetlenie odpowiedniej karty, na której można definiować szablony uprawnień.



Karta Uprawnienia, widoczna zakładka Uprawnienia statusów

4. Kliknij przycisk `Zapisz`.

Obsługa uprawnień została włączona.

UWAGA

1. Uprawnienia będą sprawdzane wyłącznie w odniesieniu do dokumentów utworzonych po zdefiniowaniu szablonów uprawnień i po włączeniu weryfikacji uprawnień w definicji listy.
2. Jeśli zostanie włączona obsługa uprawnień, ale nie zostaną zdefiniowane szablony uprawnień, użytkownicy będą mogli tworzyć nowe dokumenty, ale po utworzeniu nie będą one widoczne.
3. Wyłączenie obsługi nie spowoduje usunięcia definiowanych szablonów uprawnień ani nie usunie wpisów uprawnień dla istniejących dokumentów. Po wyłączeniu obsługi ACL dla nowych dokumentów tworzonych w ramach danej definicji nie będą generowane wpisy uprawnień i nie będą sprawdzane uprawnienia dla istniejących dokumentów

9.15.2. Nowy szablon uprawnień do dokumentu

Podsumowanie:
Szablon uprawnień ACL określa uprawnienia do tworzenia, odczytu, modyfikacji, usuwania i administrowania dotyczące wszystkich dokumentów tworzonych w ramach danej definicji dokumentu. Można definiować szablony dla indywidualnych statusów dokumentów oraz szablon ogólny, który będzie obowiązywał w każdym statusie. Osobami uprawnionymi w szablonach dokumentów mogą być: twórca dokumentu, użytkownicy posiadający co najmniej jedną ze wskazanych ról biznesowych oraz użytkownicy należący do grup interesariuszy zdefiniowanych dla dokumentu.

☛ `PROCESY BIZNESOWE` > `Dokumenty biznesowe` > przycisk `Edycja` > karta `Uprawnienia` / `Uprawnienia statusów`

WSKAZÓWKA
Zdefiniowany szablon musi określać co najmniej jedno uprawnienie. Nie można zdefiniować szablonu, w którym wszystkie uprawnienia będą wyłączone.

Uprawnienia do tworzenia dokumentów definiuje się w oknie definicji dokumentu na karcie `Uprawnienia` w sekcji *Szablony uprawnień do tworzenia dokumentu*. To uprawnienie można przypisywać tylko użytkownikom z określoną rolą. W polu wielowyboru można wybrać wiele ról — uprawnienie będzie przypisane do użytkownika, który ma co najmniej jedną rolę z wybranych na liście.

Proces tworzenia szablonów uprawnień obowiązujących w całym cyklu przetwarzania dokumentu jest identyczny jak w przypadku szablonów dla poszczególnych statusów i został opisany poniżej.

1. Wyświetl definicję odpowiedniego dokumentu biznesowego i kliknij zakładkę `Uprawnienia`/`Uprawnienia statusów`.

2. Kliknij przycisk `Nowy szablon`. Jeśli definiujesz szablon na karcie `Uprawnienia statusów`, kliknij przycisk `Nowy szablon` w sekcji statusu, dla którego chcesz utworzyć szablon. Zostanie wyświetlone okno `Szablon dla dokumentu biznesowego`.

   > **WSKAZÓWKA**
   > Elementy sterujące dostępne w tym oknie opisano osobno, patrz sekcja *Powiązane tematy*

3. Wybierz podmiot (twórca dokumentu, rola globalna, grupa interesariuszy), którego ma dotyczyć szablon, wybierając odpowiednie opcje i wartości z list rozwijanych.

   > **WSKAZÓWKA**
   > Jeśli wybierzesz kilka ról biznesowych, uprawnienia będą dotyczyć użytkowników, którzy mają przypisaną co najmniej jedną z wybranych ról.

4. Wybierz zakres uprawnień, zaznaczając odpowiednie pola wyboru.

   > **WSKAZÓWKA**
   > - Uprawnienie do aktualizacji (**U** — *update*) dokumentu jest walidowane w momencie próby zapisu zmienionego dokumentu; nawet jeśli dany użytkownik nie ma uprawnienia do aktualizacji dokumentu, może edytować wartości w polach formularza, jednak nie może zapisać tych zmian. Aby uniemożliwić edycję wartości, należy skorzystać z uprawnień definiowanych na poziomie formularza, które opisano osobno.
   > - Uprawnienie do usuwania (**D** — *delete*) dokumentów dotyczy próby usunięcia rekordu w trybie inline, co wymaga odpowiedniej konfiguracji w widoku listy. Jest ono egzekwowane również po uruchomieniu akcji systemowej *Usuń rekord*.

5. Opcjonalnie, podaj opis tworzonego szablonu uprawnień w polu `Opis`.

6. Kliknij przycisk `Zapisz`.

Szablon uprawnień został zapisany.

Powiązane tematy:

- Definicja dokumentu, Uprawnienia statusów

9.15.3. Włączanie weryfikacji uprawnień

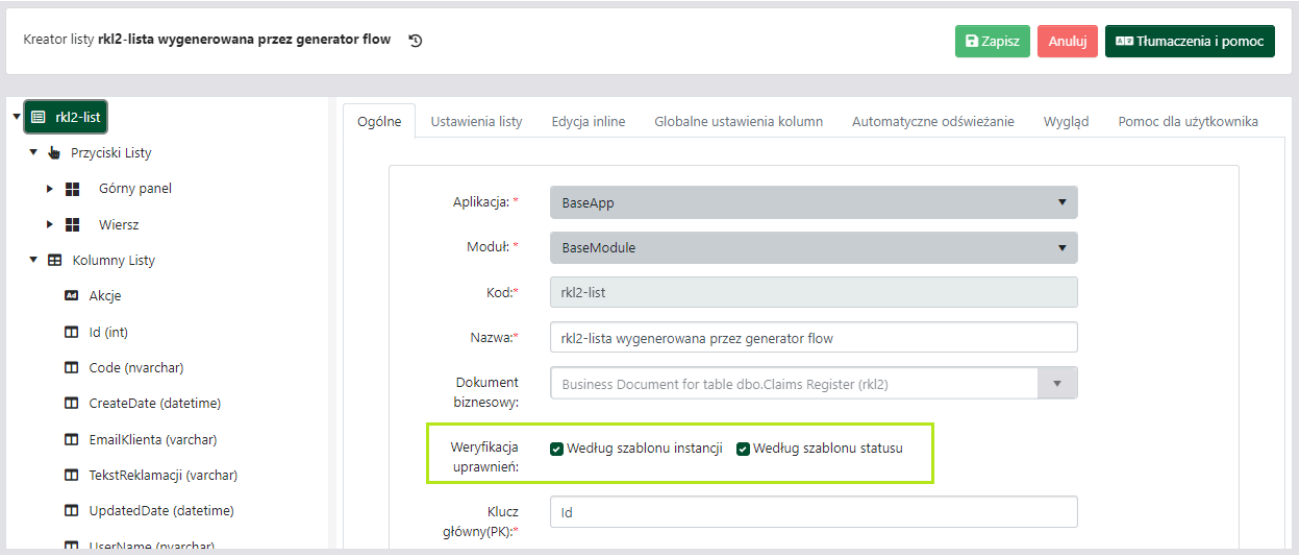
Podsumowanie:
Weryfikację uprawnień ACL włącza się w definicji widoku listy powiązanej z definicją dokumentu. Szablon instancji (bez względu na status) i szablony statusów mają osobne pola wyboru, co daje dodatkowe możliwości konfiguracyjne.

☛ `INTERFEJS UŻYTKOWNIKA` > `Listy` > `Edytuj` > `Ogólne` > `Weryfikacja uprawnień` > `Według szablonu instancji/Według szablonu statusu`

INFO
Pola wyboru do włączania weryfikacji uprawnień są dostępne tylko po włączeniu obsługi uprawnień w definicji dokumentu powiązanej z danym widokiem listy.

1. Wybierz pozycję `Listy` w sekcji `INTERFEJS UŻYTKOWNIKA` w menu back-office.
2. Kliknij przycisk `Edytuj` w wierszu listy, dla której chcesz włączyć weryfikację uprawnień.
3. Na karcie `Ogólne` w grupie `Weryfikacja uprawnień` zaznacz pola wyboru `Według szablonu instancji` i/lub `Według szablonu statusu`.
4. Kliknij przycisk `Zapisz`.

Weryfikacja uprawnień została włączona.



INFO
Przełącznik włączania weryfikacji uprawnień jest dostępny tylko po włączeniu obsługi uprawnień w definicji dokumentu powiązanej z danym widokiem listy.

1. Wybierz pozycję `Listy` w sekcji `INTERFEJS UŻYTKOWNIKA` w menu back-office.
2. Kliknij przycisk `Edytuj` w wierszu listy, dla której chcesz włączyć weryfikację uprawnień.
3. Na karcie `Ogólne` w grupie `Kontrola uprawnień` włącz przełącznik `Weryfikacja uprawnień`.
4. Kliknij przycisk `Zapisz`.

Weryfikacja uprawnień została włączona.

9.15.4. Administrowanie uprawnieniami ACL

Podsumowanie:
Uprawnienie do administrowania uprawnieniami ACL pozwala określa te uprawnienia dla indywidualnych dokumentów z poziomu formularza w aplikacji *FrontSPA*.

Posiadacz uprawnienia do administrowania po wyświetleniu dokumentu w formularzu widzi w prawym górnym rogu ikonę kluczyka . Kliknięcie tej ikony wyświetla okno dialogowe z listą uprawnionych do danej instancji dokumentu.

Lista uprawnionych do indywidualnego dokumentu

Uprawnienia te można edytować i usuwać, można także dodawać nowe uprawnienia.

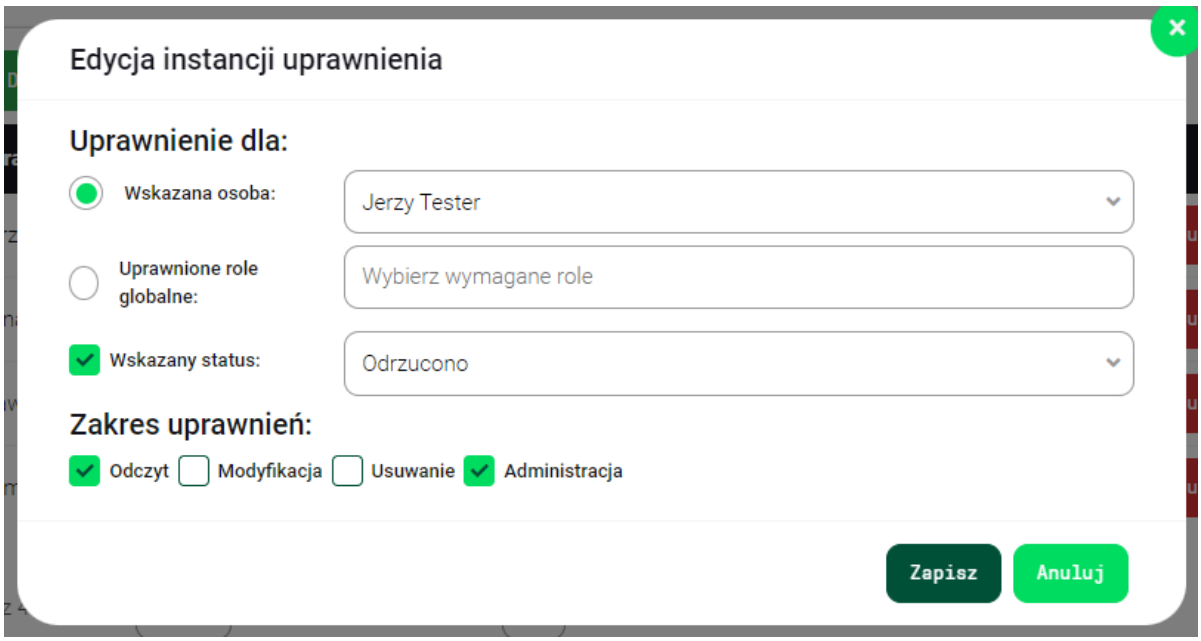
Na poziomie indywidualnego dokumentu podmiotem uprawnienia może być:

- konkretny użytkownik wybierany z listy rozwijanej (opcja `Wskazana osoba`).
- grupa użytkowników z rolą globalną wybierana z listy wielowyboru; w przypadku wyboru kilku ról uprawnienia obejmuje wszystkich użytkowników, którzy mają przypisaną co najmniej jedną z wybranych ról.

Ponadto, po zaznaczeniu pola wyboru `Wskazany status` można wskazać status dokumentu, w którym ma obowiązywać uprawnienie. Jeśli pole nie będzie zaznaczone, uprawnienie będzie dotyczyło dokumentu w każdym statusie.

Zakres uprawnień jest taki sam jak w przypadku szablonów uprawnień:

- `Odczyt`,
- `Modyfikacja`,
- `Usuwanie`,
- `Administracja`.



Edycja uprawnień dla dokumentu

Do automatyzacji przypisywania konkretnym użytkownikom uprawnień do indywidualnych dokumentów służy akcja `Uprawnienia do rekordów`.

Powiązane tematy:

- Uprawnienia do rekordów
- Nowy szablon uprawnień do dokumentu

9.15.5. Mechanika uprawnień ACL (dokumenty)

Podsumowanie:
Podstawą obsługi uprawnień ACL dla dokumentów w nAxiom są szablony uprawnień konfigurowane na poziomie definicji dokumentu. Uprawnienia z szablonów są generowane w momencie utworzenia (pierwszego zapisu) nowego dokumentu. W cyklu przetwarzania dokumentu w procesie biznesowym uprawnienia do indywidualnych dokumentów można zmieniać:

- bezpośrednio, w wyniku ręcznej modyfikacji na formularzu (ikona kluczyka) lub w wyniku wykonania akcji *Uprawnienia do rekordów*
- pośrednio, przez zmianę składu grupy interesariuszy przypisanej do dokumentu.

  Informacje o uprawnieniach są przechowywane w tabelach bazy danych i są podstawą do sprawdzania uprawnień, kiedy użytkownik próbuje wyświetlać, modyfikować i usuwać dokumenty.

Informacje o uprawnieniach są zapisywane w systemowych (schemat *core*) tabelach bazy danych, których nazwy zaczynają się od przedrostka *ACL*. Weryfikacja uprawnień dla każdego dokumentu polega na sprawdzeniu, czy bieżący użytkownik znajduje się w grupie uprawnionych (twórca dokumentu, posiadacz roli, członek grupy interesariuszy) oraz jaki ma zakres uprawnień (*RUDA*).

TABELA *CORE.ACLTEMPLATEENTRIES*

W tej tabeli przechowywane są informacje o zdefiniowanych szablonach. Na podstawie danych z tej tabeli generowane są wpisy uprawnień dla tworzonych dokumentów.

Nazwa kolumny	Typ danych	Zezwól NULL	Opis
Id	int	☐	Kolumna klucza głównego, identyfikator rekordu w tabeli
CRUDA	nvarchar(MAX)	☑	Zakres uprawnień ustawiony w szablonie
UserId	uniqueidentifier	☑	Identyfikator użytkownika, wartość inna niż *NULL* wskazuje, że jest to szablon dla twórcy dokumentu
FillRule	int	☐	Rodzaj uprawnionego, patrz przypis 1 poniżej
Description	nvarchar(MAX)	☑	Opis szablonu uprawnień
BusinessDocumentId	int	☐	Identyfikator definicji dokumentu, z którą jest powiązany szablon
StatusId	int	☑	Identyfikator statusu, w którym ma obowiązywać szablon lub *NULL* dla szablonu ogólnego
OUInstanceId	int	☑	Identyfikator jednostki organizacyjnej; patrz przypis 2 pod tabelą
StakeholderId	int	☑	Identyfikator grupy interesariuszy przypisanej do dokumentu
RowGuid	uniqueidentifier	☐	Unikatowy identyfikator szablonu

Przypis 1: Kolumna *FillRule* może zawierać następujące wartości:

- 0: (nieobsługiwane)

- 1: twórca dokumentu
- 2: właściciel dokumentu (nieobsługiwane)
- 3: przełożony twórcy (nieobsługiwane)
- 4: przełożony właściciela (nieobsługiwane)
- 5: grupa interesariuszy
- 6: użytkownicy z wskazaną rolą
- 7: użytkownicy z wskazaną rolą z wskazanej jednostki organizacyjnej (wycofane)
- 8: użytkownicy z wskazaną rolą z wskazanej jednostki organizacyjnej i jej jednostkę podrzędnych (wycofane)

Przypis 2: W nAxiom 1.11.2.0 wycofano możliwość definiowania szablonów dla ról we wskazanych jednostkach i podjednostkach organizacyjnych, jednak istniejące szablony są nadal obsługiwane. Z tego względu w tabeli pozostawiono kolumnę *OUInstanceId*.

TABELA *CORE.ACLTEMPLATEENTRYROLES*

Tabela pomocnicza, zawiera informacje o rolach powiązanych z poszczególnymi szablonami uprawnień. Ta tabela jest w relacji jeden-do-wiele z tabelą *core.ACLTemplateEntries*.

Nazwa kolumny	Typ danych	Zezwól NULL	Opis
Id	int	☐	Kolumna klucza głównego, identyfikator rekordu w tabeli
ACLTemplateEntryId	int	☐	Identyfikator szablonu uprawnień (kolumna *Id* z tabeli *core.ACLTemplateEntries*)
RoleId	int	☐	Identyfikator roli, dla której obowiązuje schemat uprawnień (kolumna *Id* z tabeli *core.Roles*)
RowGuid	uniqueidentifier	☐	Unikatowy identyfikator rekordu

TABELA *CORE.ACLINSTANCEENTRIES*

Szczegółowe wpisy uprawnień dla poszczególnych dokumentów; każdy dokument może mieć wiele wpisów, np. dla indywidualnych użytkowników, dla ról oraz dla użytkowników należących do grup interesariuszy. Ta tabela zawiera również zduplikowane kolumny systemowe z tabeli modelu danych na potrzeby funkcji tabelarycznej *core.UserTasks_View*. Opis tych kolumn pominięto.

Wpisy w tej tabeli są generowane/usuwane w wyniku następujących zdarzeń:

- Utworzenie dokumentu; po zapisaniu nowego dokumentu i ustawieniu pierwszego statusu zostają wygenerowane wpisy na podstawie szablonu ogólnego oraz szablonu dla pierwszego statusu.
- Zmiana statusu; po przejściu do kolejnego statusu zostają wygenerowane wpisy na podstawie szablonu dla bieżącego statusu; mogą również zostać wygenerowane/usunięte wpisy jeśli zmieni sią skład grupy interesariuszy.
- Zmiana uprawnień dla bieżącego dokumentu na formularzu (ikona kluczyka, wymagane uprawnienie do administrowania uprawnieniami).
- Wykonanie akcji uprawnień do rekordów.
- Wykonanie akcji odświeżenia uprawnień; ta akcja odświeża uprawnienia odpowiednio do zmian składu grupy interesariuszy.
- Wykonanie akcji systemowej Usuń rekord powoduje usunięcie wszystkich wpisów uprawnień dla danego dokumentu.

Nazwa kolumny	Typ danych	Zezwól NULL	Opis
Id	int	☐	Kolumna klucza głównego, identyfikator rekordu w tabeli
ACLId	uniqueidentifier	☐	Identyfikator dokumentu na potrzeby uprawnień (kolumna o tej samej nazwie z odpowiedniej tabeli modelu danych)
CanRead	bit	☐	Uprawnienie do odczytu
CanUpdate	bit	☐	Uprawnienie do aktualizacji; egzekwowane podczas próby zapisu zmienionych danych
CanDelete	bit	☐	Uprawnienie usunięcia dokumentu; dotyczy usunięcia w trybie inline i akcji *Usuń rekord*
CanManage	bit	☐	Uprawnienie do administrowania uprawnieniami
UserId	uniqueidentifier	☑	Identyfikator uprawnionego użytkownika
StatusId	int	☑	Identyfikator statusu, dla którego obowiązuje szablon uprawnień
OUInstanceId	int	☑	Identyfikator jednostki organizacyjnej, wskazuje na szablon interesariuszy z dodaną rolą w jednostce
OUSubinstances	bit	☐	Flaga wskazująca, czy uprawnienia mają obowiązywać w jednostkach podrzędnych danej jednostki organizacyjnej; wartość domyślna *0*; patrz przypis pod tabelą
StakeholderId	int	☑	Identyfikator grupy interesariuszy (*core.Stakeholders*), dla której obowiązuje zestaw uprawnień

Przypis: W nAxiom 1.11.2.0 wycofano możliwość definiowania szablonów dla ról we wskazanych jednostkach i podjednostkach organizacyjnych, jednak istniejące szablony są nadal obsługiwane. Z tego względu w tabeli pozostawiono kolumnę *OUSubinstances*.

TABELA *CORE.ACLINSTANCEENTRYROLE*

Zawiera informacje o rolach powiązanych z danym wpisem uprawnień z tabeli *core.ACLInstanceEntries*.

Nazwa kolumny	Typ danych	Zezwól NULL	Opis
ACLInstanceEntryId	int	☐	Kolumna klucza głównego, identyfikator rekordu z tabeli *core.ACLInstanceEntries*
RoleId	int	☐	Kolumna klucza głównego, identyfikator roli, której dotyczy dany wpis uprawnień; podczas sprawdzania uprawnień system szuka identyfikatora bieżącego użytkownika w tabeli *core.UserProfileRoles*

TABELA *CORE.ACLSTAKEHOLDERENTRIES*

Zawiera informacje o członkach grup interesariuszy.

Nazwa kolumny	Typ danych	Zezwól NULL	Opis
Id	int	☐	Kolumna klucza głównego, identyfikator rekordu w tabeli
ACLId	uniqueidentifier	☐	Identyfikator dokumentu, który ma przypisaną grupę interesariuszy (wartość z kolumny *ACLId* w tabeli dokumentu biznesowego)
UserProfileId	uniqueidentifier	☑	Identyfikator użytkownika należącego do grupy interesariuszy (z tabeli *core.UserProfiles*); wpisy uprawnień dla indywidualnych
StakeholderId	int	☑	Identyfikator grupy interesariuszy
OUInstanceId	int	☑	Identyfikator jednostki organizacyjnej
RoleId	int	☑	Identyfikator roli; jeśli dla danego dokumentu jest podana tylko rola, podczas sprawdzania uprawnień system szuka identyfikatora bieżącego użytkownika w tabeli *core.UserProfileRoles*; jeśli jest podana rola i jednostka, system szuka w tabeli *core.OUInstanceUserRoles*

TABELA *CORE.BUSINESSDOCUMENTROLESALLOWEDTOCREATEINSTANCE*

W tej tabeli przechowywane są informacje o rolach uprawnionych do tworzenia instancji w tabeli wskazanej definicji dokumentu. Na podstawie danych z tej tabeli sprawdzane są uprawnienia do tworzenia dokumentów.

Nazwa kolumny	Typ danych	Zezwól NULL	Opis
Id	int	☐	Kolumna klucza głównego, identyfikator rekordu w tabeli
BusinessDocumentId	int	☐	Identyfikator definicji dokumentu, której dotyczy uprawnienie *create*
RoleAllowedToCreateInstancesId	int	☐	Identyfikator roli, której posiadanie pozwala na tworzenie nowych dokumentów
RowGuid	uniqueidentifier	☐	Unikatowy identyfikator szablonu

WSKAZÓWKA

Brak wpisu dla określonej definicji dokumentu oznacza, że każdy użytkownik może tworzyć nowe instancje dokumentów w tej definicji.

Powiązane tematy:

- Uprawnienia ACL do dokumentów
- Włączanie obsługi uprawnień
- Włączanie weryfikacji uprawnień
- Nowy szablon uprawnień do dokumentu
- Administrowanie uprawnieniami ACL

9.16. STRUKTURA ORGANIZACYJNA

Podsumowanie:

Struktura organizacyjna to wykaz jednostek organizacyjnych oraz powiązań typu nadrzędny-podrzędny między tymi jednostkami. W nAxiom można zdefiniować kilka niezależnych struktur organizacyjnych. Każda struktura ma jedną jednostkę główną i dowolną liczbę jednostek podrzędnych. Każda jednostka podrzędna ma jedną jednostkę nadrzędną i dowolną liczbę jednostek podrzędnych.

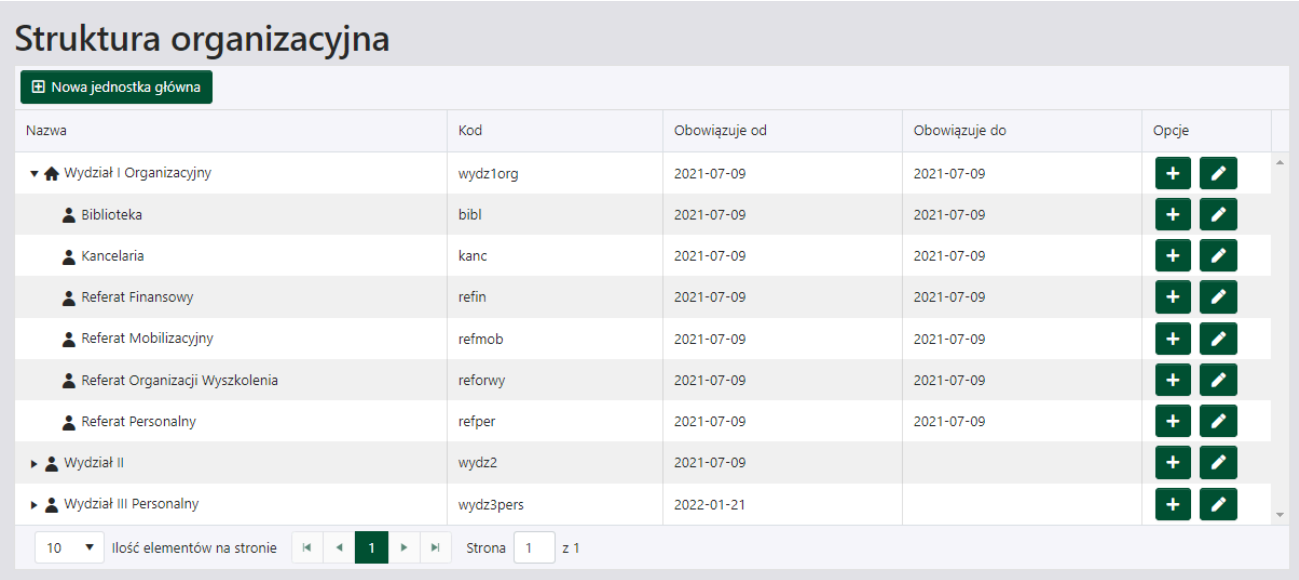
W każdej większej organizacji można wydzielić jednostki — komórki organizacyjne, np. działy, wyodrębnione części, referaty itp. — oraz powiązania między tymi jednostkami (przepływy informacji, podział obowiązków, przynależność) które tworzą *strukturę organizacyjną*. W uproszczeniu można powiedzieć, że struktura organizacyjna określa obszary kompetencji w ramach organizacji oraz przynależność użytkowników.

Dane dotyczące struktury organizacyjnej są potrzebne między innymi do tworzenia hierarchii użytkowników, do adresowania (*dekretowania*) dokumentów, a także do określania uprawnień do funkcjonalności aplikacji oraz danych.

Struktura organizacyjna jest elementem dynamicznym, może się zmieniać na skutek reorganizacji. W nAxiom zarządzanie strukturą organizacyjną jest możliwe w aplikacji *FrontSPA* jako menu dostępne dla użytkownika z określoną rolą.

W ramach standardowej obsługi struktury organizacyjnej w nAxiom można tworzyć jednostki organizacyjne, podając ich dane podstawowe, takie jak nazwa, kod, data początku i końca obowiązywania oraz status. Na potrzeby synchronizacji jednostek organizacyjnych z serwerem usług katalogowych można także określić identyfikator GUID jednostki. Te dane są zapisywane w tabeli systemowej *core.OUInstances*. Poza tym dostępna jest funkcjonalność dodawania użytkowników do jednostek organizacyjnych. Informacje o przynależności użytkowników do jednostek organizacyjnych są zapisywane w tabeli

Funkcjonalności standardowe można rozszerzyć o zapisywanie danych dodatkowych jednostek organizacyjnych (kontaktowych, adresowych itp.). W tym celu należy utworzyć odpowiednią tabelę (model standardowy) w kreatorze modeli danych i wygenerować z niego *elementy aplikacji*. Następnie nalezy podać w ustawieniach nAxiom identyfikator definicji dokumentu utworzonej dla danych dodatkowych struktury organizacyjnej oraz identyfikator formularza.



Przykładowa struktura organizacyjna

Powiązane tematy:

- Ustawienia systemu, Struktura organizacyjna

9.16.1. Dane dodatkowe jednostek struktury organizacyjnej

Podsumowanie:
Jedyne dane, jakie nAxiom pozwala systemowo zapisać dla jednostki organizacyjnej to `Nazwa`, `Kod`, `Obowiązuje od` i `Obowiązuje do`. Do zapisywania innych danych, np. adresu, numerów pomieszczeń biurowych, rodzaju jednostki, opisu itp. należy zbudować osobną tabelę

Aby móc zapisywać dodatkowe dane dla jednostek struktury organizacyjnej, wykonaj następujące czynności:

1. W kreatorze modeli danych zdefiniuj tabelę (standardowy model danych) z odpowiednimi kolumnami do zapisu danych dodatkowych.

   UWAGA
   Wszystkie kolumny użytkownika w tej tabeli dla jednostek organizacyjnych muszą mieć puste pole wyboru `Wymagane`.

2. Wygeneruj elementy aplikacji na podstawie utworzonej tabeli (`Generuj elementy aplikacji`).
3. Przejdź do *adminSPA* > `Ustawienia systemu` > `Struktura organizacyjna` i wypełnij pola `ID dokumentu biznesowego` (z tabeli *[core].[BusinessDocuments]*) i `ID definicji formularza` (z listy `INTERFEJS UŻYTKOWNIKA` > `Formularze`).

Po wykonaniu tych czynności w oknie edycji jednostki organizacyjnej pojawi się nowa zakładka `Dane dodatkowe`.

Powiązane tematy:

- Kreator modeli danych

9.17. DIAGNOSTYKA W NAXIOM

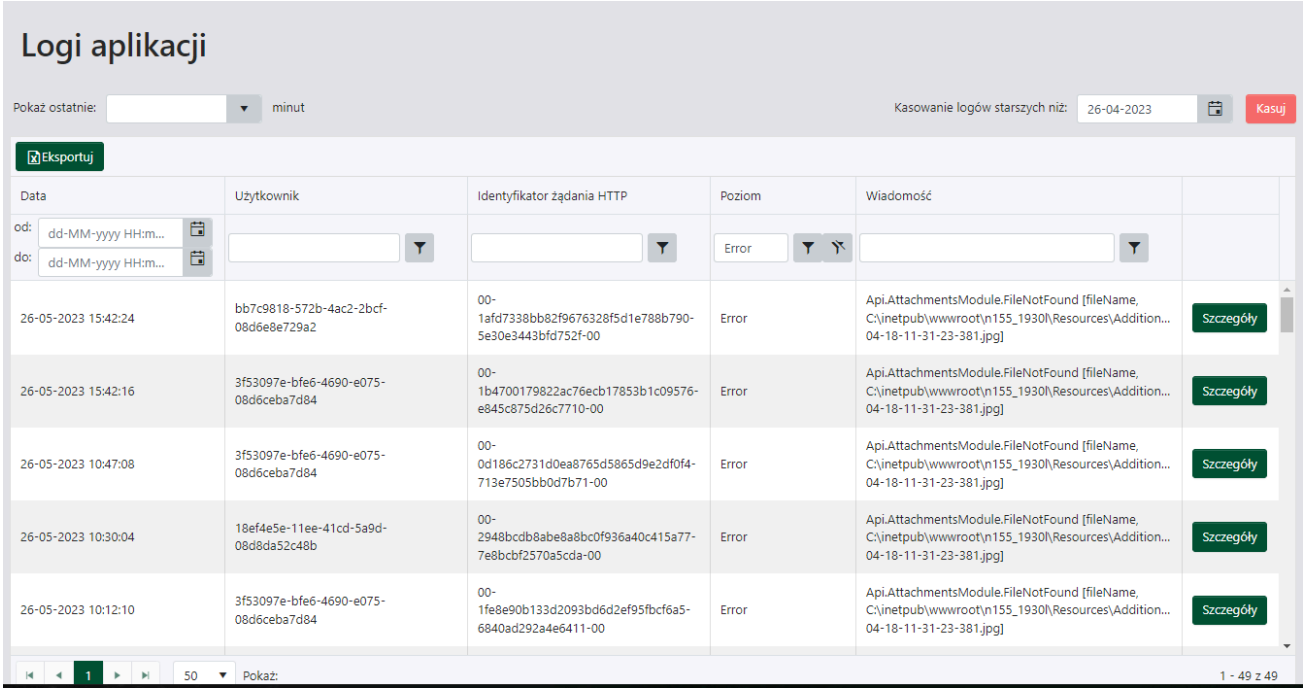
Podsumowanie:
nAxiom oferuje szereg narzędzi i danych do rejestrowania informacji generowanych przez oprogramowanie w trakcie tworzenia i użytkowania aplikacji biznesowych. Te informacje mają zastosowanie do rozwiązywania problemów występujących podczas użytkowania platformy, w tym także zgłaszanych do działu wsparcia nAxiom.

Platforma nAxiom jest zbudowana w architekturze mikroserwisów i obejmuje kilkanaście modułów, które odpowiadają za różne aspekty działania. Część z tych serwisów jest opartych na technologii *.NET* i odpowiada za obsługę *server side* (*api*, *auth*, *tenantsapi* i inne), a część wykorzystuje platformę *Angular* i obsługuje komponenty nAxiom *client side* (*admin*, *front*, *workflow* i *tenatsadmin*).

Do logowania komunikatów z serwisów dotnetowych wykorzystywane jest oprogramowanie NLog. Logi komunikatów mogą być zapisywane w plikach *log.db3* (SQLite), plikach tekstowych lub w bazie danych. Poniżej podano położenie i nazwę pliku logu dla poszczególnych serwisów nAxiom:

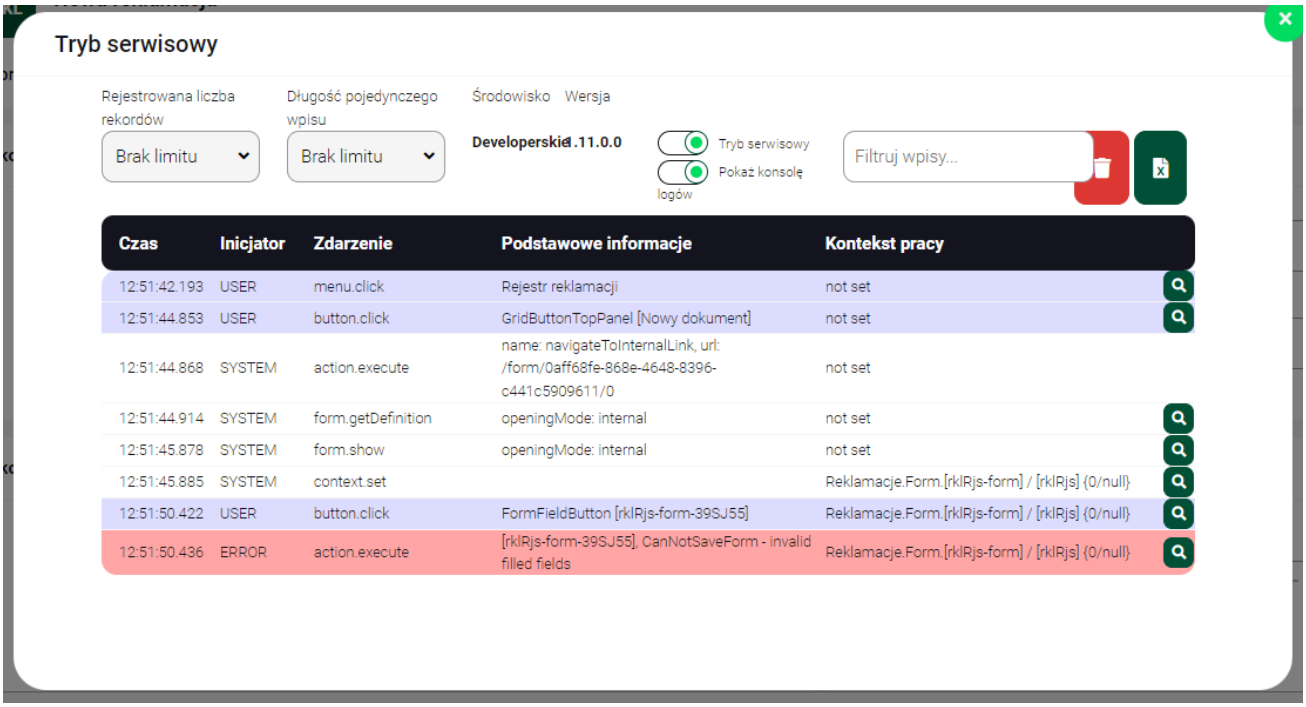
- *api*: *{folder-instalacyjny-naxiom}\api\Log.db3*; log serwisu *api* jest także dostępny w aplikacji *AdminSPA* (`Administracja` > `Logi środowiska i bieżącego tenanta`).
- *auth*: *{folder-instalacyjny-naxiom}\auth\Log.db3*
- *bpmnapi*: *{folder-instalacyjny-naxiom}\bpmnapi\Log.db3*
- *bpmnintegratorapi*: *{folder-instalacyjny-naxiom}\bpmnintegratorapi\Log.db3*
- *publicapi*: *{folder-instalacyjny-naxiom}\publicapi\Log.db3*
- *reportsapi*: *{folder-instalacyjny-naxiom}\reportsapi\logs\internal-nlog.txt*
- *taskservice*: *{folder-instalacyjny-naxiom}\taskservice\Log.db3*
- *tenantsapi*: *{folder-instalacyjny-naxiom}\tenantsapi\logs{data}_logs-all.log*

W przypadku wdrożenia z obrazów Docker logi są kierowane na konsolę w celu umożliwienia przechwycenia ich przez oprogramowanie do agregowania logów. Konfiguracja logowania jest definiowana w plikach *nlog.config* dla poszczególnych serwisów.



Logi aplikacji

Dla konsultantów duże znaczenie diagnostyczne mają informacje o pracy użytkowników końcowych w aplikacji *FrontSPA*. Te informacje są kierowane do *logów serwisowych* i wyświetlane w konsoli, skąd można je wyeksportować do pliku *xls*. Włączenie tych logów wymaga posiadania odpowiednich uprawnień PBA.



Log serwisowy

Trzeci rodzaj logów dostępnych w nAxiom to *logi bezpieczeństwa*. Są to dane dotyczące predefiniowanych zdarzeń generowanych podczas korzystania z witryny. Te dane są domyślnie zapisywane w bazie danych w tabeli *core.SecurityAudits*.

	Id	Timestamp	EventType	Message	UserName	EventCategory	Target	Ip	UserAgent
1	1	2023-04-27 12:31:07.57	EntityModified	Obiekt typu 'AttachmentRepository' Kod: default, Id...		ApplicationModel	Kod: default, Id: 1, RowGuid: eb9c6e20-6456-4f17...	NULL	
2	2	2023-04-27 12:31:07.72	EntityModified	Obiekt typu 'AttachmentRepository' Kod: default-e...		ApplicationModel	Kod: default-email, Id: 2, RowGuid: cbf35aa6-da95...	NULL	
3	3	2023-04-27 12:31:07.73	EntityModified	Obiekt typu 'AttachmentRepository' Kod: default-sy...		ApplicationModel	Kod: default-system, Id: 3, RowGuid: 0a001fe8-fa4...	NULL	
4	4	2023-04-27 12:32:08.00	UserLoggedInSuccessfully	Użytkownik 'consultant' został poprawnie zalogow...	consultant	Authorization	consultant	::1	Mozilla/5.0 (Wi
5	5	2023-04-27 12:47:44.20	UserLoggedInSuccessfully	Użytkownik 'consultant' został poprawnie zalogow...	consultant	Authorization	consultant	::1	Mozilla/5.0 (Wi
6	6	2023-04-27 12:54:14.92	UserLoggedInSuccessfully	Użytkownik 'consultant' został poprawnie zalogow...	consultant	Authorization	consultant	::1	Mozilla/5.0 (Wi
7	7	2023-04-27 12:57:31.62	UserLoggedInSuccessfully	Użytkownik 'consultant' został poprawnie zalogow...	consultant	Authorization	consultant	::1	Mozilla/5.0 (Wi
8	8	2023-04-27 12:59:40.27	UserLoggedInSuccessfully	Użytkownik 'consultant' został poprawnie zalogow...	consultant	Authorization	consultant	::1	Mozilla/5.0 (Wi
9	9	2023-04-27 13:01:48.94	UserLoggedInSuccessfully	Użytkownik 'consultant' został poprawnie zalogow...	consultant	Authorization	consultant	::1	Mozilla/5.0 (Wi
10	10	2023-04-27 14:12:37.76	EntityAdded	Obiekt typu 'ReportDefinition' Kod: test1, Id: 0, Ro...	consultant	ApplicationModel	Kod: test1, Id: 0, RowGuid: b2e9fc47-9064-496f-9...	::1	Mozilla/5.0 (Wi
11	11	2023-04-27 15:38:26.79	UserLoggedInSuccessfully	Użytkownik 'consultant' został poprawnie zalogow...	consultant	Authorization	consultant	::1	Mozilla/5.0 (Wi
12	12	2023-04-27 16:02:55.46	UserLoggedInSuccessfully	Użytkownik 'admin' został poprawnie zalogowany	admin	Authorization	admin	::1	Mozilla/5.0 (Wi
13	13	2023-04-27 16:03:16.40	UserPermisionsModified	Uprawnienia użytkownika admin zostały zmienione.	admin	Permissions	admin	::1	Mozilla/5.0 (Wi
14	14	2023-04-27 16:03:23.28	UserLoggedInSuccessfully	Użytkownik 'admin' został poprawnie zalogowany	admin	Authorization	admin	::1	Mozilla/5.0 (Wi
15	15	2023-04-27 16:04:51.34	UserProfileUpdated	Profil użytkownika admin został zmodyfikowany.	admin	User	admin	::1	
16	16	2023-04-28 12:02:04.61	UserLoggedInSuccessfully	Użytkownik 'consultant' został poprawnie zalogow...	consultant	Authorization	consultant	::1	Mozilla/5.0 (Wi
17	17	2023-05-08 10:26:57.49	UserLoggedInSuccessfully	Użytkownik 'admin' został poprawnie zalogowany	admin	Authorization	admin	::1	Mozilla/5.0 (Wi
18	18	2023-05-08 10:28:12.37	UserProfileUpdated	Profil użytkownika consultant został zmodyfikowany.	admin	User	consultant	::1	
19	19	2023-05-08 10:30:40.47	UserPermisionsModified	Uprawnienia użytkownika consultant zostały zmien...	admin	Permissions	consultant	::1	Mozilla/5.0 (Wi
20	20	2023-05-08 10:45:05.49	UserLoggedInSuccessfully	Użytkownik 'consultant' został poprawnie zalogow...	consultant	Authorization	consultant	::1	Mozilla/5.0 (Wi
21	21	2023-05-08 12:22:26.34	UserLoggedInSuccessfully	Użytkownik 'consultant' został poprawnie zalogow...	consultant	Authorization	consultant	::1	Mozilla/5.0 (Wi
22	22	2023-05-10 09:09:41.35	UserLoggedInSuccessfully	Użytkownik 'consultant' został poprawnie zalogow...	consultant	Authorization	consultant	::1	Mozilla/5.0 (Wi
23	23	2023-05-10 10:32:42.11	UserLoggedInSuccessfully	Użytkownik 'consultant' został poprawnie zalogow...	consultant	Authorization	consultant	::1	Mozilla/5.0 (Wi
24	24	2023-05-10 10:33:15.77	UserLoggedInSuccessfully	Użytkownik 'consultant' został poprawnie zalogow...	consultant	Authorization	consultant	::1	Mozilla/5.0 (Wi
25	25	2023-05-10 11:18:46.91	UserLoggedInSuccessfully	Użytkownik 'consultant' został poprawnie zalogow...	consultant	Authorization	consultant	::1	Mozilla/5.0 (Wi
26	26	2023-05-10 11:28:37.35	UserLoggedInSuccessfully	Użytkownik 'admin' został poprawnie zalogowany	admin	Authorization	admin	::1	Mozilla/5.0 (Wi
27	27	2023-05-10 11:31:10.76	UserProfileCreated	Profil użytkownika tako został utworzony.	admin	User	tako	::1	
28	28	2023-05-10 11:35:09.10	UserLoggedInSuccessfully	Użytkownik 'consultant' został poprawnie zalogow...	consultant	Authorization	consultant	::1	Mozilla/5.0 (Wi
29	29	2023-05-10 12:31:58.08	EntityAdded	Obiekt typu 'ApplicationDefinition' Kod: *, Id: 0, Ro...	consultant	ApplicationModel	Kod: *, Id: 0, RowGuid: 5f53e675-8ea6-4d48-b44a...	::1	Mozilla/5.0 (Wi
30	30	2023-05-10 13:48:50.69	UserLoggedInSuccessfully	Użytkownik 'consultant' został poprawnie zalogow...	consultant	Authorization	consultant	::1	Mozilla/5.0 (Wi

Tabela core.SecurityAudits

W nAxiom jest także dostępny *MiniProfiler for .NET* — narzędzie do profilowania żądań HTTP wysyłanych podczas korzystania z aplikacji pod kątem wydajności. Jego krótki opis zamieszczono w jednej z kolejnych sekcji.

Powiązane tematy:

- Logi bezpieczeństwa
- Logi środowiska i bieżącego tenanta
- Konfiguracja logowania serwisów nAxiom
- Logi serwisowe

9.17.1. Logi bezpieczeństwa

Podsumowanie:
Logi bezpieczeństwa zawierają dane dotyczące czynności wykonywanych przez użytkowników witryny, które mają związek z bezpieczeństwem (logowanie, zmiany w profilu użytkownika, modyfikowanie obiektów aplikacji).

Dane do logów bezpieczeństwa są domyślnie zapisywane w bazie danych w tabeli *core.SecurityAudits*. Alternatywnie mogą one być zapisywane w pliku *SecurityAudit.db3* (SQLite) indywidualnie dla każdego serwisu. Odpowiada za to ustawienie w pliku *appsettings.json* każdego serwisu:

```
"SecurityAudit": {
  "UseSqlLite": false
}
```

W tabeli rejestrowane są następujące typy zdarzeń w podziale na kategorie (kolumny *EventCategory* i *EventType*)

- *Authorization*:
  - *AccountLockedOutLoginAttempt*: próba logowania z zablokowanego konta;
  - *ActiveDirectoryUserLoggedInSuccessfully*: pomyślne zalogowanie użytkownika LDAP;
  - *ActiveDirectoryUserLoginAttemptFailed*: niepowodzenie logowania użytkownika LDAP;
  - *ExpiredAccountLoginAttempt*: próba logowania na konto, które wygasło;
  - *ExpiredAccountSimpleLoginAttempt*: próba uproszczonego logowania na konto, które wygasło;
  - *ExternalLoginAttemptFailed*: niepowodzenie logowania zewnętrznego;
  - *ExternalLoginSuccessfull*: powodzenie logowania zewnętrznego; dla tego zdarzenia, w kolumnie *Details* zapisywana jest informacja o poziomie uwierzytelniania użytkownika w formacie *UserAuthenticationLevel: wartość* (o ile konfiguracja logowania SSO obejmuje przekazywanie tej informacji);
  - *InactiveAccountLoginAttempt*: próba logowania na nieaktywne konto;
  - *InactiveAccountSimpleLoginAttempt*: próba uproszczonego logowania na nieaktywne konto;
  - *InvalidCredentialsLoginAttempt*: próba logowania z nieprawidłowymi poświadczeniami;
  - *PasswordReseted*: zresetowanie hasła;
  - *PasswordResetFailed*: niepowodzenie w resetowaniu hasła;
  - *PasswordResetLinkExpired*: upłynięcie terminu ważności linku do resetowania hasła;
  - *PasswordResetLinkSent*: wysłanie linku do resetowania hasła;
  - *SetPasswordLinkSent*: wysłanie linku do ustawienia hasła;
  - *SimpleLoginSuccessfull*: powodzenie uproszczonego logowania;
  - *UserLoggedInSuccessfully*: pomyślne zalogowanie użytkownika;
  - *UserLoggedOutSuccessfully*: pomyślne wylogowanie użytkownika;
- *User*:
  - *PasswordChanged*: zmiana hasła;
  - *UserProfileCreated*: utworzenie profilu użytkownika;
  - *UserProfileDeleted* usunięcie profilu użytkownika;

    - *UserProfileUpdated* aktualizacja profilu użytkownika;
- *Dictionaries*:
    - *DictionaryCreated*: utworzenie słownika;
    - *DictionaryRemoved*: usunięcie słownika;
    - *DictionaryUpdated*: aktualizacja słownika;
    - *DictionaryValueCreated*: utworzenie pozycji słownika;
    - *DictionaryValueRemoved*: usunięcie pozycji słownika;
    - *DictionaryValueUpdated*: aktualizacja pozycji słownika;
    - *HierarchicalDictionaryCreated*: utworzenie słownika hierarchicznego;
    - *HierarchicalDictionaryUpdated*: aktualizacja słownika hierarchicznego;
    - *HierarchicalDictionaryValueCreated*: utworzenie pozycji słownika hierarchicznego;
    - *HierarchicalDictionaryValueUpdated*: aktualizacja pozycji słownika hierarchicznego;
- *Permissions*:
    - *NoApplicationPermisson*: brak uprawnień do aplikacji;
    - *NoPermisson*: brak uprawnień;
    - *RolePermisionsModified*: zmiana uprawnień roli;
    - *UserPermisionsModified*: zmiana uprawnień użytkownika;
- *ApplicationModel*:
    - *EntityAdded*: dodanie obiektu;
    - *EntityDeleted*: usunięcie obiektu;
    - *EntityModified*: modyfikacja obiektu;
- *BusinessDocument*:
    - *BusinessDocumentCreated*: utworzenie definicji dokumentu;
    - *BusinessDocumentDeleted*: usunięcie definicji dokumentu;
    - *BusinessDocumentModified*: zmiana definicji dokumentu;
    - *BusinessDocumentTransitionsSchemaModified*: zmiana diagramu procesu;
- *FormDefinition*:
    - *ACLFormTemplateEntryModified*: zmiana uprawnień AC formularza;
    - *FormDefinitionCopied*: skopiowanie definicji formularza;
    - *FormDefinitionCreated*: utworzenie definicji formularza;
    - *FormDefinitionDeleted*: usunięcie definicji formularza;
    - *FormDefinitionModified* modyfikacja definicji formularza;
- *GridDefinition*:
    - *GridDefinitionCopied*: skopiowanie definicji widoku listy;
    - *GridDefinitionCreated*: utworzenie definicji widoku listy;

    - *GridDefinitionDeleted*: usunięcie definicji widoku listy;
    - *GridDefinitionModified*: modyfikacja definicji widoku listy;
- *DataModel*:
    - *GeneratingFlow*: wygenerowanie elementów aplikacji.

Strukturę tabeli *SecurityAudits* przedstawiono poniżej.

Nazwa kolumny	Typ danych	Zezwól NULL	Opis
Id	*bigint*	☐	Kolumna klucza głównego, identyfikator rekordu w tabeli
Timestamp	*datetime2(2)*	☐	Sygnatura czasowa wpisu
EventType	*nvarchar(64)*	☑	Typ rejestrowanego zdarzenia
Message	*nvarchar(512)*	☑	Opis zdarzenia
UserName	*nvarchar(128)*	☑	Nazwa użytkownika wykonującego operację
EventCategory	*nvarchar(64)*	☑	Kategoria zdarzenia zdarzenia
Target	*nvarchar(128)*	☑	Dane identyfikujące obiekt, którego dotyczyło zdarzenie, np. kod, identyfikator, *RowGuid* itp.
Ip	*nvarchar(64)*	☑	Adres IP, z którego wysłano żądanie
UserAgent	*nvarchar(256)*	☑	Informacje o aplikacji klienckiej, z której wysłano żądanie
Details	*nvarchar(MAX)*	☑	Dodatkowe, szczegółowe informacje o zdarzeniu, jeśli są dostępne
EntityType	*nvarchar(128)*	☑	Typ obiektu z definicji aplikacji, którego dotyczyło zdarzenie
RequestId	*nvarchar(64)*	☑	GUID żądania HTTP, które zainicjowało jedno lub więcej zdarzeń
Description	*nvarchar(MAX)*	☑	Informacje o identyfikatorze roli i/lub jednostki organizacyjnej dla operacji dodawania i usuwania ról (w tym ról w jednostkach) i jednostek organizacyjnych

Powiązane tematy:

- Logi środowiska i bieżącego tenanta
- Logi serwisowe

9.17.2. Konfiguracja logowania serwisów nAxiom

Podsumowanie:
Logowanie informacji z serwisów nAxiom obsługujących *server side*, to jest opartych na platformie .NET, odbywa się przy użyciu oprogramowania NLog. Za konfigurację logowania odpowiadają pliki *nlog.config* w podfolderach poszczególnych serwisów w folderze instalacyjnym witryny. Informacje są logowane w pliku bazy danych SQLite (*log.db3*) i lub plikach textowych w podfolderze *logs* dla poszczególnych serwisów.

INFO
W przypadku aktualizacji witryny nAxiom, instalator archiwizuje znalezione pliki *db3*, zmieniając im nazwę na *Log-<znacznikczasu>.db3_arch*.

W plikach logów w formacie SQLite (*db3*) logowane są następujące informacje:

- *MachineName*: nazwa komputera
- *Timestamp*: znacznik czasu zdarzenia
- *Level*: poziom istotności zdarzenia; patrz niżej
- *TenantIdentity*: identyfikator tenanta
- *UserIdentity*: id zalogowanego użytkownika (jeśli dostępne)
- *HttpRequestTraceIdentifier*: unikalny identyfikator żądania HTTP pozwalający łączyć wiele informacji zalogowanych w obrębie jednego żądania obsługiwanego przez system.
- *ControllerName*: nazwa kontrolera
- *ActionName*: nazwa akcji
- *Logger*: nazwa modułu rejestrującego dany wpis
- *Message*: treść komunikatu
- *Exception*: zgłoszony wyjątek
- *CustomParamModel*: wartości parametrów używanych w zapytaniu SQL wykonywanym w ramach akcji (JSON); zapisywane tylko dla poziomu istotności *Error*
- *QueryModel*: jak wyżej
- *RequestName*: nazwa żądania
- *Request*: treść żądania
- *MigrationLogId*: identyfikator logu migracji

Pliki *nlog.config* to pliki XML z konfiguracją logowania każdego serwisu *dotnetowego*. Plik musi zawierać dwie sekcje:

- *targets*: miejsca docelowe logowania informacji; np. plik tekstowy, plik bazy danych, konsola, e-mail.
- *rules*: reguły logowania informacji; zawierają elementy *logger*, które wiążą miejsca docelowe z poziomem istotności.

Elementy *logger* mają następujące atrybuty:

- *name*: nazwa wzorca logowania;
- *minlevel*: minimalny poziom istotności;
- *maxlevel*: maksymalny poziom istotności;
- *level*: konkretny poziom istotności;
- *levels*: rozdzielana przecinkami lista poziomów istotności;
- *writeTo*: rozdzielana przecinkami lista elementów *target*, w których mają być zapisywane informacje;
- *final*: po tej regule nie są przetwarzane żadne inne reguły;
- *enabled*: ustawienie wartości *false* wyłącza regułę z przetwarzania.

Przykładowa konfiguracja pliku *nlog.config* (dla serwisu *api*):

```
<?xml version="1.0" encoding="utf-8"?>
<nlog xmlns="http://www.nlog-project.org/schemas/NLog.xsd"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      autoReload="true"
      internalLogLevel="Error"
      internalLogFile="${basedir}/logs/internal-nlog.txt">

        <extensions>
                <add assembly="NLog.Web.AspNetCore"/>
        </extensions>

        <!--Błędy uniemożliwiające działanie aplikacji-->
        <variable name='FatalLevel' value='Fatal'/>
        <!--Błędy, które mogą, ale nie muszą, zatrzymać aplikacji-->
        <variable name='ErrorLevel' value='Error'/>
        <!--Nieoczekiwane zdarzenia, które nie zatrzymują aplikacji-->
        <variable name='WarnLevel' value='Warn'/>
        <!--Normalne zdarzenia podczas działania aplikacji-->
        <variable name='InfoLevel' value='Info'/>
        <!--Informacje do debugowania-->
```

```
      <variable name='DebugLevel' value='Debug'/>
      <!--Szczegółowe informacje do debugowania-->
      <variable name='TraceLevel' value='Trace'/>
<!--Wyłączenie logowania-->
      <variable name='OffLevel' value='Off'/>

      <targets>
<!--Plik tekstowy; atrybut layout definiuje rejestrowane ingformacje-->
              <target xsi:type="File"
                              name="logfile"
                              maxArchiveFiles="7"
                              archiveAboveSize="10485760"
                              fileName="${basedir}/logs/${shortdate}_logs-all.log"
                              layout="${longdate}|${event-properties:item=EventId_Id}|${uppercase:${level}
              ${logger}|${message} ${exception:format=tostring}" />

<!--Plik bazy danych; atrybut commandText definiuje zapytanie tworzące
tabelę z rejestrowanymi informacjami-->
              <target name="sqlitedb" xsi:type="Database" keepConnection="false"
                              dbProvider="System.Data.SQLite.SQLiteConnection, System.Data.SQLite"
                              connectionString="Data Source=${basedir}/Log.db3;"
                              commandText="insert into Logs(MachineName, Timestamp, Level, TenantIdentity,
      UserIdentity, ControllerName, ActionName, Logger, Message, Exception, CustomParamModel,
      QueryModel, Request, RequestName, HttpRequestTraceIdentifier, MigrationLogId)
      values(@MachineName, @Timestamp, @Level, @TenantIdentity, @UserIdentity,
      @ControllerName, @ActionName, @Logger, @Message, @Exception, @CustomParamModel,
      @QueryModel, @Request, @RequestName, @HttpRequestTraceIdentifier, @MigrationLogId)">
                      <parameter name="@MachineName" layout="${machinename}" />
                      <parameter name="@Timestamp" layout="${date}" />
                      <parameter name="@Level" layout="${level}" />
                      <parameter name="@TenantIdentity" layout="${mdc:item=TenantGuid}" />
                      <parameter name="@UserIdentity" layout="${mdc:item=UserGuid}" />
                      <parameter name="@ControllerName" layout="${aspnet-mvc-controller}" />
                      <parameter name="@ActionName" layout="${aspnet-mvc-action}" />
                      <parameter name="@Logger" layout="${logger}" />
                      <parameter name="@Message" layout="${message}" />
                      <parameter name="@Exception" layout="${exception:tostring}" />
                      <parameter name="@CustomParamModel" layout="${mdlc:CustomParamModel}" />
                      <parameter name="@Request" layout="${mdlc:Request:when=level=LogLevel.Error}" />
                      <parameter name="@RequestName" layout="${mdlc:RequestName}" />
                      <parameter name="@HttpRequestTraceIdentifier" layout="${aspnet-traceidentifier}" />
                      <parameter name="@QueryModel" layout="${mdlc:QueryModel}" />
```

```
                        <parameter name="@MigrationLogId" layout="${mdlc:item=MigrationLogId}" />
                </target>

        </targets>


        <rules>
                <logger name="*" minlevel="${OffLevel}" writeTo="logfile" />

                <!--Pominięcie wpisów z modułów Microsoft i IdentityServer4
    do poziomu Info włącznie-->
                <logger name="Microsoft.*" maxlevel="${InfoLevel}" final="true" />
                <logger name="IdentityServer4.*" maxlevel="${InfoLevel}" final="true" />

                <!-- Zapis wszystkich informacji od poziomu Info do pliku bazy danych-->
                <logger name="*" minlevel="${InfoLevel}" writeTo="sqlitedb" />

        </rules>

        <!-- Uwzględnienie innych plików nlog -->
        <include file="nlog-*.config" />
</nlog>
```

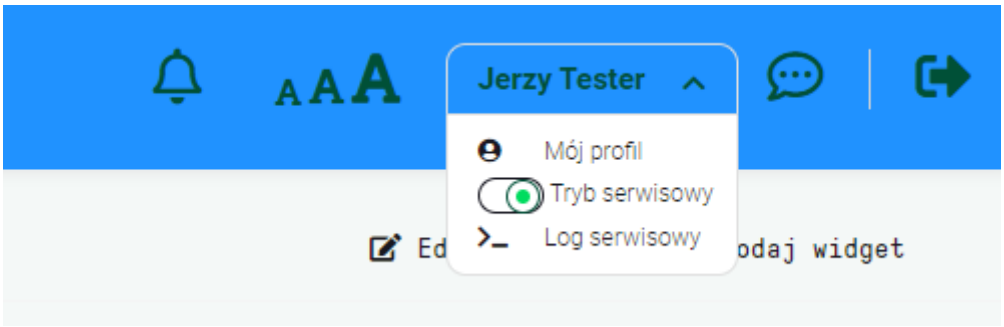
Powiązane tematy:

- Diagnostyka w nAxiom
- Logi bezpieczeństwa
- Logi środowiska i bieżącego tenanta
- Logi serwisowe

9.17.3. Logi serwisowe

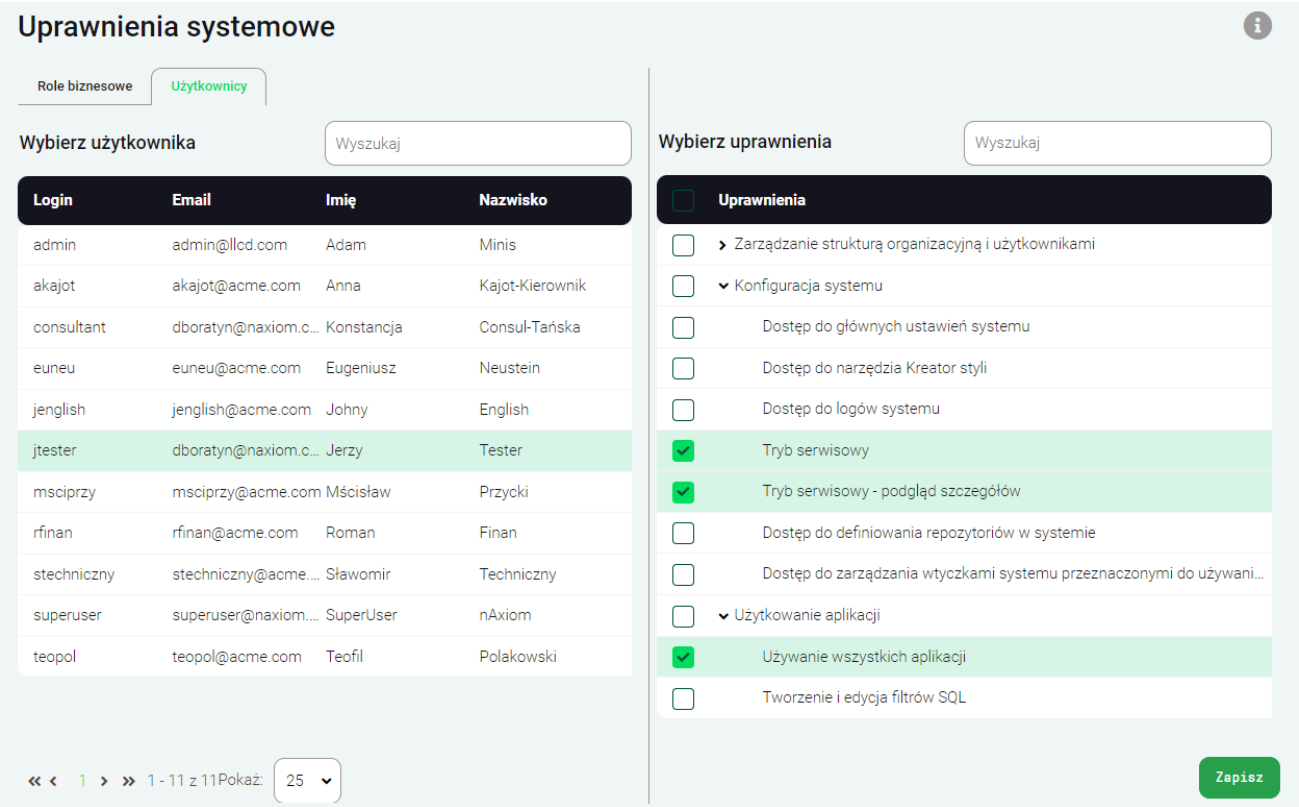
Podsumowanie:
Tryb serwisowy to tryb, w którym logowane są informacje na temat zdarzeń występujących w działaniu serwisu *FrontSPA*. Do włączenia tego trybu użytkownik musi posiadać odpowiednie uprawnienia PBA. Logowane są zdarzenia inicjowane przez użytkownika i przez system oraz błędy. Logowane informacje mogą być wyświetlane na bieżąco w konsoli logów serwisowych lub w oknie dialogowym, skąd można je wyeksportować do pliku Excel.

Użytkownik zalogowany w aplikacji *FrontSPA* może włączyć tryb serwisowy, klikając swoje imię i nazwisko na pasku nagłówka (z prawej strony u góry w oknie przeglądarki), a nastepnie włączając przełącznik `Tryb serwisowy` w wyświetlonym menu.



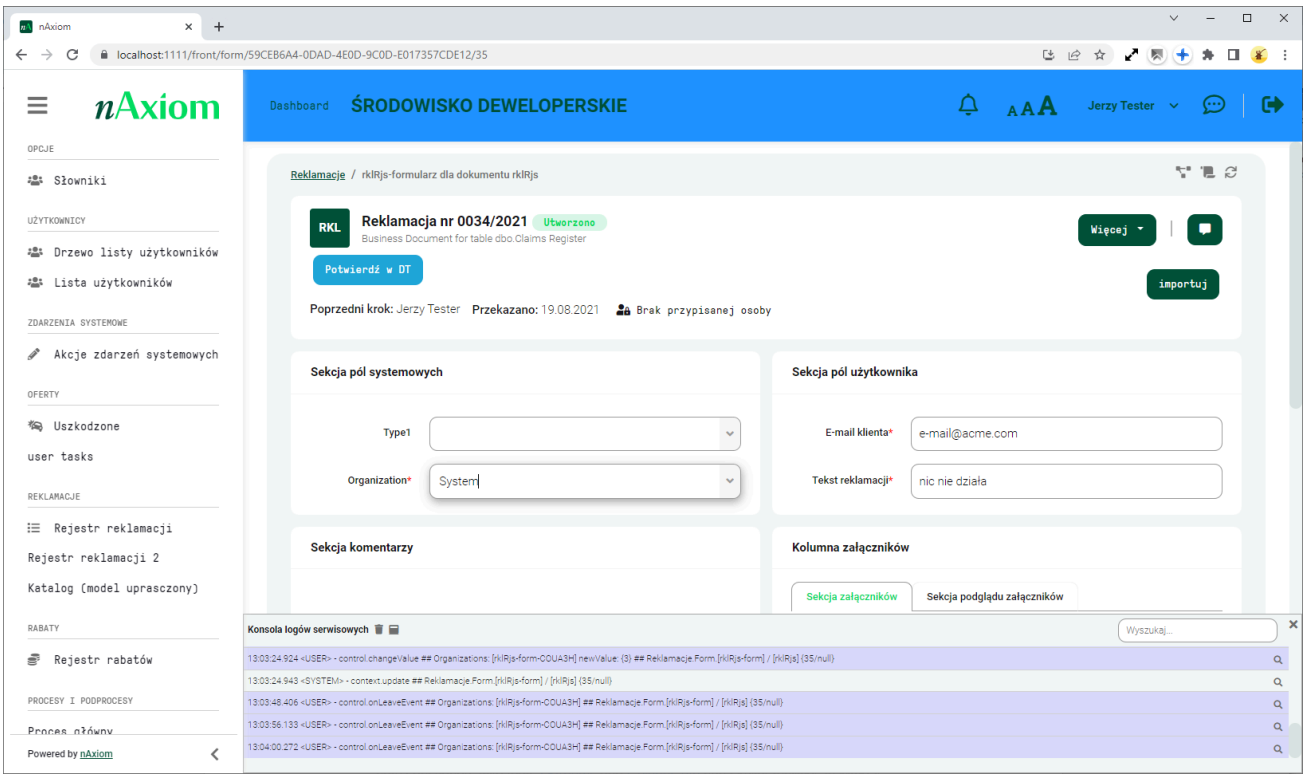
Tryb serwisowy

Ten przełącznik jest dostępny tylko dla użytkowników, którzy mają odpowiednie uprawnienie PBA nadane przez użytkownika z rolą systemową *admin*. Drugie uprawnienie widoczne na ilustracji uprawnia do podglądu szczegółów zalogowanych informacji.



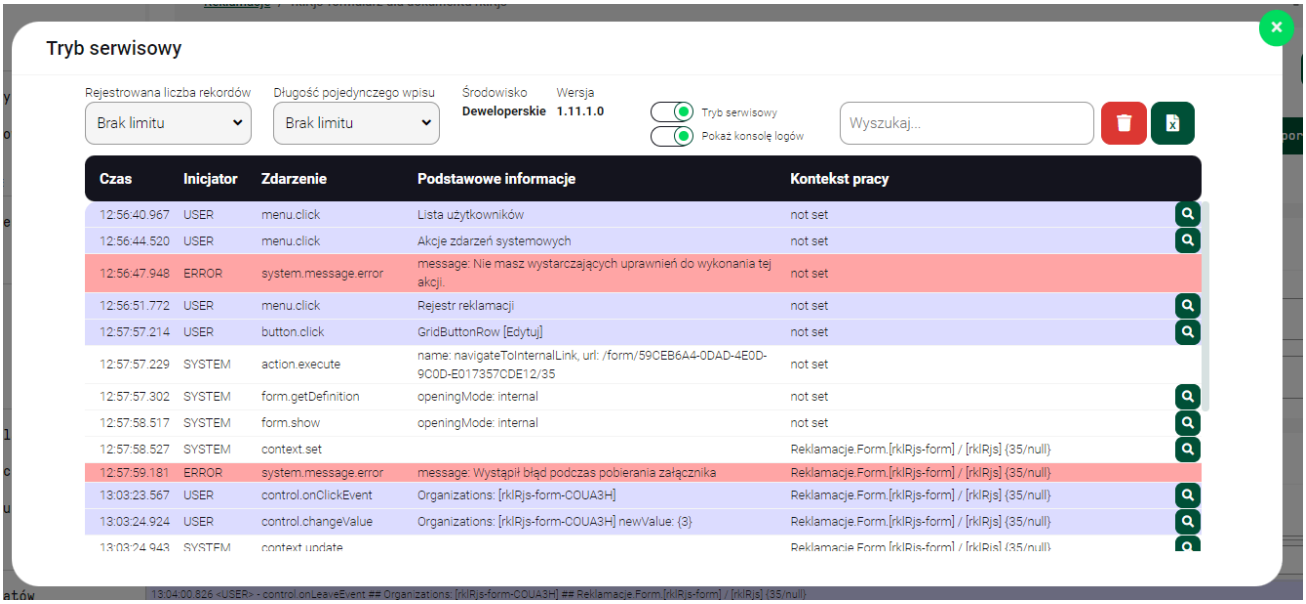
Uprawnienia PBA do trybu serwisowego

Włączenie trybu serwisowego powoduje wyświetlenie konsoli, w której są widoczne informacje o zdarzeniach podczas pracy użytkownika w aplikacji biznesowej.



Konsola logów serwisowych

Logi można wyświetlić w oknie dialogowym, służy do tego polecenie `Log serwisowy` w menu profilu użytkownika. To okno można wyświetlić z konsoli logów, klikając ikonę `Przejdź do okna logów`.



Okno dialogowe Tryb serwisowy

W oknie `Tryb serwisowy` można skonfigurować limity liczby zapisanych rekordów i długości wpisu. Zmiana ustawień wymaga wyłączenia trybu serwisowego. Ponadto w tym oknie można wyłączyć wyświetlanie konsoli logów, wyczyścić, log serwisowy oraz wyeksportować zawartość logu serwisowego do pliku Excel.

Logi serwisowe zawierają następujące informacje:

- *Czas*: godzina występienia zdarzenia; informacje o dacie dla danego logu są zapisywane w pliku eksportu.
- *Inicjator*: strona inicjująca wystąpienie zdarzenia; możliwe wartości to *User*, *System* i *Error*.
- *Zdarzenie*: rodzaj zarejestrowanego zdarzenia; lista zdarzeń znajduje się poniżej.
- *Podstawowe informacje*: informacje dotyczące zdarzenia, np. nazwa kliknięgo menu.
- *Kontekst pracy*: dodatkowe informacje dotyczące zdarzenia, np. informacje o formularzu.

Po kliknięciu ikony z lupą z prawej strony wpisu wyświetlane są szczegóły wpisu w formacie JSON.

Szczegóły logu serwisowego

Lista logowanych zdarzeń jest następująca:

- *attachment.add*: dodanie załącznika
- *attachment.add.version*: dodanie wersji załącznika
- *attachment.delete*: usunięcie załącznika

- *attachment.delete.version*: usunięcie wersji załącznika
- *attachment.download*: pobranie załącznika
- *attachment.download.version*: pobranie wersji załącznika
- *attachment.preview*: wyświetlenie podglądu załącznika
- *form.reload*: przeładowanie (odświeżenie) formularza
- *form.getDefinition*: pobranie definicji formularza
- *form.show*: wyświetlenie formularza
- *form.actionOnOpen*: wykonanie akcji podczas otwierania formularza
- *form.runScript*: uruchomienie skryptu dla zmiany kontekstu formularza
- *form.section.runScript*: uruchomienie skryptu dla sekcji formularza
- *form.close*: zamknięcie formularza
- *control.changeValue*: zmiana wartości w kontrolce
- *control.onChangeEvent*: wyzwolenie zdarzenia *OnChange* dla kontrolki
- *control.onLeaveEvent*: wyzwolenie zdarzenia *OnLeave* dla kontrolki
- *control.onClickEvent*: wyzwolenie zdarzenia *OnClick* dla kontrolki
- *control.onChangeAction*: wykonanie akcji dla zdarzenia *OnChange*
- *control.onLeaveAction*: wykonanie akcji dla zdarzenia *OnLeave*
- *control.onClickAction*: wykonanie akcji dla zdarzenia *OnClick*
- *control.Depedency.Initialize*: zainicjowanie zależności
- *control.Dependency.Mapping*: mapowanie zależności
- *control.Dependency.Filtering*: filtrowanie na podstawie zależności
- *control.Dependency.Visibility*: wyświetlanie na podstawie zależności
- *button.click*: kliknięcie przycisku
- *menu.click*: kliknięcie menu
- *context.set*: ustawienie kontekstu formularza
- *context.update*: aktualizacja kontekstu formularza
- *action.execute*: wykonanie akcji
- *system.message.error*: systemowy komunikat o błędzie

Powiązane tematy:

- Diagnostyka w nAxiom
- Logi bezpieczeństwa
- Logi środowiska i bieżącego tenanta
- Konfiguracja logowania serwisów nAxiom

9.17.4. MiniProfiler for .NET

Podsumowanie:
MiniProfiler (https://miniprofiler.com/) to narzędzie do monitorowania żądań HTTP wysyłanych przez nAxiom podczas pracy użytkowników. Na podstawie jego danych można optymalizować wydajność aplikacji.

Korzystanie z profilera wymaga modyfikacji pliku *appsettings.json* dla serwisu *api*:

```
"Profiling": {
  "Enable": false,
  "ProfilerRouteBasePath": "/profiler",
  "EnableAsyncEventsProfiling": true
}
```

- *Enable*: włącza profiler: domyślnie *false*.
- *ProfilerRouteBasePath*: ścieżka bazowa punktów końcowych profilera; lista wyników profilera jest dostępna pod adresem: *{adres-internal-api}/{ProfilerRouteBasePath}/results-index* (np: https://witryna-naxiom/back/profiler/results-index)
- *EnableAsyncEventsProfiling*: określa czy poza żądaniami HTTP profilowane będą też zdarzenia asynchroniczne przetwarzane w systemie (np audyt, aktualizacja danych pod widok moje zadania, indeksowanie danych pod wyszukiwanie pełnotekstowe etc).

UWAGA
Profiler udostępnia podgląd wszystkich zapytań SQL wykonywanych przez aplikację, dlatego należy go włączać tylko w środowisku developerskim na potrzeby badania wydajności aplikacji.

9.18. OBSŁUGA MODELU TRZECH ŚRODOWISK

Podsumowanie:
Model pracy w trzech środowiskach zakłada zainstalowanie trzech witryn, z których każda odgrywa rolę odrębnego środowiska: deweloperskiego, testowego i produkcyjnego. Oprócz wizualnego rozróżnienia środowisk, model ten jest wspierany także przez możliwość definiowania zmiennych środowiskowych oraz określania w definicji akcji, w którym środowisku akcja ma być wykonywana.

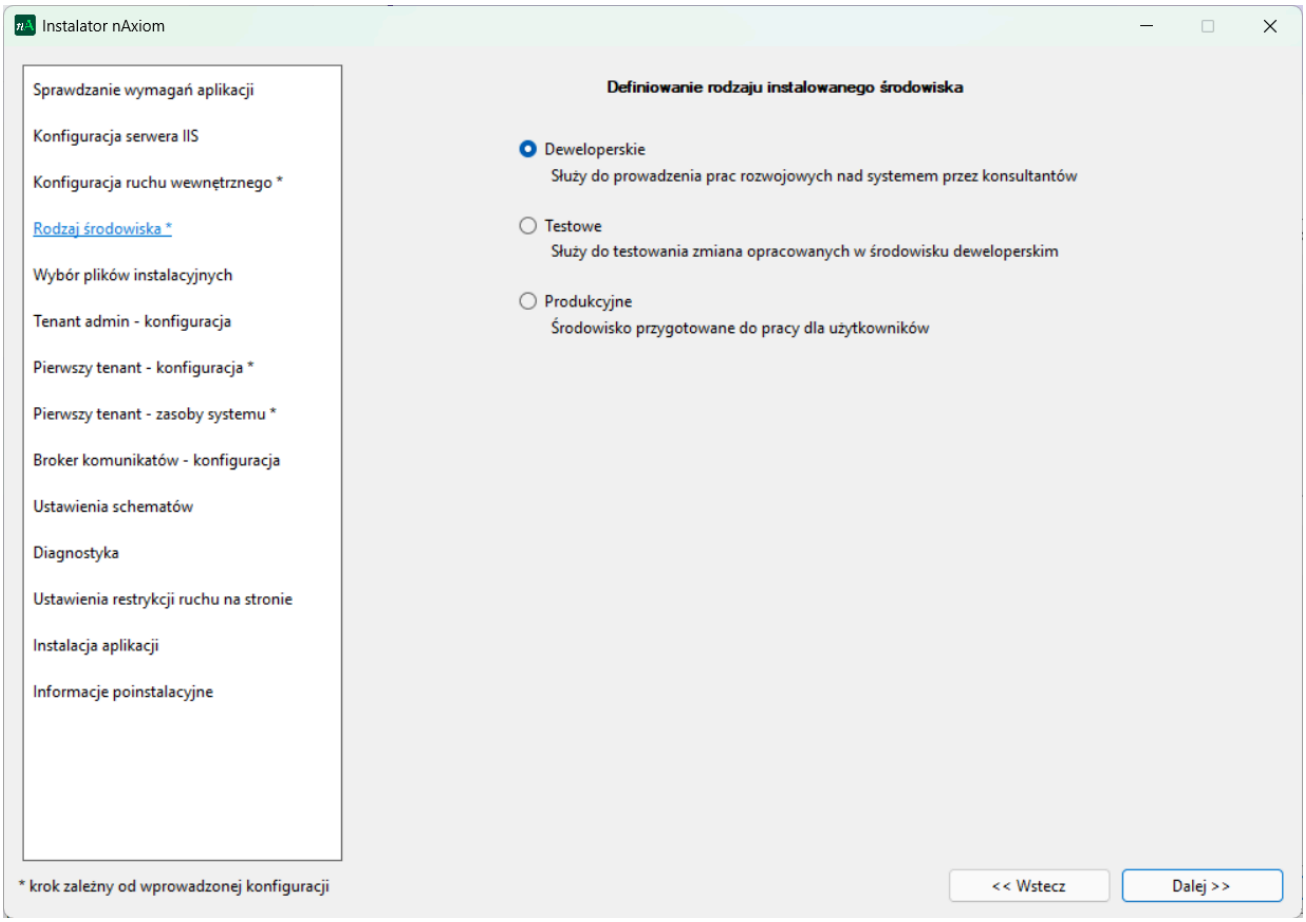
W TYM TEMACIE

- Wybór rodzaju środowiska
- Przełączanie trybu pracy
- Definiowanie akcji dla różnych środowisk

WYBÓR RODZAJU ŚRODOWISKA

Instalując witrynę nAxiom, można wybrać jej przeznaczenie: witryna może tworzyć środowisko deweloperskie (tworzenie aplikacji), testowe (sprawdzanie działania aplikacji) lub produkcyjne (przetwarzanie danych biznesowych).

Wybór tej opcji jest możliwy tylko podczas pierwszej instalacji witryny, dlatego ten krok jest oznaczony gwiazdką jako zależny od konfiguracji. Domyślnie wybrana jest opcja `Deweloperskie`. Ta opcja jest również stosowana do witryn aktualizowanych z wersji nAxiom wcześniejszych niż 1.8. Pozostałe wartości to: `Testowe` i `Produkcyjne`.



Wybór rodzaju środowiska

Odpowiednie ustawienia dotyczące rodzaju środowiska są zapisywane w plikach *appsettings.json* dla serwisów ***api***, ***auth***, ***taskservice***, ***tenantsapi***, ***publicapi***, ***mobileapi***,

bpmnapi i ***executorapi*** w kluczu *Environment*. Obsługiwane wartości to *Development*, *Test*, *Production*. Zmieniając tę wartość w obu plikach, można zmienić ustawienie środowiska dla witryny. Zastosowanie tej zmiany wymaga restartowania witryny (w Menedżerze usług IIS). W przypadku wdrożenia nAxiom z obrazów dokerowych zmiany można dokonać, używając plików *appsettings-custom.json* (patrz artykuł Wdrożenie nAxiom z obrazów Docker).

```
"AppConfiguration": {
  "Environment": "Test"
}
```

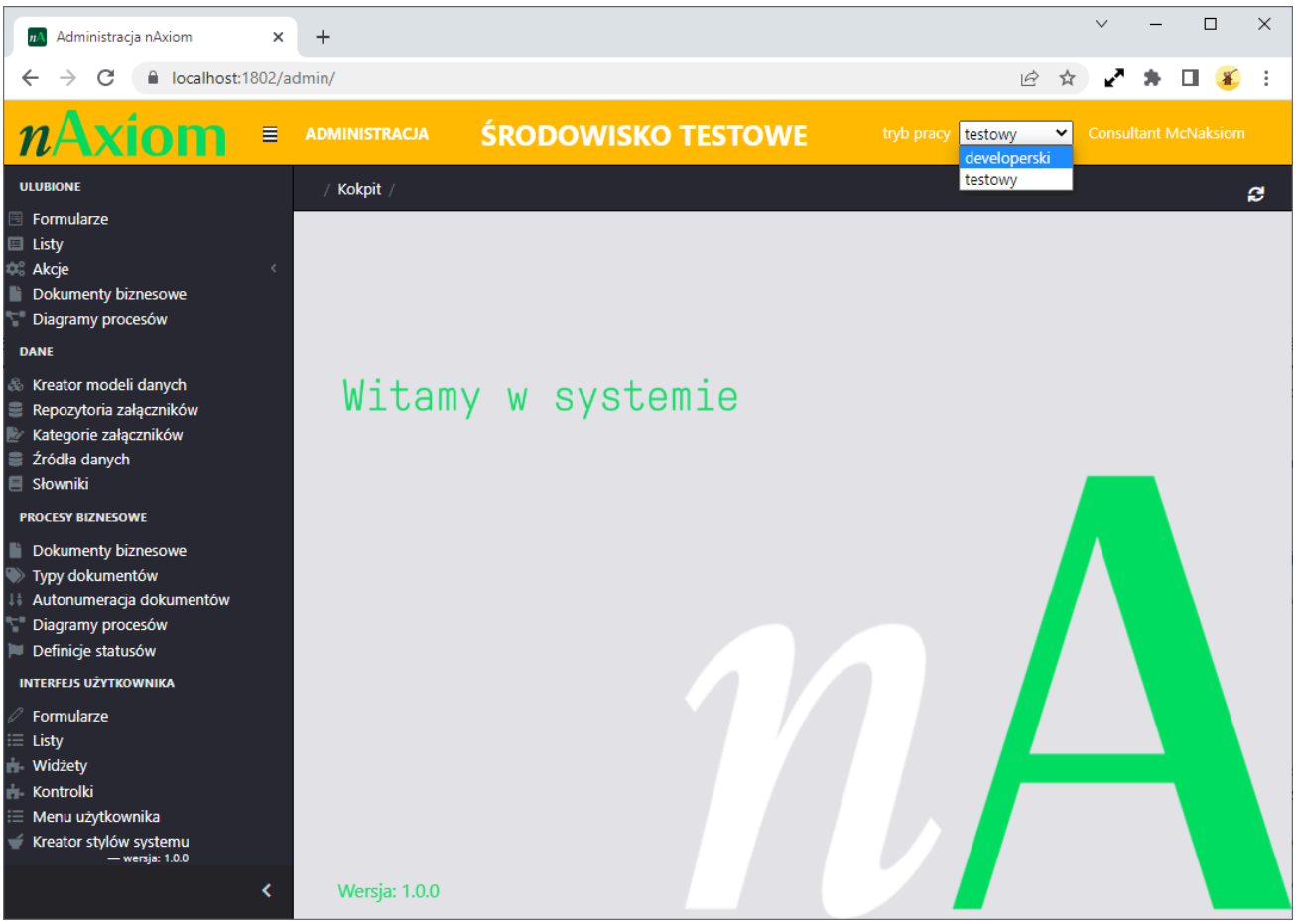
W środowisku deweloperskim i testowym witryna jest oznaczona kolorowym paskiem u góry okna przeglądarki z nazwą rodzaju środowiska (zarówno front office, jak i back office). Środowisko produkcyjne nie ma żadnych oznaczeń.

PRZEŁĄCZANIE TRYBU PRACY

W środowisku deweloperskim i testowym po stronie back office można zmienić tryb pracy danego środowiska, to jest wybrać wartości zmiennych środowiskowych, które mają być używane po stronie front office.

Aby zmienić wartości zmiennych środowiskowych używanych w danej witrynie, wybierz odpowiednią pozycję z listy rozwijanej `tryb pracy` na górnym pasku w back office. W środowisku deweloperskim można wybrać tryb pracy środowiska testowego, to znaczy, że po stronie front office będą wykorzystywane wartości zmiennych środowiskowych zdefiniowane dla środowiska testowego.

W środowisku testowym można wybrać pracę w trybie środowiska deweloperskiego. W środowisku produkcyjnym zmiana trybu nie jest możliwa; zmienne środowiskowe zawsze przyjmują tam wartości dla środowiska produkcyjnego.



Wybór trybu pracy

PRZEŁĄCZANIE TRYBU PRACY

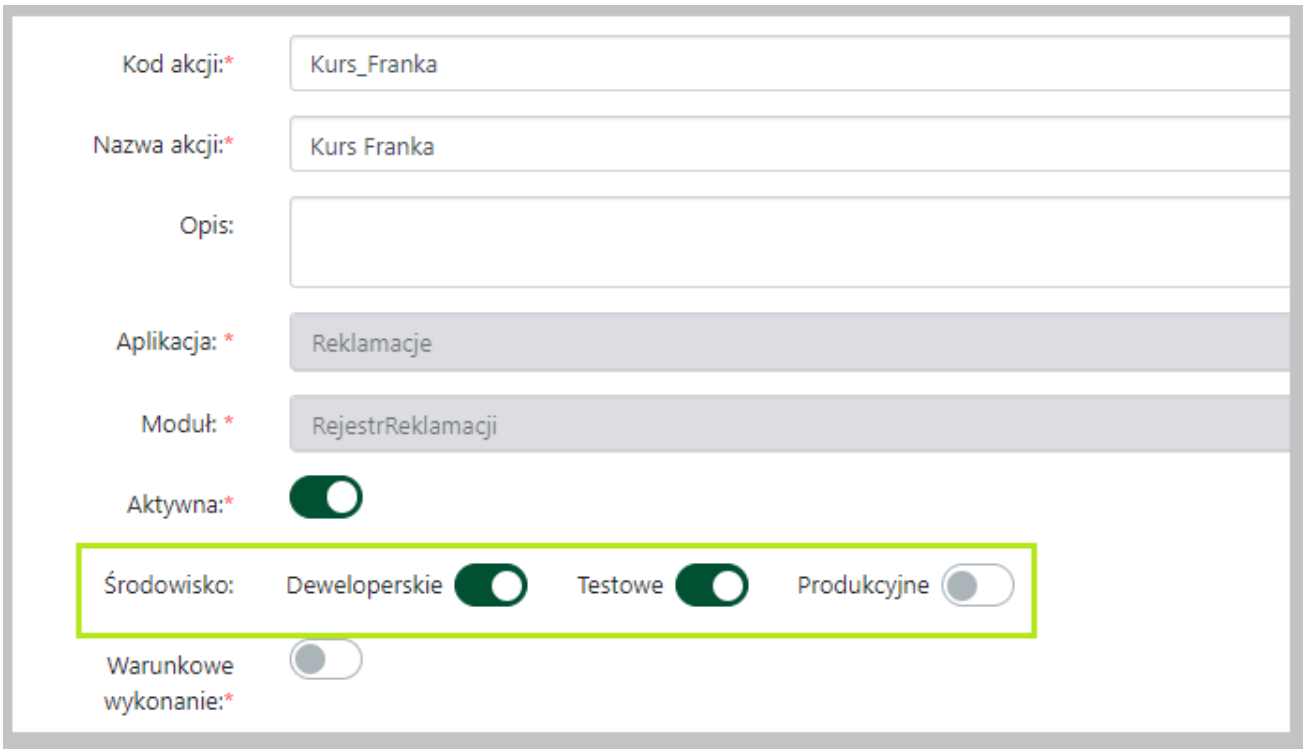
W nAxiom można określać środowisko, w którym mają być uruchamiane akcje. Ta funkcjonalność pozwala definiować zestawy akcji dedykowane dla określonych środowisk.

Wyświetl okno definiowania/edycji akcji, dla której chcesz przypisać środowisko i ustaw odpowiednie właściwości.

`Środowisko`: przełączniki decydujące o wykonywaniu akcji w określonym środowisku; jeśli przełącznik jest wyłączony, wykonanie akcji zostanie pominięte (przy czym do aplikacji będzie zwracana informacja o pomyślnym wykonaniu); aby używać różnych wersji akcji w różnych środowiskach, należy zdefiniować te akcje, odpowiednio poustawiać przełączniki i przypisać je wszystkie np. do przycisku na formularzu; domyślnie wszystkie przełączniki są włączone:

- `Deweloperskie`: decyduje o wykonywaniu akcji w środowisku deweloperskim
- `Testowe`: decyduje o wykonywaniu akcji w środowisku testowym
- `Produkcyjne`: decyduje o wykonywaniu akcji w środowisku produkcyjnym

Na przykład, aby w środowisku deweloperskim i testowym używać innej akcji niż w środowisku produkcyjnym, należy zdefiniować dwie akcje i w jednej włączyć tylko przełączniki środowiska deweloperskiego i testowego, a w drugiej tylko dla produkcyjnego. Następnie obie akcje należy przypisać do przycisku lub zdarzenia uruchamiającego. W zależności od środowiska uruchamiana będzie tylko jedna akcja, druga będzie ignorowana.



Wybór środowiska dla akcji

Powiązane tematy:

- Definiowanie zmiennych środowiskowych

9.19. INTEGRACJA NAXIOM Z USŁUGAMI MICROSOFT 365

Podsumowanie:
Integracja aplikacji biznesowych tworzonych w nAxiom z usługami Microsoft 365 wymaga korzystania z interfejsu API Microsoft Graph. Podstawową tej integracji jest zarejestrowanie aplikacji w usługach Azure Active Directory (AAD) oraz przyznanie jej odpowiednich uprawnień umożliwiających wysyłanie uprawnień do interfejsu MS Graph API. Te uprawnienia musi przydzielić administrator AAD w organizacji. Taką integrację można wykorzystać na przykład do monitorowania wskazanej skrzynki poczty elektronicznej lub do pobierania i tworzenia wydarzeń z kalendarza w MS Outlook.

W TYM TEMACIE

- Wprowadzenie
- Rejestracja aplikacji w usłudze Azure Active Directory
- Przypisywanie uprawnień
- Źródło danych w nAxiom
- Pobieranie zdarzeń z kalendarza

WPROWADZENIE

Dokumentacja punktów końcowych interfejsu MS Graph API jest dostępna na stronie https://learn.microsoft.com/en-us/graph/overview?view=graph-rest-1.0.

Firma Microsoft udostępnia specjalne narzędzie, przy użyciu którego można testować zapytania do interfejsu: https://developer.microsoft.com/en-us/graph/graph-explorer.

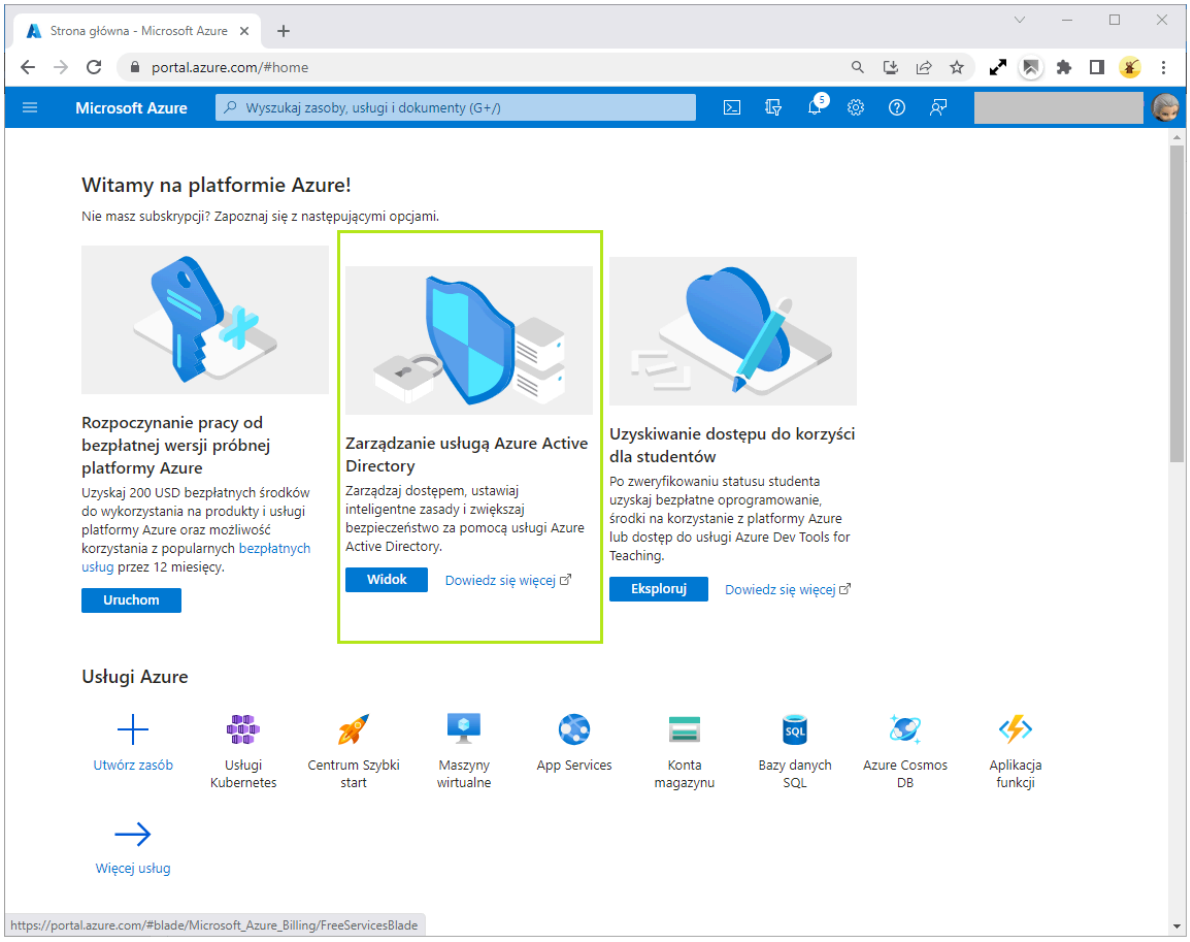
Aplikacja może wysyłać żądania do interfejsu API na dwa sposoby:

- jako zalogowany użytkownik — wymagane są uprawnienia delegowane,
- jako klient (aplikacja) — wymagane są uprawnienia aplikacji.

W dalszej części opisany jest ten drugi przypadek. Należy pamiętać, że uprawnienia aplikacji (klienta) dotyczą wszystkich użytkowników w danej dzierżawie. Administrator AAD może ograniczyć zakres tych uprawnień do określonych kont/użytkowników w sposób opisany na stronie: https://learn.microsoft.com/en-us/graph/auth-limit-mailbox-access.

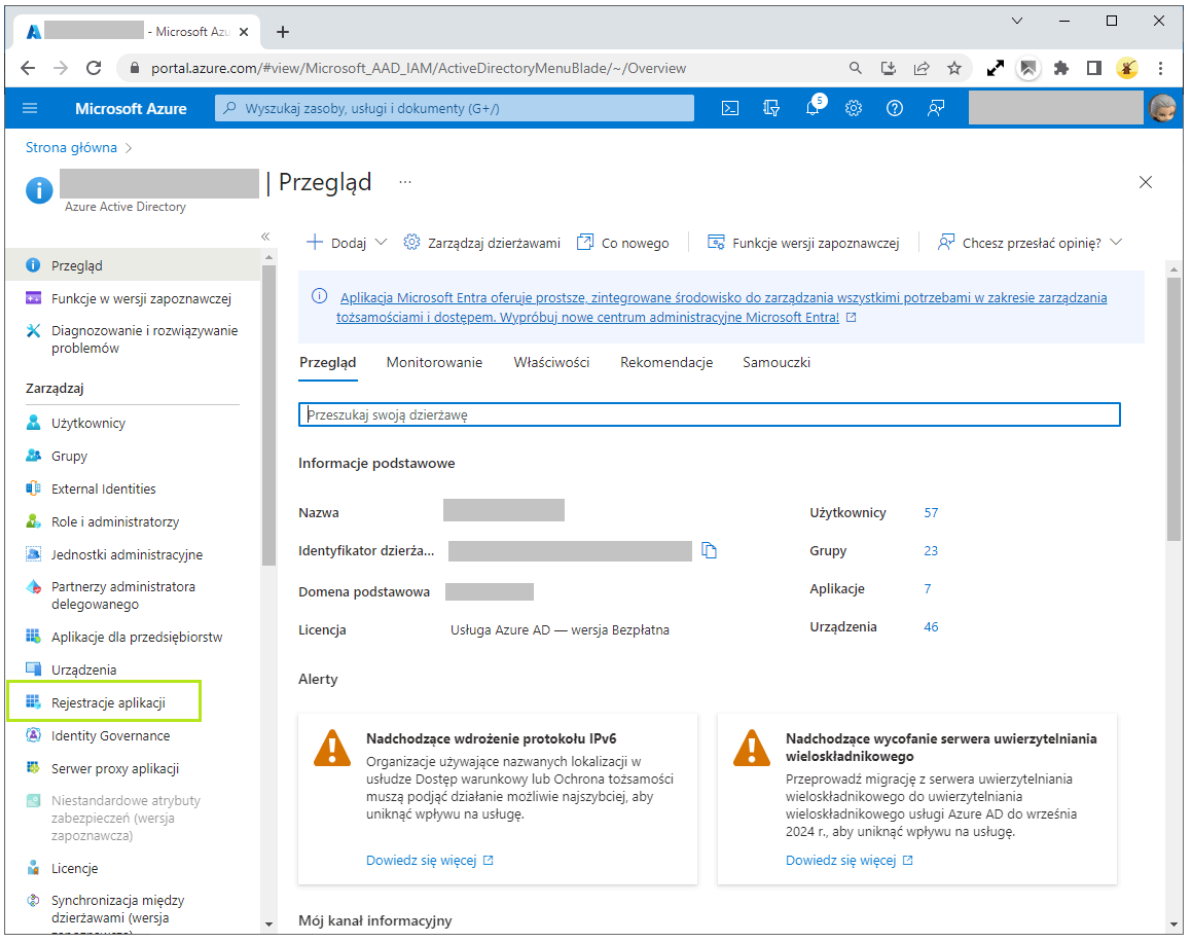
REJESTRACJA APLIKACJI W USŁUDZE AZURE ACTIVE DIRECTORY

1. Zaloguj się do portalu Azure (portal.azure.com).
2. Na ekranie głównym wybierz `Zarządzanie usługą Active Directory` (środkowy kafelek).



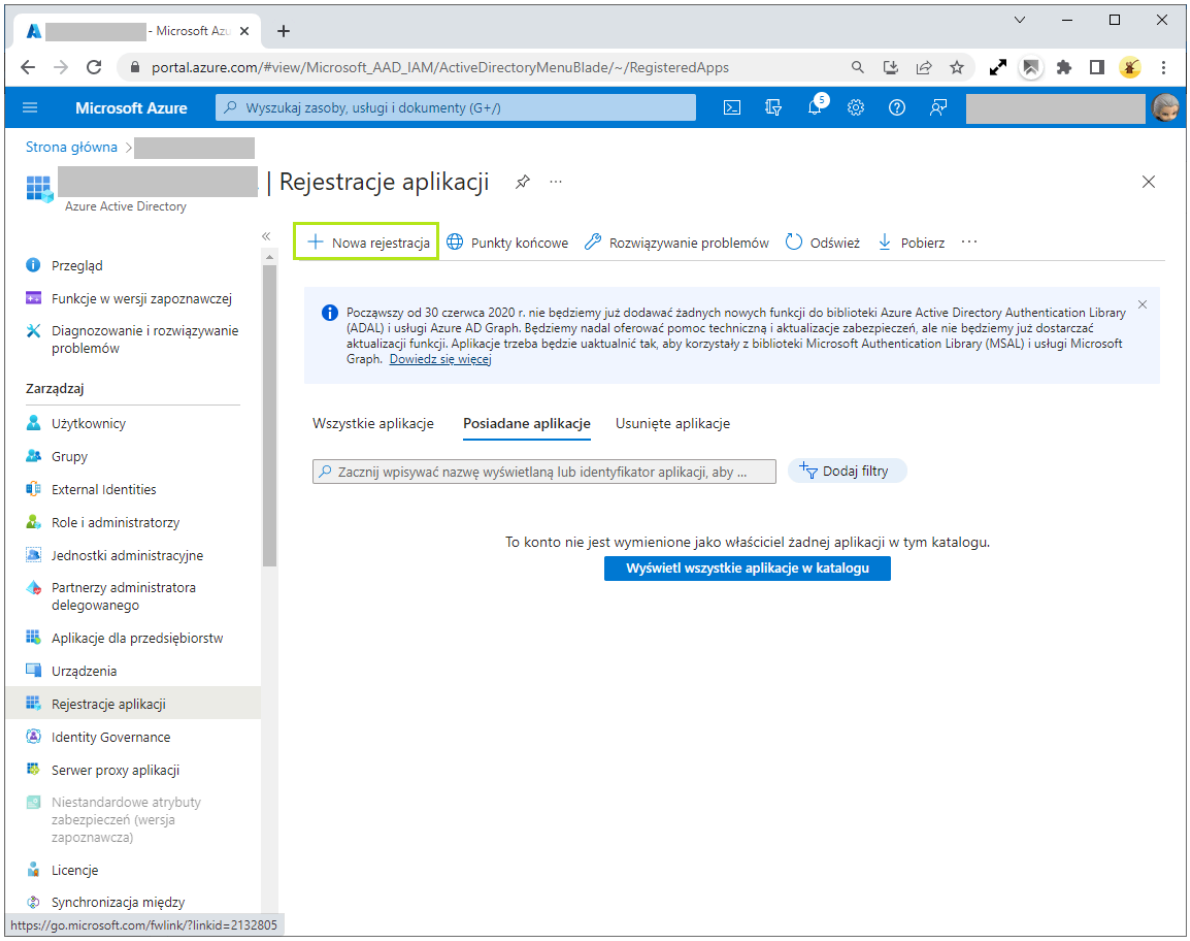
Portal Azure, Zarządzanie AD

3. W menu z lewej strony kliknij pozycję `Rejestracje aplikacji`.



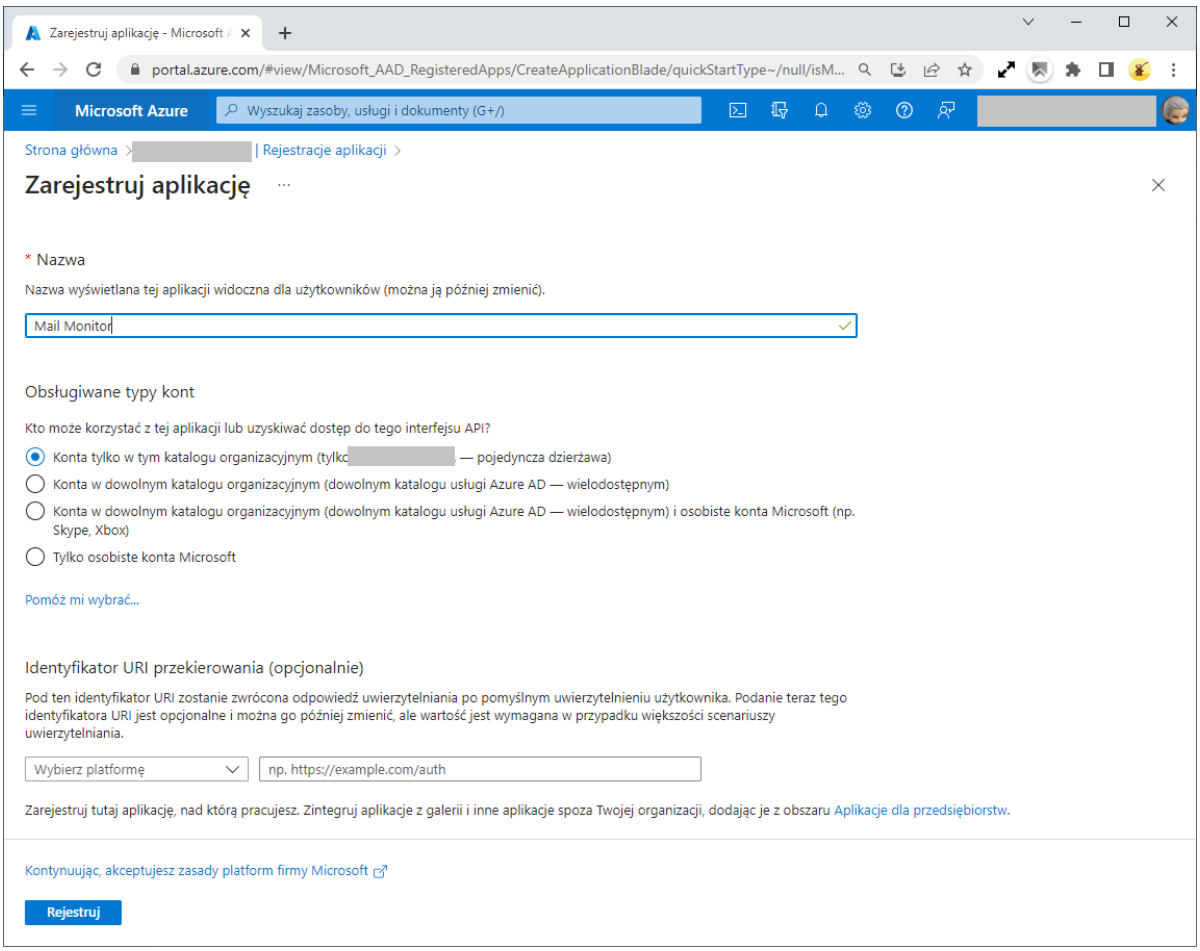
Portal Azure, Rejestracje aplikacji

4. Na wyświetlonej stronie kliknij link `Nowa rejestracja` u góry po lewej.



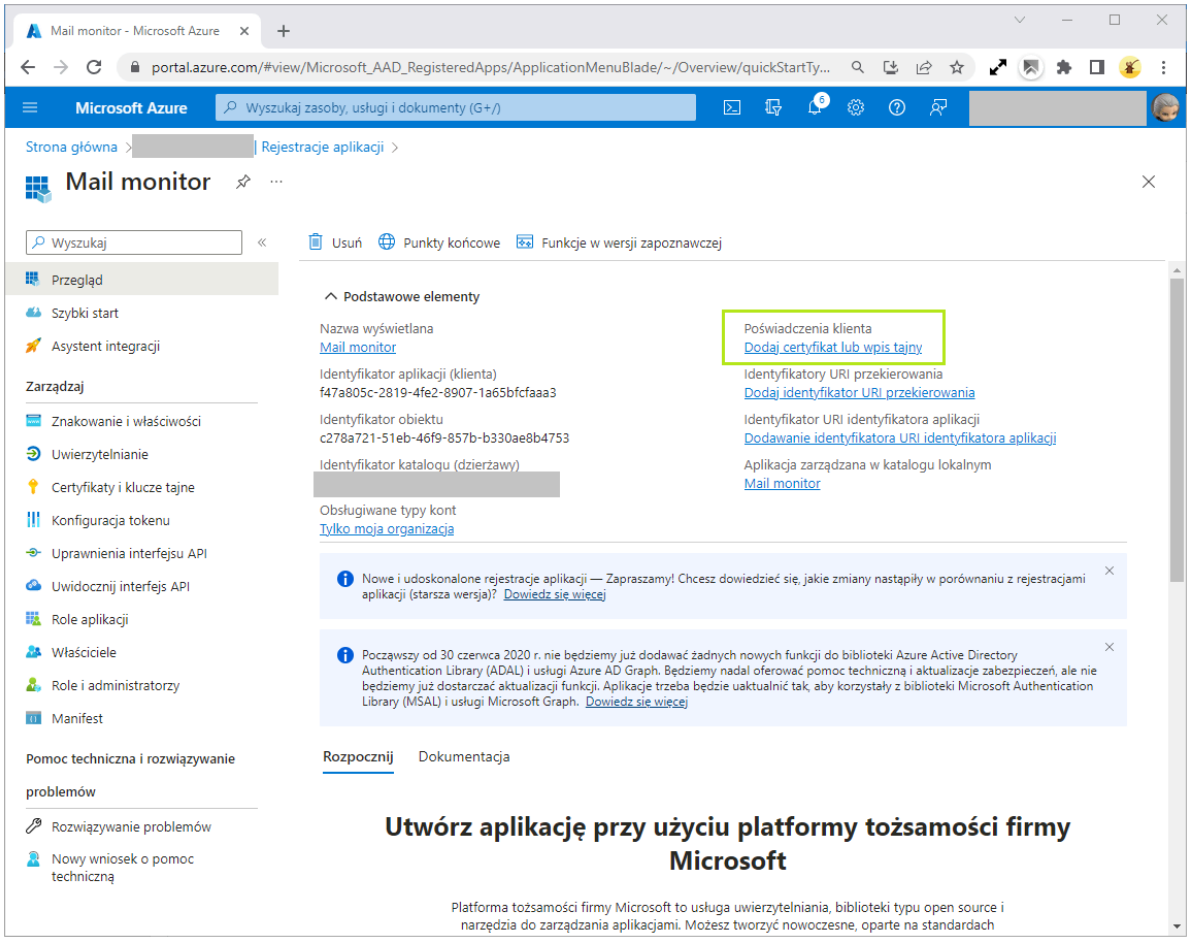
Portal Azure, Nowa rejestracja

5. Wpisz nazwę aplikacji i kliknij przycisk `Rejestruj` u dołu.



Rejestrowanie aplikacji

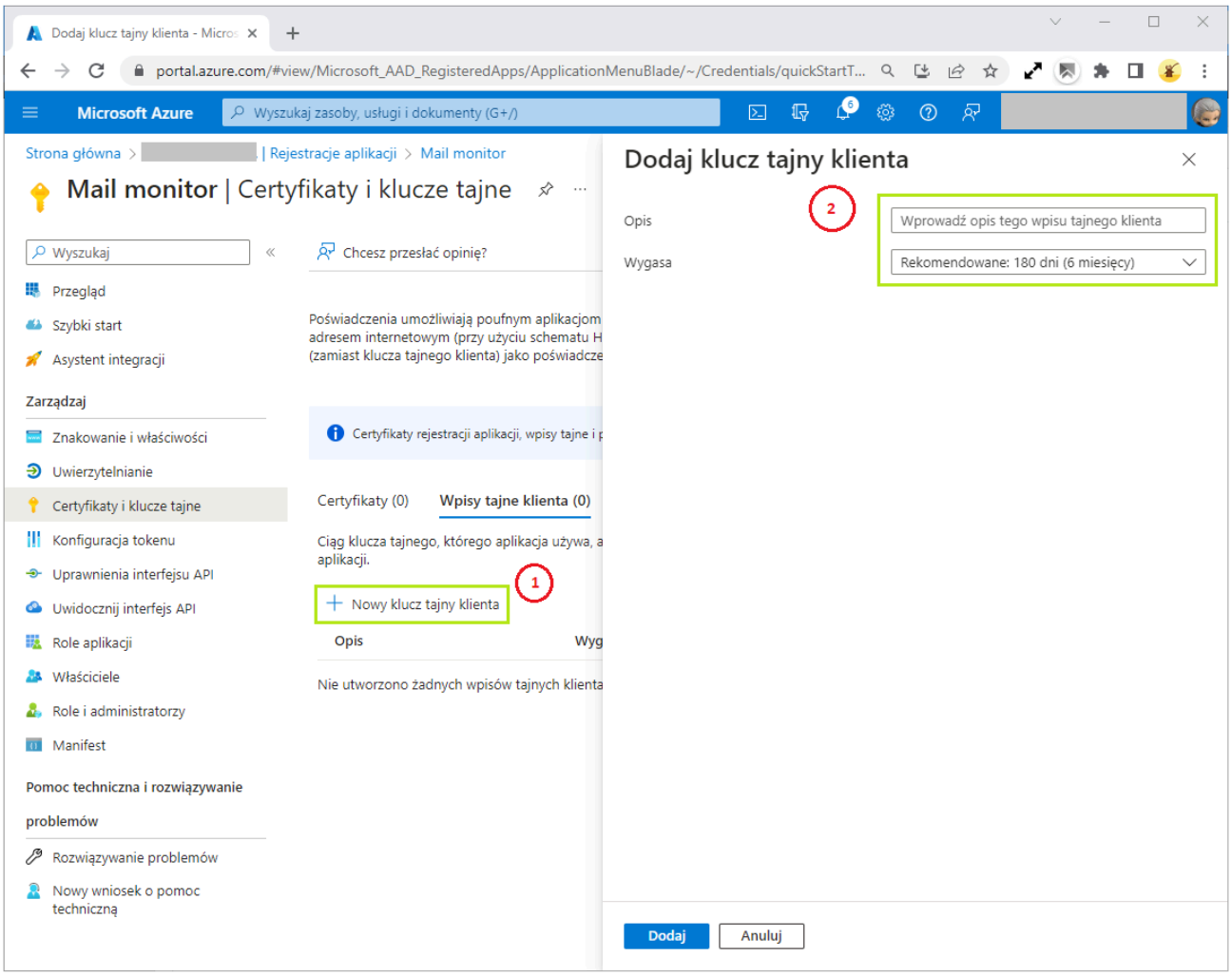
6. Kliknij link `Dodaj certyfikat lub wpis tajny`.



Dodaj certyfikat lub wpis tajny

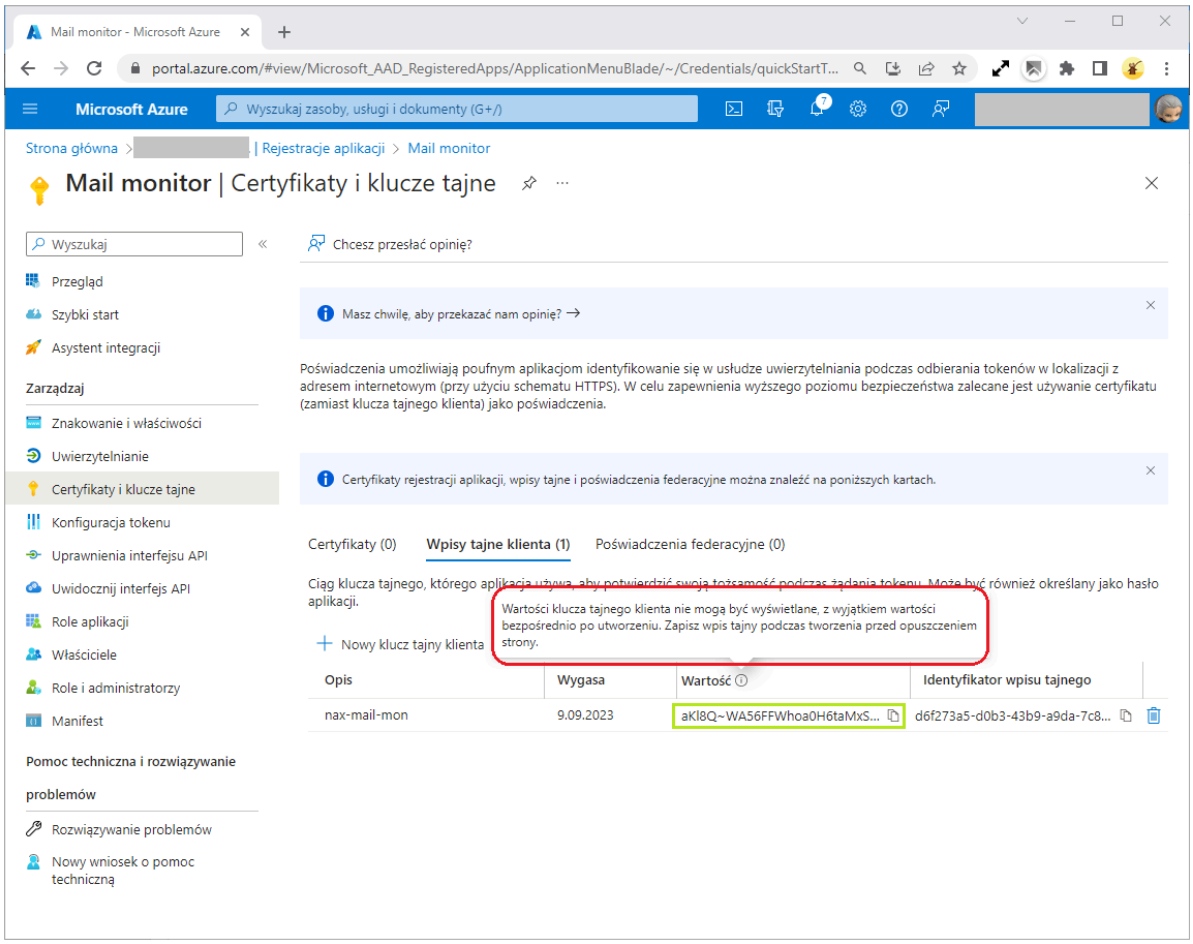
7. Kliknij link `Nowy klucz tajny klienta`, a następnie wpisz opis i wybierz okres ważności.

> **UWAGA**
> należy pamiętać o konieczności aktualizacji klucza tajnego zgodnie z ustawionym okresem ważności.



Nowy klucz tajny

1. Po kliknięciu przycisku `Dodaj` zostanie wyświetlona wartość wpisu tajnego klienta, którą należy skopiować i zapisać w bezpiecznym miejscu, ponieważ po wylogowaniu nie będzie ona więcej wyświetlana

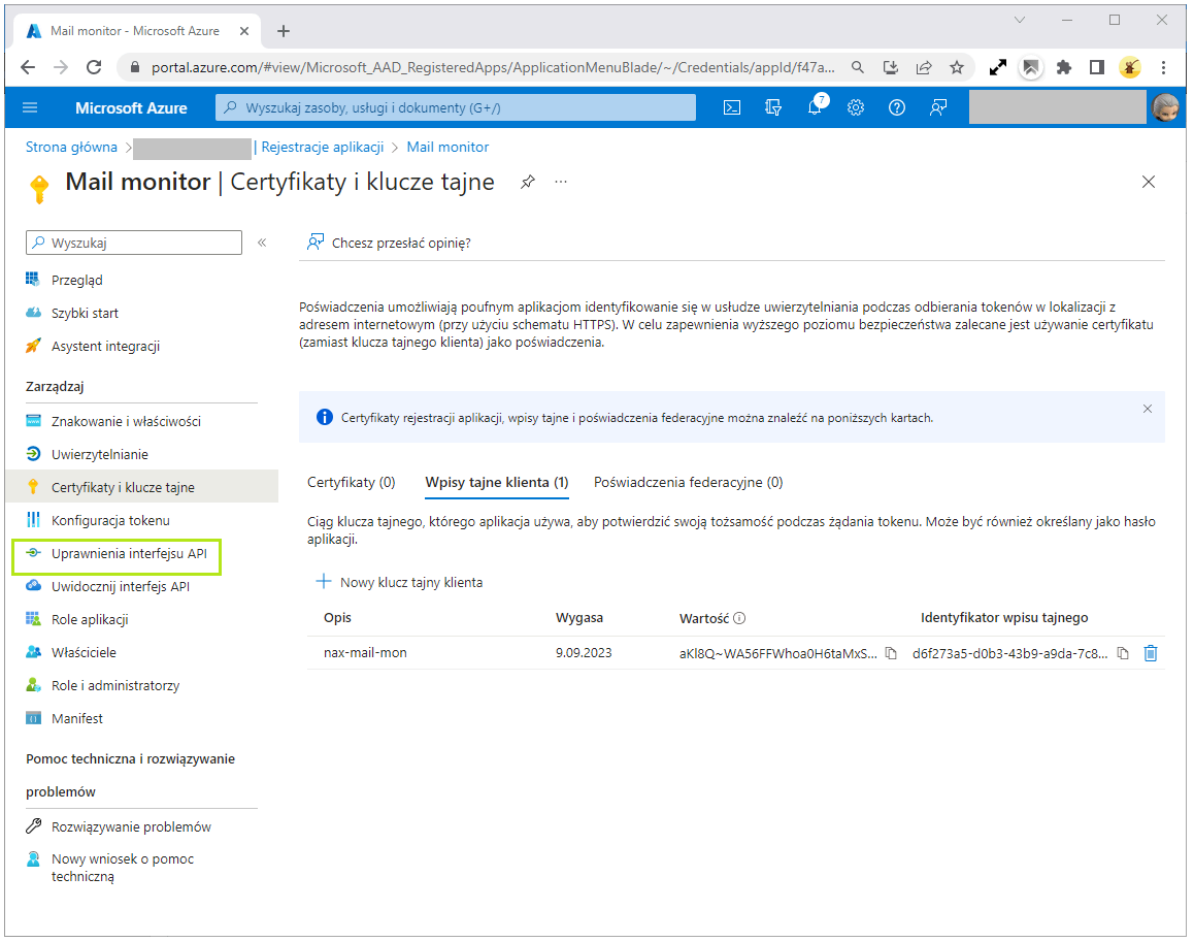


Wartość klucza tajnego

PRZYPISYWANIE UPRAWNIEŃ

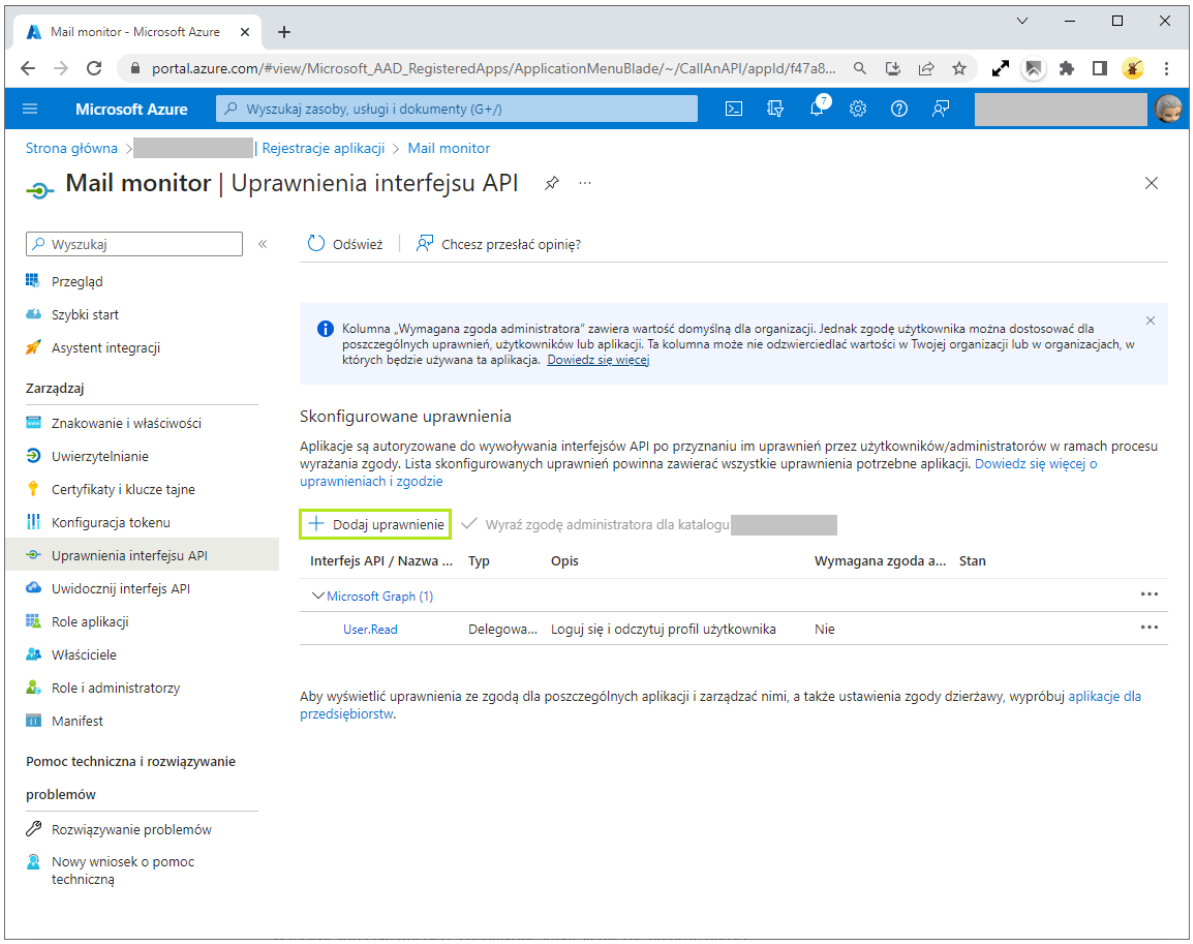
Po zarejestrowaniu aplikacji należy przypisać jej odpowiednie uprawnienia w interfejsie MS Graph API. Wybór opcji zależy od zasobów, z których ma korzystać aplikacja.

1. Wyświetl stronę zarejestrowanej aplikacji i kliknij polecenie `Uprawnienia interfejsu API` w menu z lewej strony.



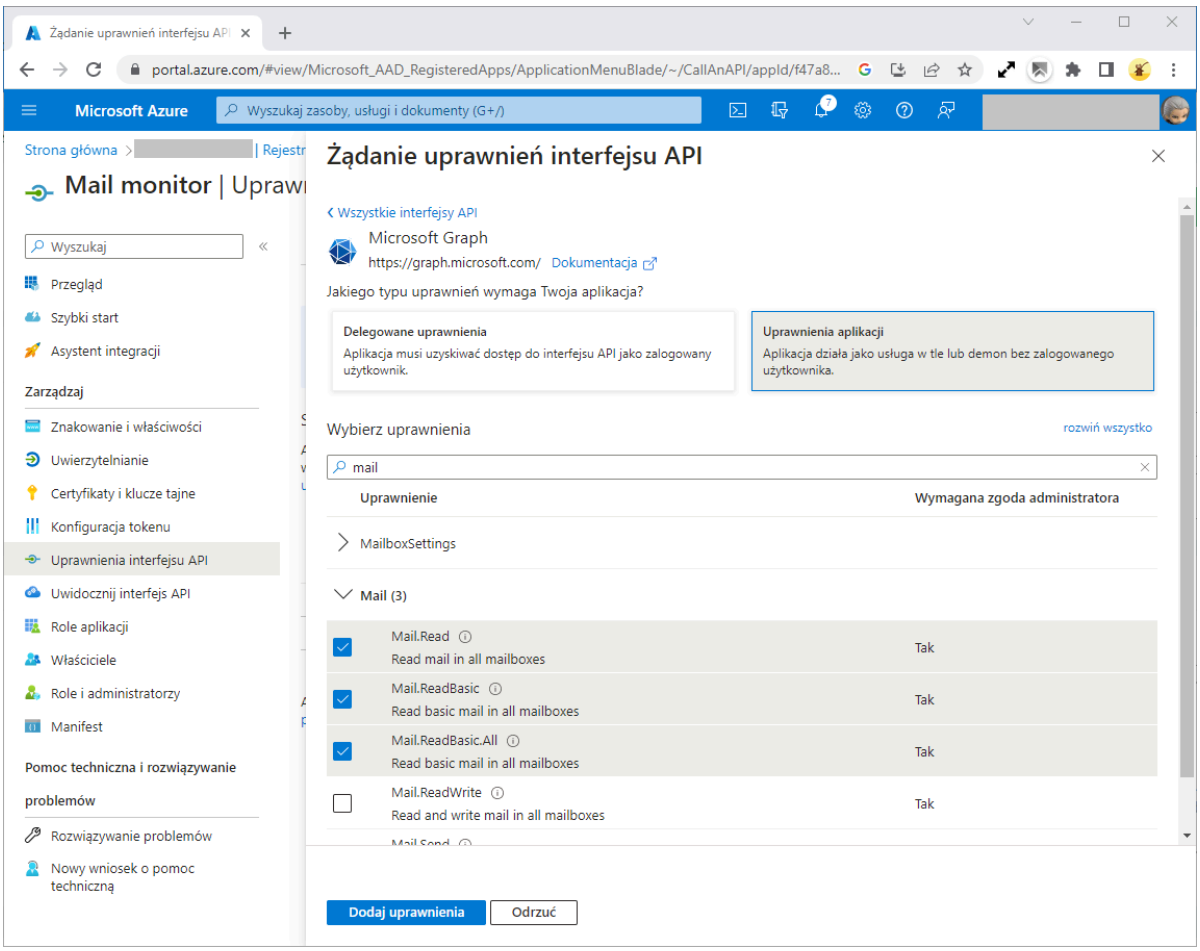
Uprawnienia interfejsu API

2. Zarejestrowana aplikacja ma domyślnie przypisane delegowane uprawnienie *User.Read*. Kliknij `Dodaj uprawnienie`.



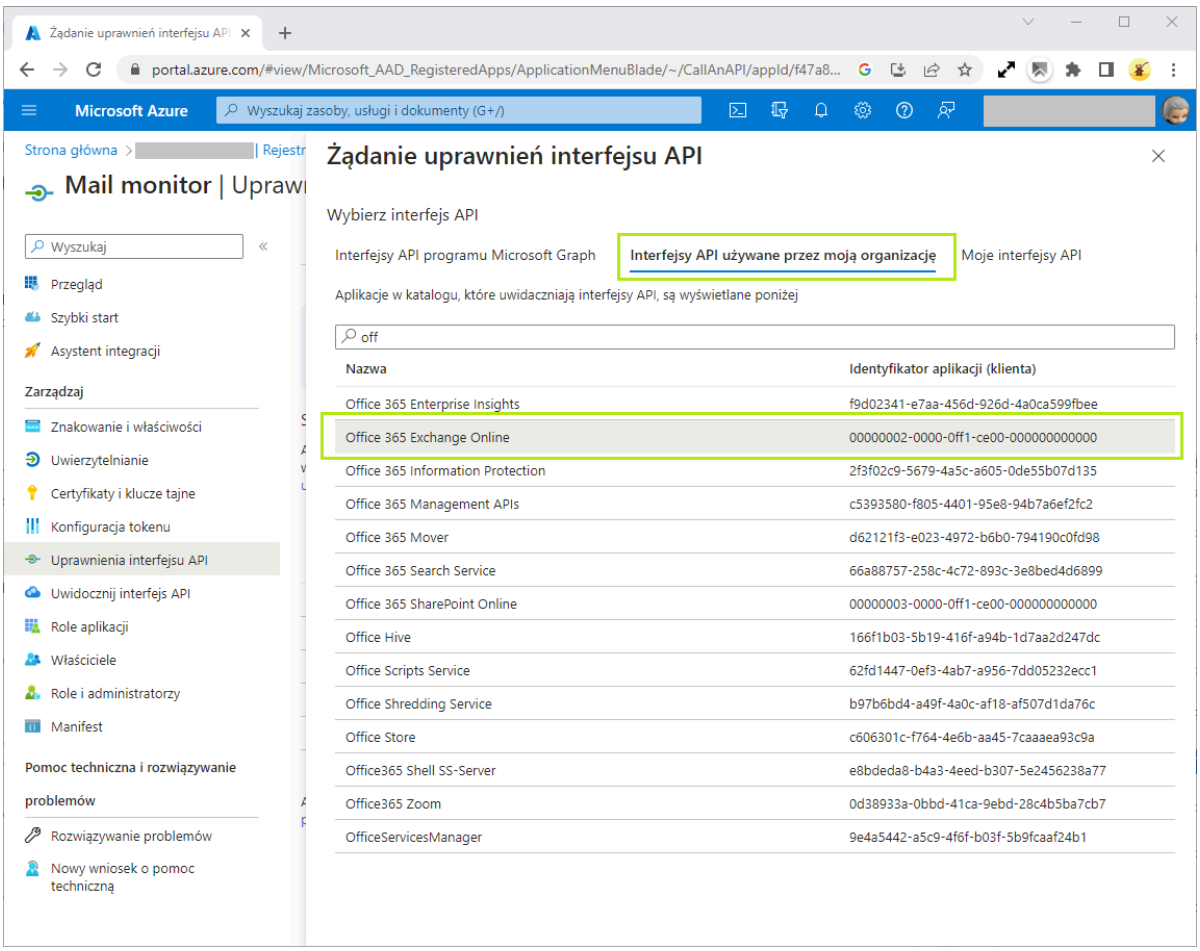
Dodaj uprawnienie

3. Dostęp do skrzynek pocztowych przez MS Graph API:
   Na stronie `Żądanie uprawnień interfejsu API` stronie kliknij kafelek `MS Graph`, a na następnej kafelek `Uprawnienia aplikacji`. Przewiń do sekcji `Mail`, zaznacz uprawnienia *Mail.Read*, *Mail.ReadBasic* i *Mail.ReadBasic.All* i kliknij przycisk `Dodaj uprawnienie`.



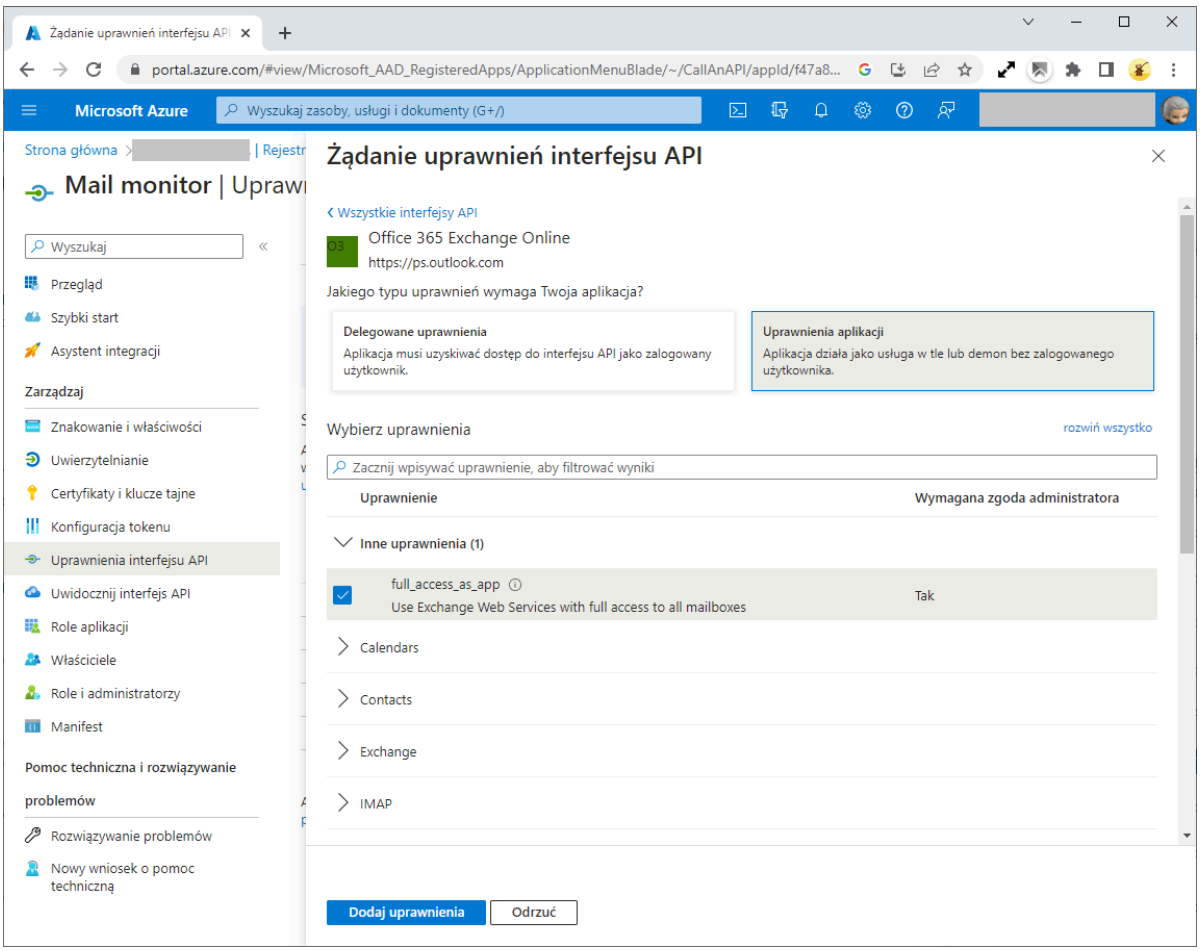
Uprawnienie

4. Dostęp do skrzynek pocztowych przez Exchange Web Services:
   a. Na stronie `Żądanie uprawnień interfejsu API` kliknij kategorię `Interfejsy API używane przez moją organizację` i znajdź pozycję `Office 365 Exchange Online`.



Interfejsy API organizacji

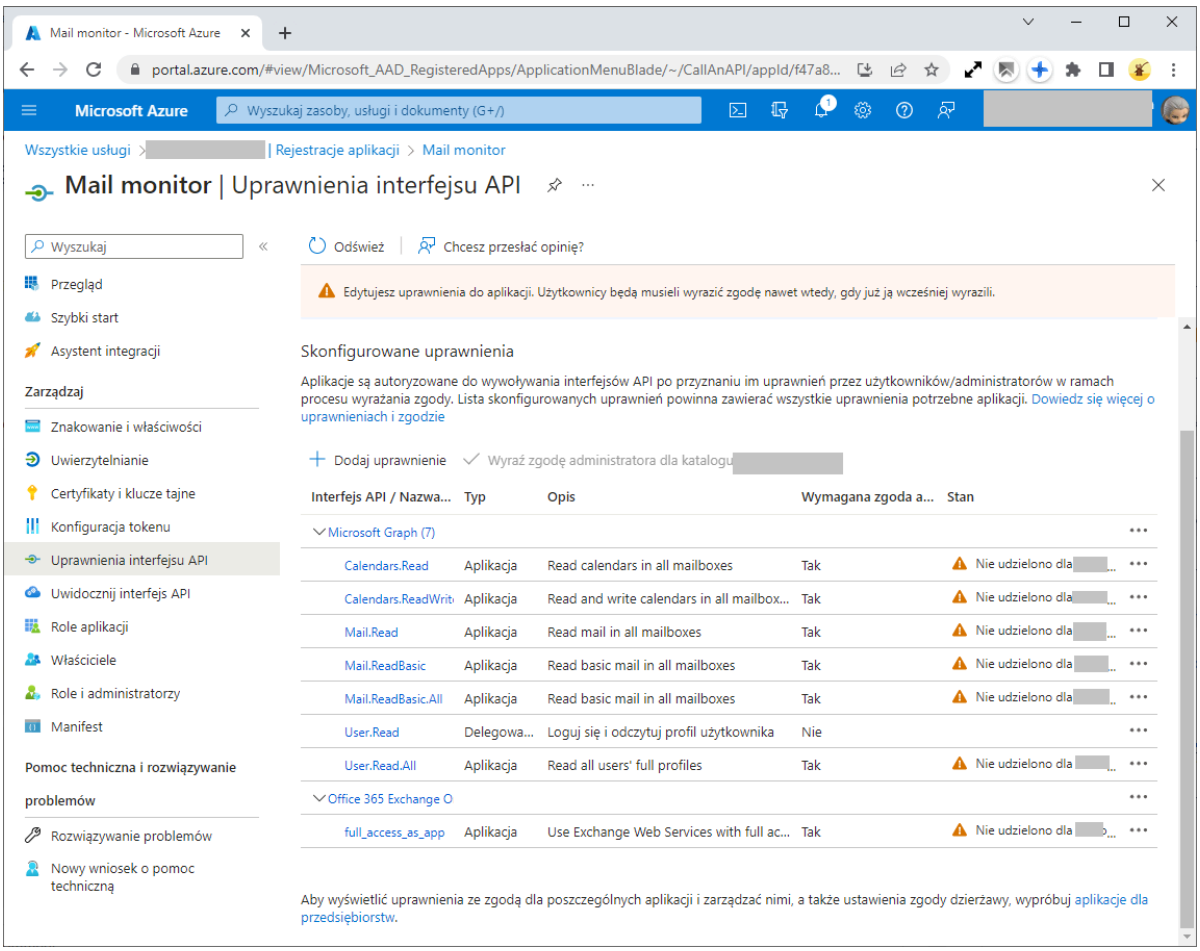
b. Kliknij tę pozycję, a następnie kliknij kafelek `Uprawnienia aplikacji` i zaznacz uprawnienie *full_access_as_app*.



Uprawnienie full_access_as_app

5. Dostęp do kalendarzy przez interfejs MS Graph:
   Na stronie `Żądanie uprawnień interfejsu API` stronie kliknij kafelek `MS Graph`, a na następnej kafelek `Uprawnienia aplikacji`. Przewiń do sekcji `Calendars`, zaznacz uprawnienia *Calendars.Read*, *Calendars.ReadWrite, a następnie w sekcji* `Users` *zaznacz uprawnienie *Users.Read.All* i kliknij przycisk `Dodaj uprawnienie`.

6. Jeśli użytkownik nie ma uprawnień administracyjnych, włączenie dodanych uprawnień wymaga zgody administratora.



Nieaktywne uprawnienia

ŹRÓDŁO DANYCH W NAXIOM

W aplikacji *AdminSPA* wybierz polecenie `DANE` > `Źródła danych` i zdefiniuj nowe źródło danych typu REST API na serwerze *graph.microsoft.com*, podając właściwości jak na ilustracji poniżej.

Edycja źródła danych

Kod:*	MsGraph
Nazwa:*	MsGraph
Aplikacja: *	SandboxApp
Moduł: *	SandboxModule
Typ źródła danych:*	Api: REST
Typ uwierzytelnienia:*	OAuth 2.0 Client Credentials
Adres serwera:*	https://graph.microsoft.com
Open api url:	
Open api plik:	Wybierz plik Nie wybrano pliku
Serwer autoryzacji:	https://login.microsoftonline.com/(Identyfikator dzierżawcy)/oauth2/v2.0/token
Scope:	https://graph.microsoft.com/.default
Client ID:	(identyfikator aplikacji klienta w Azure AD)
Client Secret:	(klucz tajny aplikacji klienta)

Nowe źródło danych

W polach `Client ID` i `Client Secret` należy wpisać odpowiednio identyfikator klienta (aplikacji) i klucz tajny uzyskane podczas rejestrowania aplikacji w usłudze AAD.

POBIERANIE ZDARZEŃ Z KALENDARZA

1. Utwórz nową akcję Zapytanie do Webserwisów.
2. Wybierz źródło danych na serwerze MS GRAPH.
3. W polu `Ścieżka żądania` wstaw adres punktu końcowego: *https://graph.microsoft.com/v1.0/users/{@UserId}/calendar/events*.
4. W polu `Procedura obsługi odpowiedzi` wpisz instrukcję SQL, która przetworzy zwrócone dane, na przykład:

```
INSERT INTO [MojaTabela]  (
  [GraphUser],
  [StartDatetime],
  [EndDatetime],
  [Title],
  [IsAllDay]
)
SELECT
  {@ IdentyfikatorUżytkownika},
  [StartDatetime],
  [EndDatetime],
```

```
  [Title],
  0
FROM OPENJSON ({@_Response},'$.value')
WITH (
  [StartDatetime] datetime2 '$.start.dateTime',
  [EndDatetime] datetime2 '$.end.dateTime',
  [Title] nvarchar(300) '$.subject'
  )
```

Następnie można utworzyć formularz z sekcją kalendarza, w której będą wyświetlane pobrane wydarzenia. Aby zdarzenia w kalendarzu były aktualne, akcję pobierającą można skonfigurować jako zadanie cykliczne.

Powiązane tematy:

- Źródła danych
- Akcja zapytań do web service
- Mail Monitor
- Sekcje formularza - Kalendarz

9.20. WYSYŁANIE I ODBIERANIE KORESPONDENCJI EPUAP

Podsumowanie:
nAxiom oferuje narzędzia umożliwiające współpracę z ePUAP — Elektroniczną Platformą Usług Administracji Publicznej. Korzystając z tych narzędzi, w aplikacjach nAxiom można wymieniać korespondencję z jednostkami administracji publicznej.

Proces poczty wychodzącej polega na utworzeniu odpowiedniego typu dokumentu XML z załącznikami lub bez, jego podpisaniu (opcjonalnie) i przesłaniu do adresata na odpowiedni adres skrytki.

W TYM TEMACIE

- Konfiguracja ePUAP Service
- Przykład pełnej konfiguracji
- Instalacja i deinstalacja usługi ePUAP
- Utworzenie pisma XML
- Podpis pisma
- Wysłanie pisma
- Poczta przychodząca
- Nierozpoznane rodzaje pism
- Rozpoznane rodzaje pism
- Książka adresowa ePUAP
- Dodatek A. Przykład pliku konfiguracyjnego
- Dodatek B. Przykład pełnej konfiguracji
- Dodatek C. Przykładowe pisma ePUAP

KONFIGURACJA EPUAP SERVICE

W celu uruchomienia integracji z Elektroniczną Skrzynką Podawczą ePUAP należy skonfigurować odpowiednio usługę odpowiedzialną za synchronizację książki adresowej, pobieranie korespondencji oraz akcję wysyłki dokumentów. Ustawienia konfiguracyjne znajdują się w pliku *appsettings.json* komponentu *NAX.Modules.ePUAP.Service*.

Przykład pliku konfiguracyjnego, który należy uzupełnić, zamieszczono w dodatku A.

Opis parametrów:

UWAGA
Klucze pogrubione są wymagane.

- ***AuthUrl***: adres do serwisu uwierzytelniającego
- ***ServiceUrl***: adres udostępniania usługi ePUAP
- ***PublicApiUrl***: adres serwisu Public API
- *LogResponseXml*: flaga zapisu odebranych pism na dysku w formacie XML
- *LogsPath*: ścieżka dla logów
- *AllowedIPs*: adresy, z których można łączyć się z usługą ePUAP
- *AddressesConfiguration*: konfiguracja synchronizacji książki adresowej; nie jest wymagana; przykładowa konfiguracja:
    - ***RepeatInterval***: interwał synchronizacji w milisekundach

    - ***Url***: adres pobierania pliku CSV z książką adresową
    - ***TableSchema***: schemat tabeli książki adresowej
    - ***TableName***: nazwa tabeli książki adresowej
    - ***KeyColumn***: klucz główny
    - ***NameColumn***: nazwa podmiotu
    - ***RegonColumn***: REGON
    - ***StreetColumn***: ulica
    - ***PostalCodeColumn***: kod pocztowy
    - ***CityColumn***: miejscowość
    - ***MailboxAddressColumn***: adres skrytki
    - ***MailboxNameColumn***: nazwa skrytki
- *IntegrationConfigurations*: lista konfiguracji skrytek do integracji
    - ***Active***: flaga aktywności konfiguracji
    - ***ValidateSignature***: flaga walidacji podpisu pisma (status podpisu, podgląd podpisu)
    - ***EpuapDnsIdentity***: identyfikator DNS platformy ePUAP
    - ***PZDnsIdentity***: identyfikator DNS platformy Profil Zaufany
    - ***ClientCertificateName***: wartość *Wystawiony dla* z certyfikatu klienta (certyfikat z magazynu certyfikatów maszyny w kontenerze *Osobisty*)
    - ***EpuapServerCertificateName***: wartość *Wystawiony dla* z certyfikatu serwera ePUAP (certyfikat z magazynu certyfikatów maszyny w kontenerze *Zaufane osoby*)
    - ***PZServerCertificateName***: wartość *Wystawiony dla* z certyfikatu serwera PZ (certyfikat z magazynu certyfikatów maszyny w kontenerze *Zaufane osoby*)wartość wystawiony dla certyfikatu serwera PZ (certyfikat umieszczony w certyfikatach maszyny w kontenerze 'Zaufane osoby'),
    - ***EntityName***: nazwa podmiotu publicznego klienta ePUAP
    - ***MailboxName***: nazwa skrytki podmiotu publicznego klienta ePUAP
    - ***MailboxAddress***: adres skrytki podmiotu publicznego klienta ePUAP
    - ***EpuapPullServiceUrl***: adres web serwisu *PULL ePUAP* do pobierania dokumentów
    - ***EpuapPOBoxServiceUrl***: adres web serwisu *SKRYTKA ePUAP* do wysyłki pism
    - ***TpSigningServiceUrl***: adres web serwisu *TpSigning5* Profilu Zaufanego do podpisu pism
    - ***FileRepoServiceUrl***: adres web serwisu *FileRepoService* do wysyłania i pobierania załączników pism
    - ***UnrecognizedDocumentConfiguration***: konfiguracja nierozpoznanego typu

pisma (tj. pisma, które nie będzie odbierane lub dla którego nie zostanie skonfigurowany rozpoznany typ pisma)

- ***SaveXmlAttachment***: flaga zapisu odebranego pisma jako załącznika w formacie XML
- ***BusinessDocumentConfiguration***: konfiguracja dokumentu biznesowego i kategorii załączników (opcjonalnie, w razie braku kategorii wszystkie załączniki zostaną zapisane w kategorii domyślnej dokumentu, o ile istnieje)
    - ***DocumentDefinitionId***: ID definicji dokumentu biznesowego
    - ***FormDefinitionId***: ID definicji formularza
    - *AttachmentsCategoryId*: ID kategorii załączników pobranych z pisma ePUAP
    - *XmlAttachmentCategoryId*: ID kategorii załącznika dla pisma w formacie XML
    - *XmlPreviewAttachmentCategoryId*: ID kategorii załącznika dla podglądu pisma ePUAP w formacie PDF
    - *SignaturePreviewAttachmentCategoryId*: ID kategorii załącznika dla podglądu podpisu pisma ePUAP
- ***RequiredFieldsConfiguration***: wymagane pola na dane, które muszą zostać zapisane po pobraniu pisma
    - ***CaseId***: pole ID sprawy ePUAP, pole wymagane dla ewentualnej odpowiedzi na pismo
    - ***DocumentId***: pole ID dokumentu ePUAP
    - ***EntityName***: pole nazwy podmiotu, do którego przyszło pismo
    - ***MailboxAddress***: pole adresu skrytki, do której przyszło pismo
    - ***MailboxName***: pole nazwy skrytki, do której przyszło pismo
- *SignatureValidationFieldsConfiguration*: konfiguracja pól dotyczących informacji o podpisie
    - ***DocumentSignatureStatus***: pole na status podpisu pisma ePUAP
- *StaticFieldsConfiguration*: konfiguracja pól statycznych dostępnych podczas pobierania pisma z ePUAP; te pola, w szczególności pole *ResponseMailboxAddress*, będą przydatne, jeśli planowane jest odpowiadanie na pisma ePUAP
    - *SubjectId*: pole identyfikatora podmiotu nadawcy
    - *PersonType*: pole typu osoby nadawcy
    - *FirstName*: pole imienia nadawcy

            - *LastName*: pole nazwiska/nazwy nadawcy
            - *Pesel*: pole numeru PESEL nadawcy (osoba fizyczna)
            - *Nip*: pole numeru NIP
            - *Regon*: pole numeru REGON
            - *SenderUserName*: pole loginu nadawcy
            - *SenderSystemName*: pole identyfikatora systemu nadawcy
            - *DateOfPosting*: pole daty nadania (strefa czasowa ZULU)
            - *ResponseMailboxAddress*: pole adresu odpowiedzi
    - *RecognizedDocumentsConfigurations*: lista konfiguracji rozpoznawanych typów pism
        - ***RecognitionConfiguration***: konfiguracja rozpoznawania pisma XML
            - *ValuePath*: ścieżka do elementu XML, po którym będzie identyfikowane pismo
            - ***Value***: wartość jakiej szukamy, aby rozpoznać pismo
        - *StyleTransformationPath*: ścieżka na dysku do pliku transformaty XSL do wygenerowania podglądu dokumentu, brak oznacza próbę pobrania pliku z crd.gov.pl
        - *XmlToFormFieldConfiguration*: lista konfiguracji mapowania wartości z XML na pola nAxiom
            - *FormField*: pole w nAxiom, gdzie zostanie zapisana wartość
            - ***XmlPath***: ścieżka do elementu XML, z której pobrać wartość

Przykład XML przesłanego od osoby fizycznej do podmiotu publicznego, w którym używane są dane z sekcji *StaticFieldsConfiguration*.

```
<danePodmiotu>
  <identyfikator>[redacted]-test</identyfikator>
  <typOsoby>FIZ</typOsoby>
  <imieSkrot>[redacted]</imieSkrot>
  <nazwiskoNazwa>Testowy</nazwiskoNazwa>
  <pesel></pesel>
  <zgoda>true</zgoda>
</danePodmiotu>
<daneNadawcy>
  <uzytkownik>[redacted]-test</uzytkownik>
</daneNadawcy>
<dataNadania>2022-02-01T10:47:15.53Z</dataNadania>
<nazwaSkrytki>Skrytka_ESP</nazwaSkrytki>
<adresSkrytki>/[redacted]TestPodPub/SkrytkaESP</adresSkrytki>
<adresOdpowiedzi>/[redacted]-test/domyslna</adresOdpowiedzi>
```

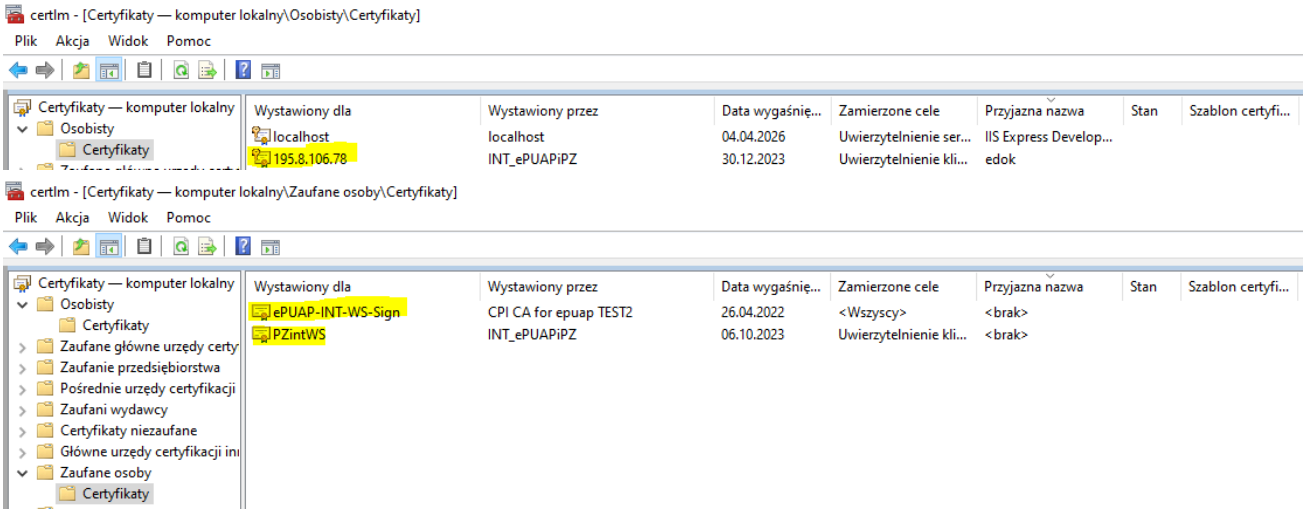
Przykład XML w komunikacji pomiędzy podmiotami publicznymi.

```
<danePodmiotu>
  <identyfikator>[redacted]TestPodPub</identyfikator>
  <typOsoby>PP</typOsoby>
  <nazwiskoNazwa>[redacted] SPÓŁKA AKCYJNA</nazwiskoNazwa>
  <nip>8130334531</nip>
  <regon>008033000</regon>
  <zgoda>true</zgoda>
</danePodmiotu>
<daneNadawcy>
  <system>kr375xmnen</system>
</daneNadawcy>
<dataNadania>2022-01-31T13:04:13.383Z</dataNadania>
<nazwaSkrytki>Skrytka_ESP</nazwaSkrytki>
<adresSkrytki>/[redacted]TestPodPub/SkrytkaESP</adresSkrytki>
<adresOdpowiedzi>/[redacted]TestPodPub/SkrytkaESP</adresOdpowiedzi>
```

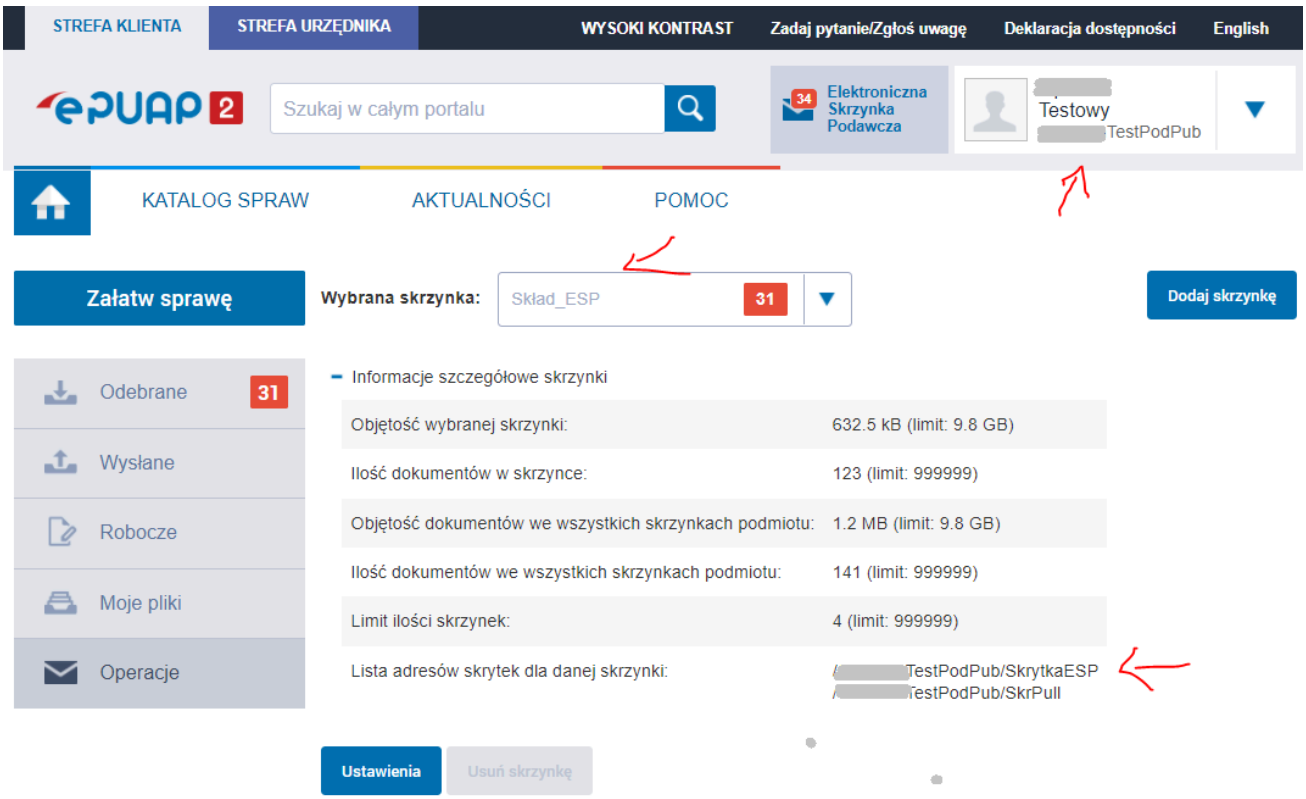

PRZYKŁAD PEŁNEJ KONFIGURACJI

Przykład pełnej konfiguracji zawiera Dodatek B.

Nazwy certyfikatów podane w konfiguracji pochodzą z certyfikatów zainstalowanych na bieżącej maszynie (*ClientCertificateName*, *EpuapServerCertificateName*, *PZServerCertificateName*).



Informacje na temat nazwy własnego podmiotu, nazwy skrzynki oraz adresu skrytki można znaleźć na stronie ePUAP po przejściu do Elektronicznej Skrzynki Podawczej swojej instytucji.



INSTALACJA I DEINSTALACJA USŁUGI EPUAP

Aby zainstalować usługę, przejdź do katalogu serwisu (*ścieżka_do_witryny_naxiom/podfolder_usługi_epuap*), uruchom terminal PowerShell jako administrator i uruchom polecenia podane poniżej. Pamiętaj, aby dla modułu AUTH w pliku appsettings.json dodać nowego klienta dla usługi ePUAP, który będzie korzystał z PublicAPI.

```
{
  "ClientId": "epuap.client",
  "ClientName": "",
  "AllowedGrantTypes": "ResourceOwnerPassword",
  "SecretKeySha256": "secret",
  "AllowedScopes": [ "public-api" ]
}
```

Polecenie instalacji:

```
.\NAX.Modules.ePUAP.Service.exe install -client "epuap.client" -secret "secret"
      -user "admin" -password "hasło użytkownika"
```

Polecenie wystartowania usługi:

```
.\NAX.Modules.ePUAP.Service.exe start
```

Polecenie zatrzymania usługi:

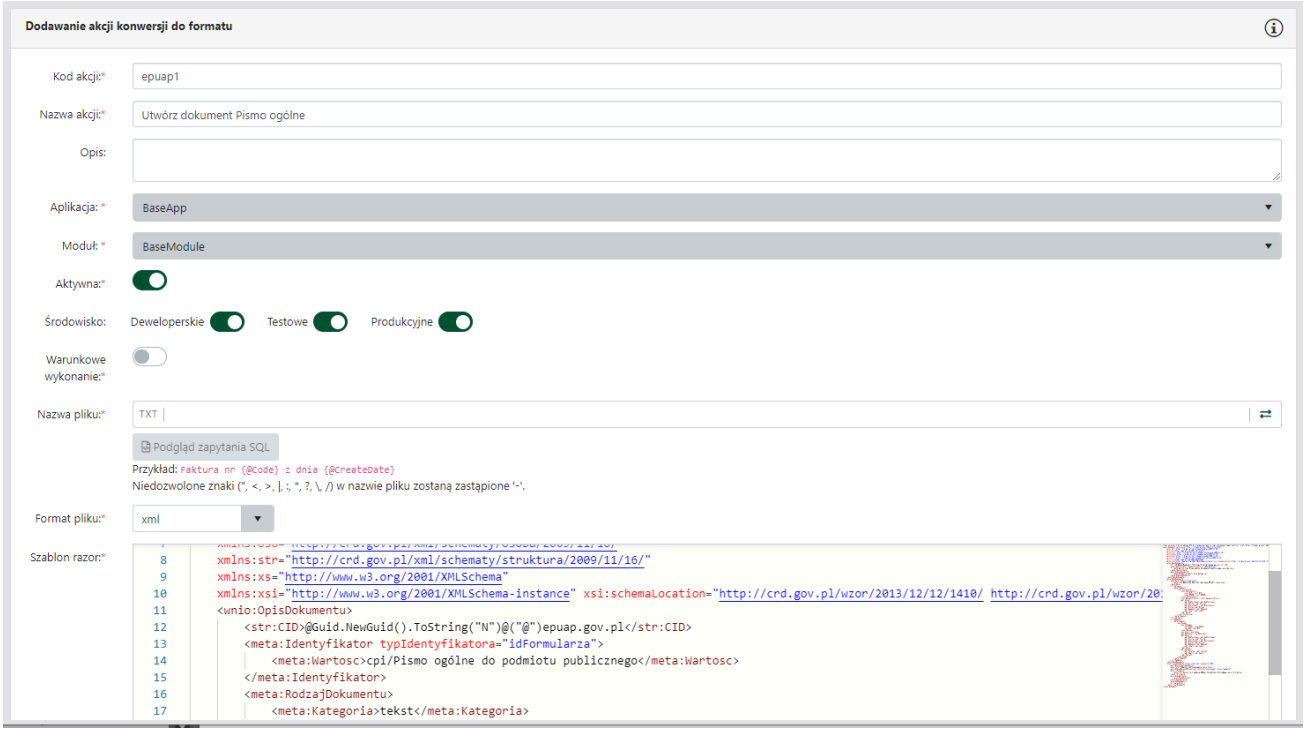
```
.\NAX.Modules.ePUAP.Service.exe stop
```

Odinstalowanie:

```
.\NAX.Modules.ePUAP.Service.exe uninstall
```

UTWORZENIE PISMA XML

W celu wygenerowania dokumentu w formacie XML na podstawie danych z formularza należy zdefiniować akcję `Tworzenie pliku tekstowego` i zdefiniować w niej szablon dokumentu ePUAP, używając języka Razor.



Wymagane parametry są określone przez wzór pisma. W Dodatku C zamieszczono przykłady następujących pism ogólnych:

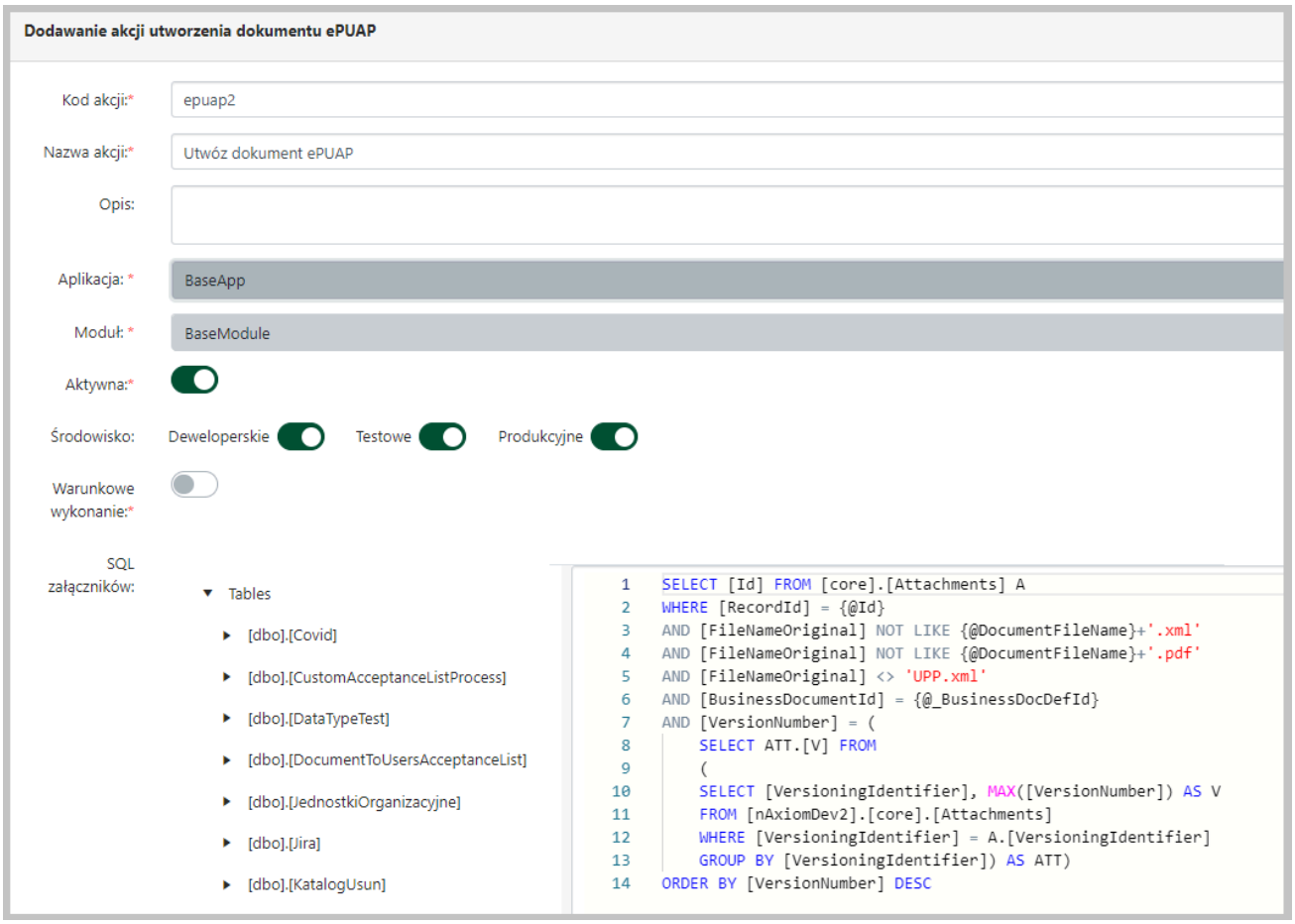
- od osoby fizycznej do podmiotu publicznego
- od podmiotu publicznego do osoby fizycznej
- od podmiotu publicznego do podmiotu publicznego

Pierwszy szablon utworzy pismo ogólne do podmiotu publicznego — nadawcą jest osoba fizyczna. Drugi szablon pozwala na utworzenie pisma ogólnego do osoby fizycznej — nadawcą jest podmiot publiczny. Trzeci szablon to pismo ogólne do podmiotu publicznego — nadawcą jest podmiot publiczny.

Te szablony różnią się w węzłach *<str:Adresaci>* i *<str:Nadawcy>*. W strukturze XML są to listy; węzeł *<meta:Podmiot>* może wewnątrz przyjmować definicję dla instytucji (*<inst:Instytucja>*) lub osoby fizycznej (*<oso:Osoba>*).

Aby umożliwić dodanie załączników do pisma należy w szablonie XML umieścić węzeł *<str:Zalaczniki/>* w sekcji *<wnio:TrescDokumentu format="text/xml" kodowanie="XML">*.

Następnie należy zdefiniować akcję `ePUAP utwórz dokument`. Aby móc dodawać załączniki do pisma, należy skonstruować zapytanie, które zwróci ID tych załączników. Zapytanie pomija pliki o tej samej nazwie co pismo, aby w wypadku ponownego generowania pisma do załączników nie został dodany już wygenerowany plik XML oraz podgląd w PDF. Brane pod uwagę są najnowsze wersje załączników.



Należy skonfigurować resztę parametrów akcji, czyli nazwę podmiotu, nazwę i adres skrytki oraz ID poprzednio utworzonej akcji generowania plików.

Nazwa podmiotu:* TXT | {@EntityName}

Podgląd zapytania SQL

Przykład: `Faktura nr {@Code} z dnia {@CreateDate}`

Nazwa skrytki:* TXT | {@MailboxName}

Podgląd zapytania SQL

Przykład: `Faktura nr {@Code} z dnia {@CreateDate}`

Adres skrytki:* TXT | {@MailboxAddress}

Podgląd zapytania SQL

Przykład: `Faktura nr {@Code} z dnia {@CreateDate}`

Akcja generowania pliku:* TXT | 28

Podgląd zapytania SQL

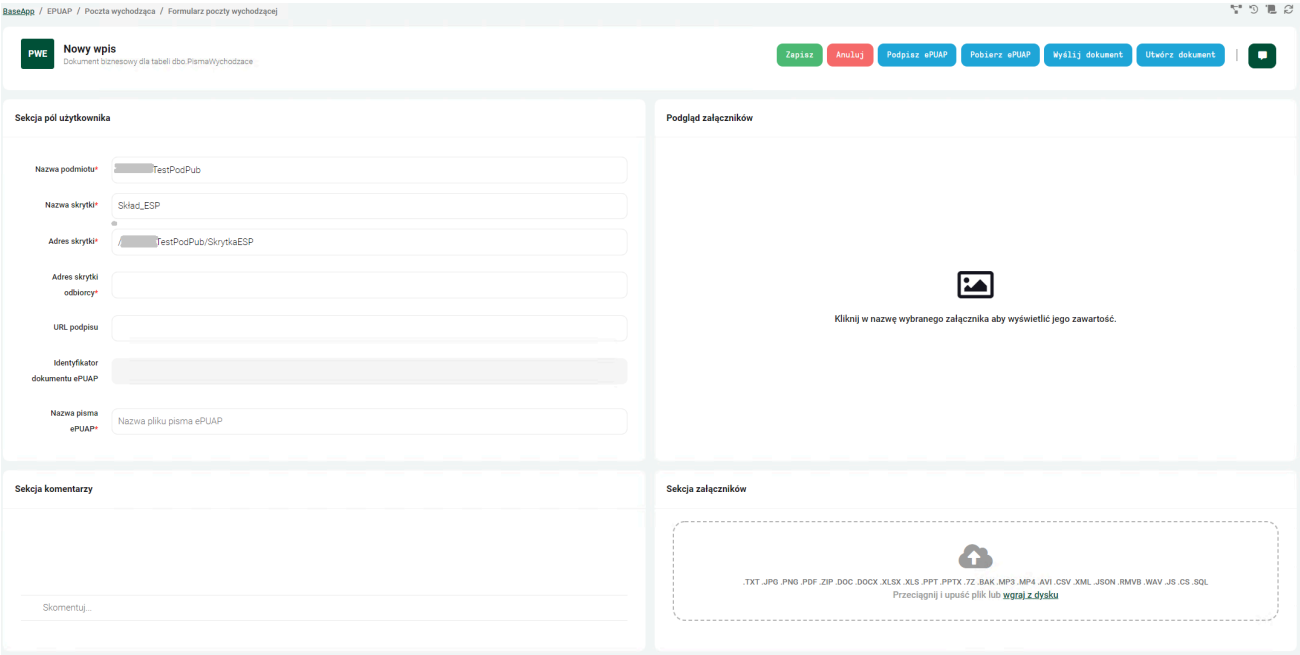
Przykład: `Faktura nr {@Code} z dnia {@CreateDate}`

Należy wskazać ID akcji "Tworzenie pliku tekstowego"

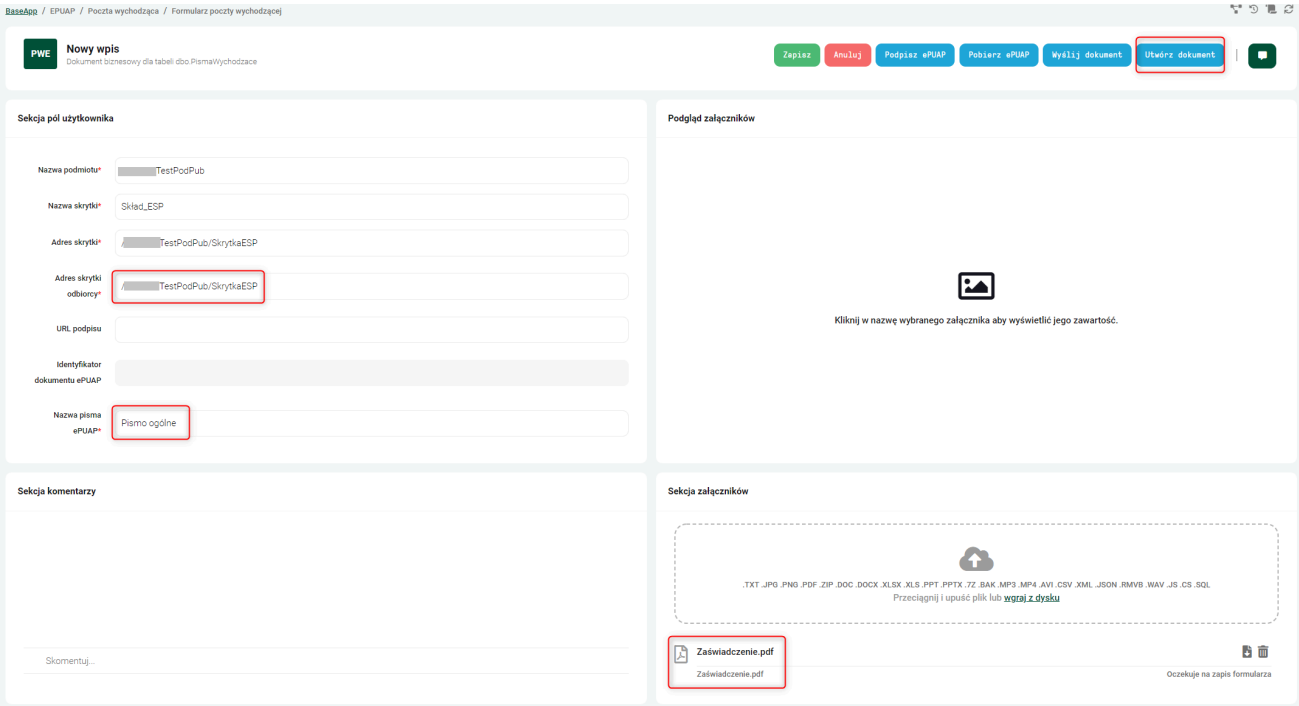
Miejsca użycia: Pola w formularzach:

Utwórz dokument [Formularz poczty wychodzącej] (PWE-form

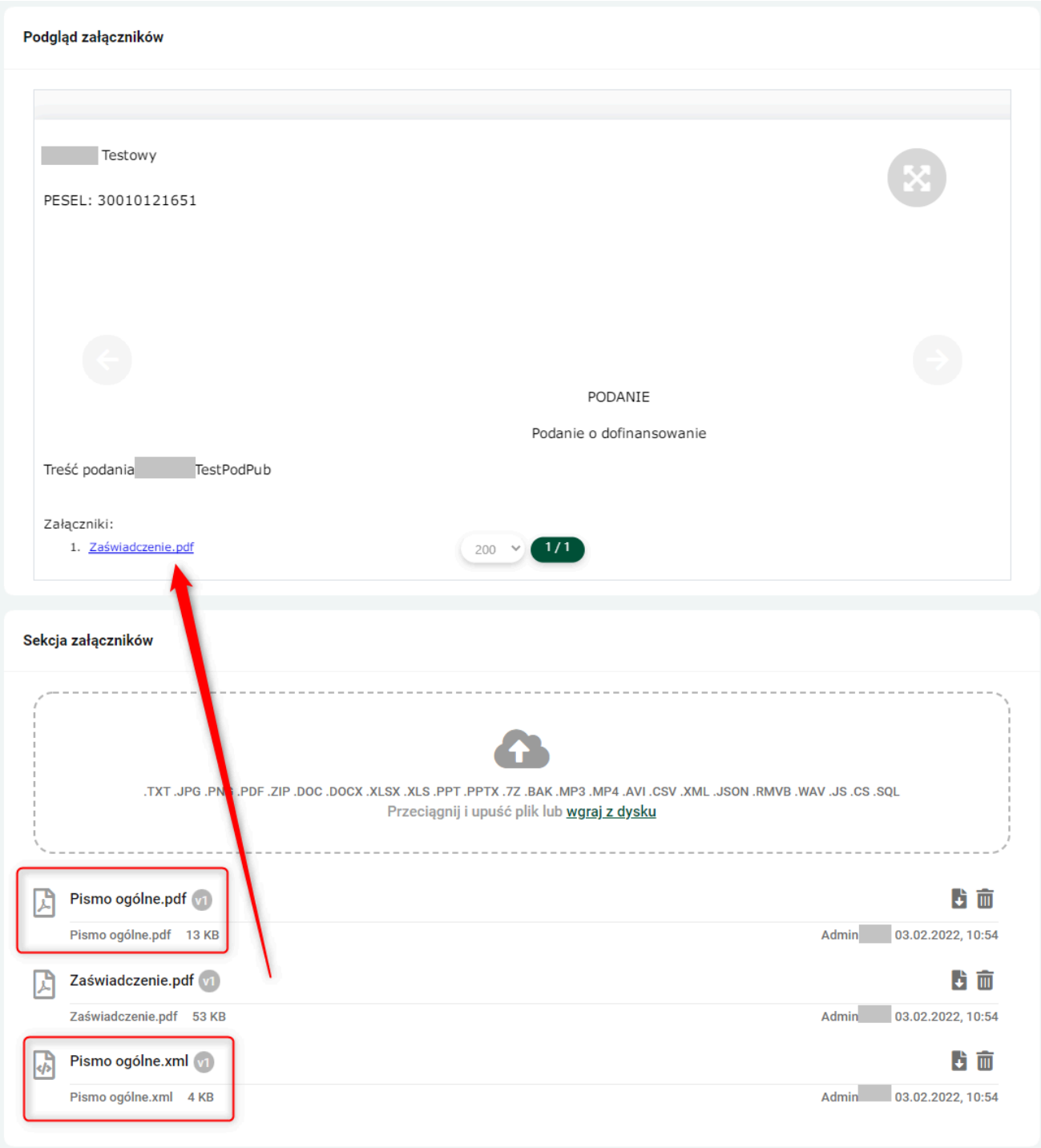
Przykładowy najprostszy formularz dla poczty wychodzącej może wyglądać jak poniżej; służy on tylko do wypełnienia wymaganych pól.



W ramach testu można utworzyć pismo, które zostanie wysłane na własną skrzynkę. Podaj własny adres jako adres odbiorcy oraz nazwę pliku XML. Można dodać plik załącznika, który zostanie wysłany wraz z pismem.



Uruchomienie akcji powoduje utworzenie dwóch załączników: pliku XML jako właściwego pisma wysyłanego do ePUAP oraz pliku PDF z podglądem tego pisma. Na podglądzie pisma jest informacja o dołączeniu do niego załącznika (plik *Zaświadczenie.pdf*).



PODPIS PISMA

Aby zautomatyzować proces obsługi podpisu, w pierwszym kroku należy zdefiniować akcję pobrania podpisanego pisma. Jest to akcja `ePUAP – pobierz podpisany dokument`.

Należy zdefiniować zapytanie SQL, które pobierze ID załącznika do pisma. Jest to wymagane, ponieważ pobranie podpisanego pisma nie zwraca nazwy pliku, co mogłoby spowodować problem z aktualizacją odpowiedniego pliku załącznika.

Dodawanie akcji pobrania podpisanego dokumentu ePUAP

Kod akcji:* epuap3

Nazwa akcji:* Pobierz podpisany dokument

Opis:

Aplikacja: * BaseApp

Moduł: * BaseModule

Aktywna:*

Środowisko: Deweloperskie Testowe Produkcyjne

Warunkowe wykonanie:*

SQL:

- Tables
  - [dbo].[Covid]
  - [dbo].[CustomAcceptanceListProcess]
  - [dbo].[DataTypeTest]
  - [dbo].[DocumentToUsersAcceptanceList]

```
1  SELECT [Id] FROM [core].[Attachments]
2  WHERE [RecordId] = {@RecordId}
3  AND [FileNameOriginal] LIKE {@DocumentFileName}+'%.xml'
4  AND [BusinessDocumentId] = {@BusinessDocumentId}
```

Następnie należy ustawić pozostałe standardowe parametry oraz przekazać URL podpisu.

Nazwa podmiotu:* TXT {@EntityName}

Podgląd zapytania SQL

Przykład: `Faktura nr {@Code} z dnia {@CreateDate}`

Nazwa skrytki:* TXT {@MailboxName}

Podgląd zapytania SQL

Przykład: `Faktura nr {@Code} z dnia {@CreateDate}`

Adres skrytki:* TXT {@MailboxAddress}

Podgląd zapytania SQL

Przykład: `Faktura nr {@Code} z dnia {@CreateDate}`

URL podpisu:* TXT {@SigningUrl}

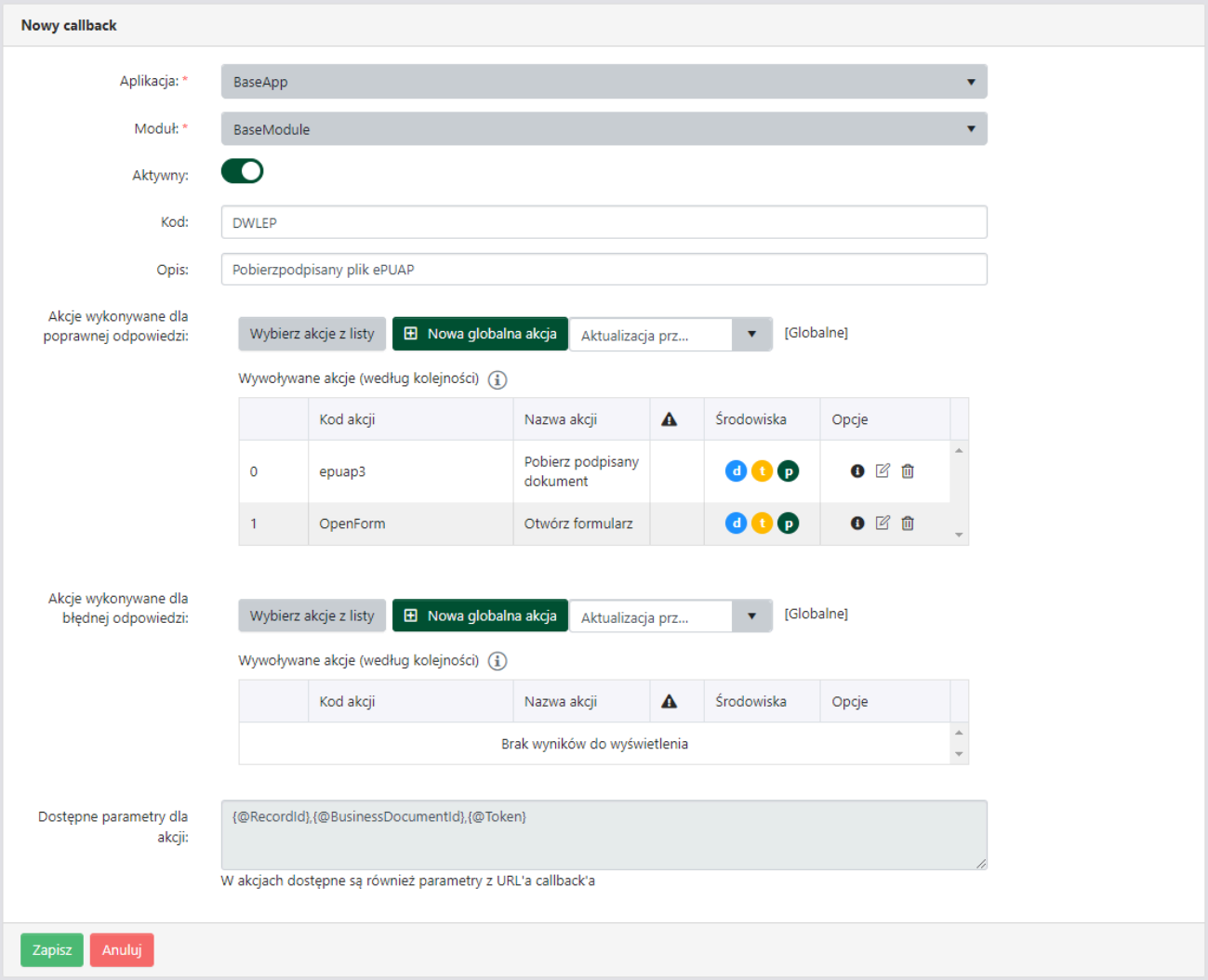
Podgląd zapytania SQL

Przykład: `Faktura nr {@Code} z dnia {@CreateDate}`

Miejsca użycia: Pola w formularzach:

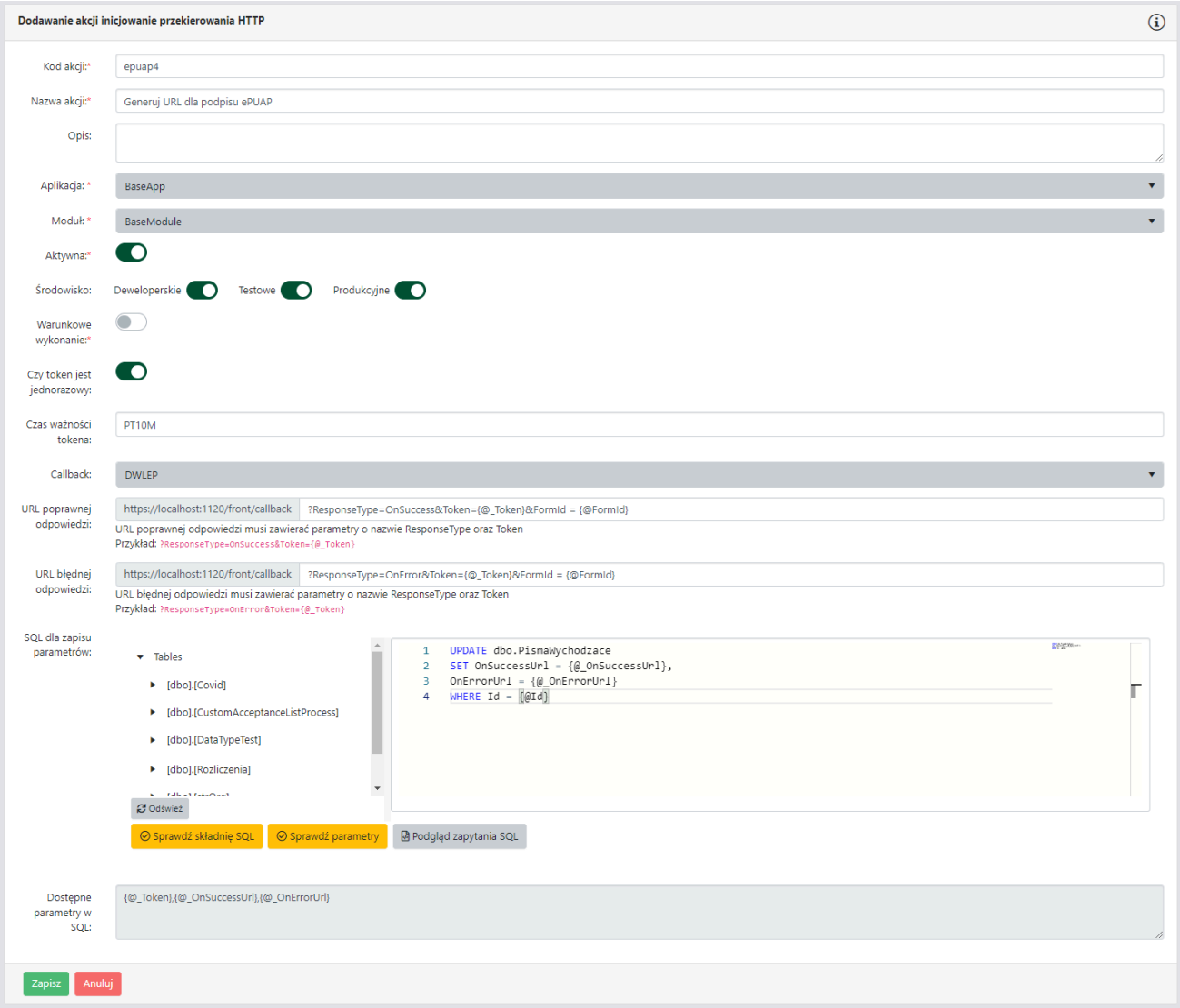
Pobierz ePUAP [Formularz poczty wychodzącej] (PWE-form

Następny krok to utworzenie obsługi *callback*, czyli przekierowania, wykonywanego przez usługę Profil Zaufany po podpisaniu pliku. W celu dodania odpowiedniej akcji, przejdź do menu `ADMINISTRACJA` > `Lista Callbacks` i dodaj akcję.



W akcjach poprawnej odpowiedzi dodaj akcje pobrania podpisanego dokumentu i otwarcia formularza, aby wrócić na formularz, z którego zainicjowano podpis.

Teraz utwórz akcję typu `Inicjowanie przekierowania`, która utworzy URL dla procedury *callback*.

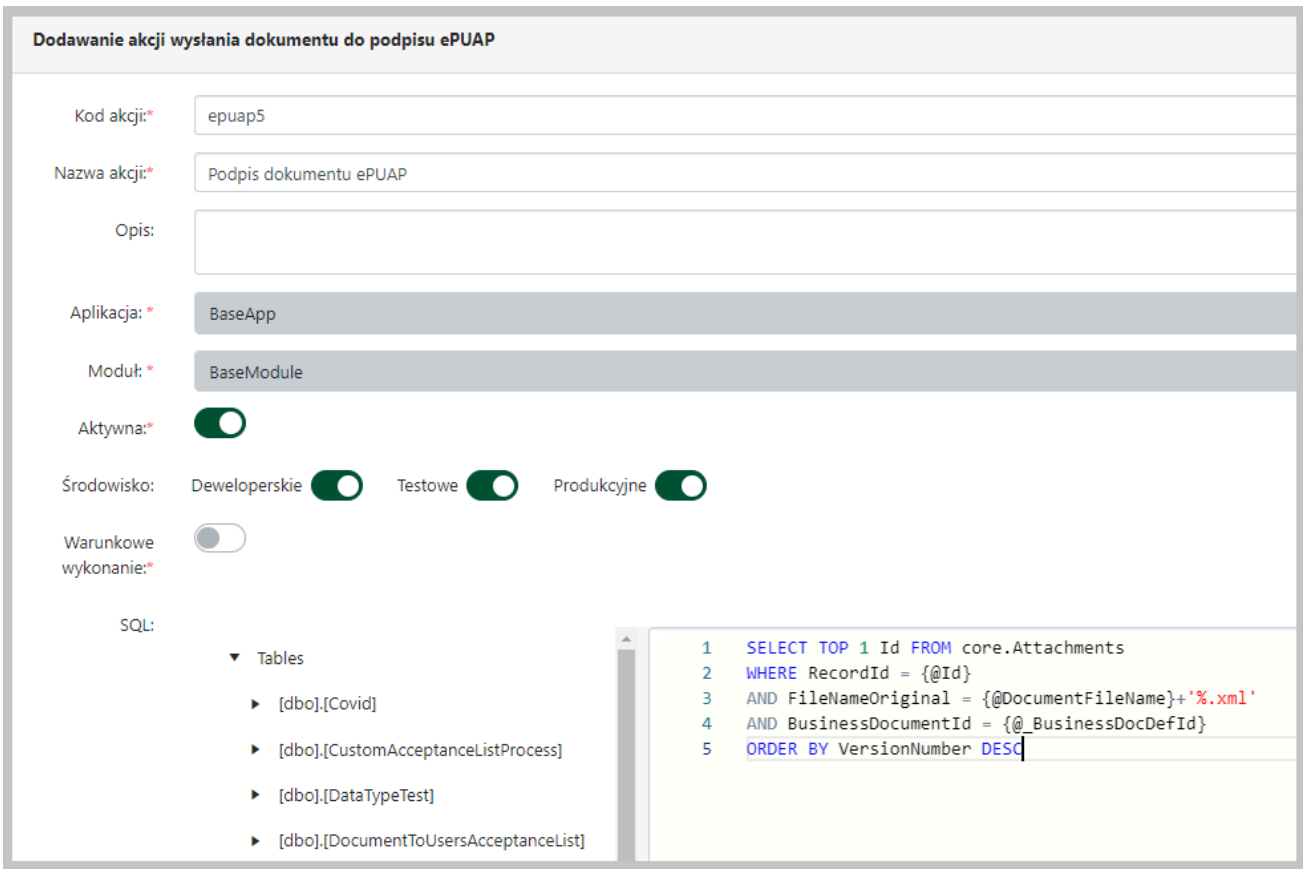


UWAGA

Pamiętaj, aby do standardowego URL dodać parametr *FormId* — *&FormId={@FormId}*. W przeciwnym razie zostanie zwrócony błąd podczas pobierania podpisanego dokumentu.

W SQL użyj instrukcji *UPDATE*, która zapisze na formularzu parametry *OnSuccessUrl* oraz *OnErrorUrl* w celu przekazania w akcji wysyłki dokumentu do podpisu.

Utwórz akcję `ePUAP podpisz dokument`; użyj w niej zapytania SQL zwracającego ID dokumentu wysyłanego do podpisu.



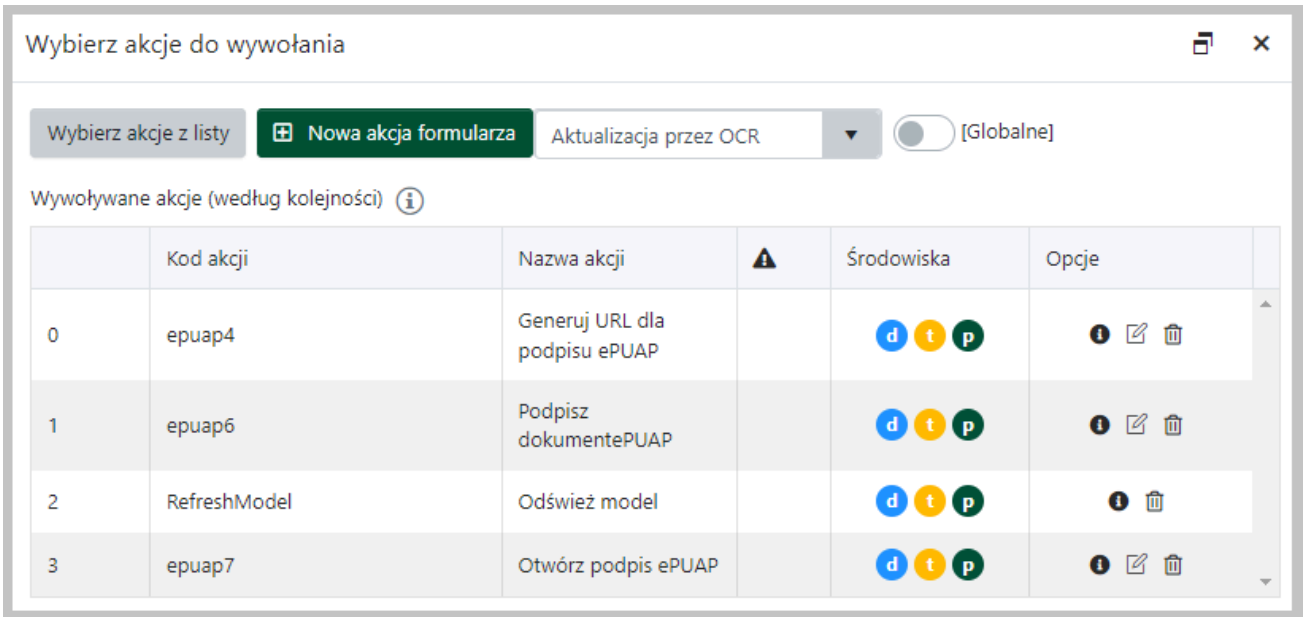
Uzupełnij pozostałe parametry konfiguracji.

Nazwa podmiotu:*	TXT | {@EntityName} Podgląd zapytania SQL Przykład: `Faktura nr {@Code} z dnia {@CreateDate}`
Nazwa skrytki:*	TXT | {@MailboxName} Podgląd zapytania SQL Przykład: `Faktura nr {@Code} z dnia {@CreateDate}`
Adres skrytki:*	TXT | {@MailboxAddress} Podgląd zapytania SQL Przykład: `Faktura nr {@Code} z dnia {@CreateDate}`
URL poprawnego podpisu:*	SQL | SELECT [OnSuccessUrl] FROM [dbo].[PismaWychodzace] WHERE [Id] = {@Id} Podgląd zapytania SQL Przykład: `SELECT CONCAT('Faktura nr ', {@Code}, ' z dnia ', {@CreateDate})`
URL błędu podpisu:*	SQL | SELECT [OnErrorUrl] FROM [dbo].[PismaWychodzace] WHERE [Id] = {@Id} Podgląd zapytania SQL Przykład: `SELECT CONCAT('Faktura nr ', {@Code}, ' z dnia ', {@CreateDate})`
URL anulowania podpisu:	TXT | Podgląd zapytania SQL Przykład: `Faktura nr {@Code} z dnia {@CreateDate}`
Pole na URL podpisu:*	SigningUrl
Dane dodatkowe:	TXT | Podgląd zapytania SQL Przykład: `Faktura nr {@Code} z dnia {@CreateDate}`
Miejsca użycia:	Przypisany formularz: Formularz poczty wychodzącej (PWE-form)

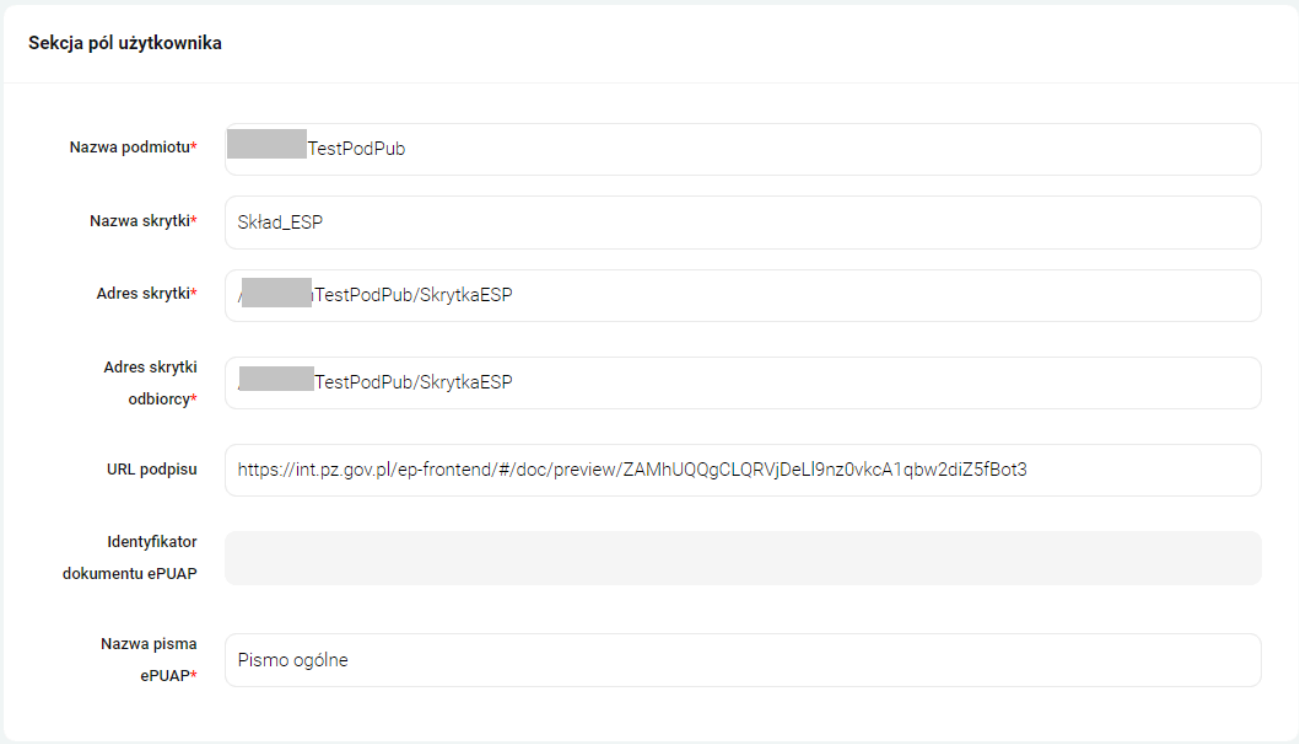
Parametry z adresami URL określają adresy przekierowania żądania przez usługę Profilu zaufanego w przypadku poprawnego podpisu, błędu lub anulowania żądania. Te parametry są ustawiane przez zapytania, ponieważ w ciągu akcji, które należy wykonać, nie można obecnie zastosować kilkukrotnie akcji `Odśwież model`.

W polu `Pole na URL podpisu` wskaż pole formularza do zapisu adresu URL, pod który użytkownik musi przejść, aby pismo podpisać. Po wygenerowaniu takiego URL użytkownik musi przejść pod ten adres w ciągu krótkiego czasu, który obecnie wynosi około 1 minuty.

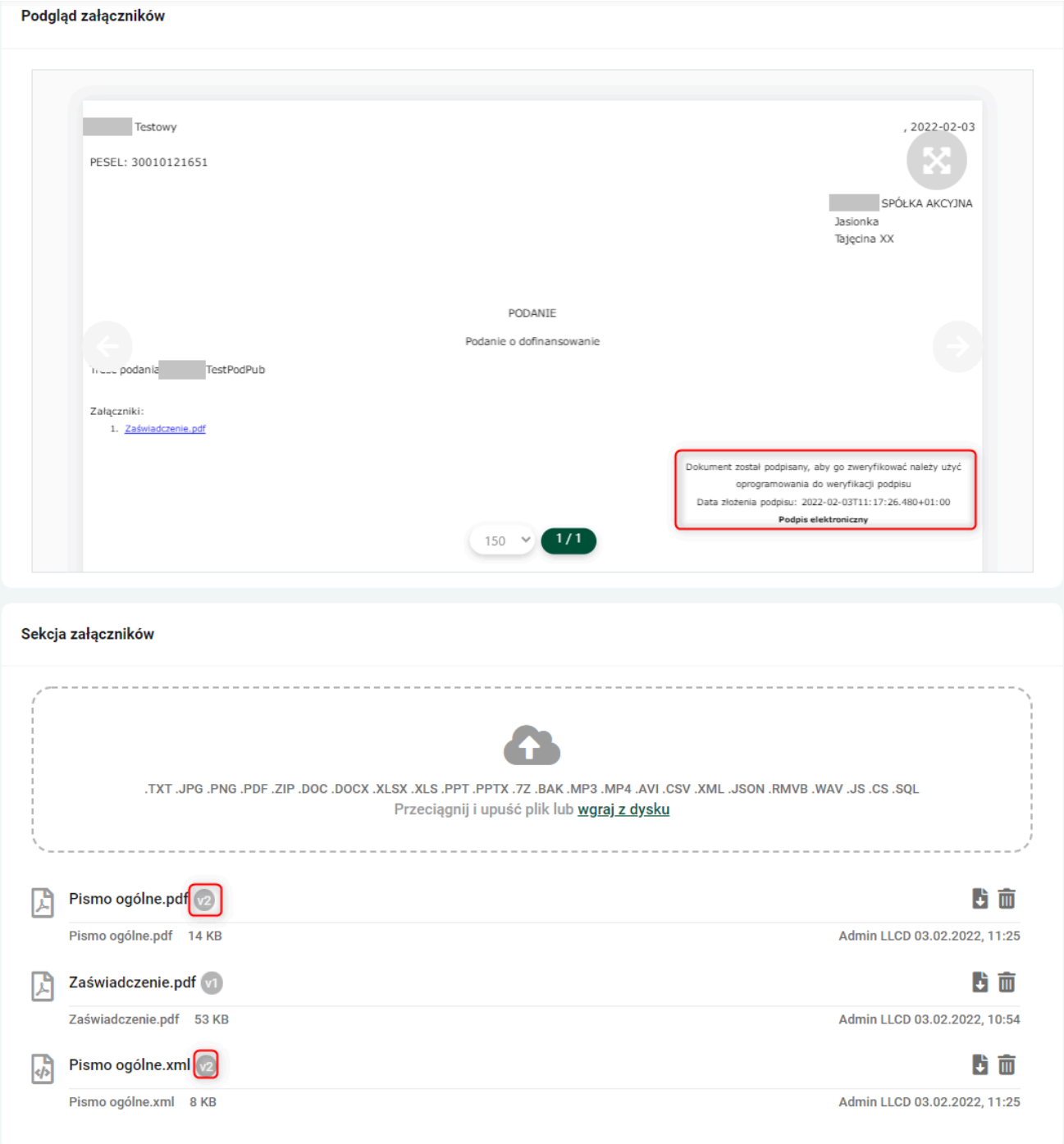
Po akcji wysyłki pisma w sekwencji akcji można dodać akcję `Odśwież formularz` i akcję `Otwórz link zewnętrzny` w celu przejścia pod wygenerowany adres URL.



Po podpisaniu pisma w usłudze Profil Zaufany następuje przekierowanie na podany adres *callback*, który powinien obsłużyć pobranie pisma oraz otworzyć ponownie formularz. Na formularzu jest widoczny adres URL, który służył do podpisu i pobrania pisma.

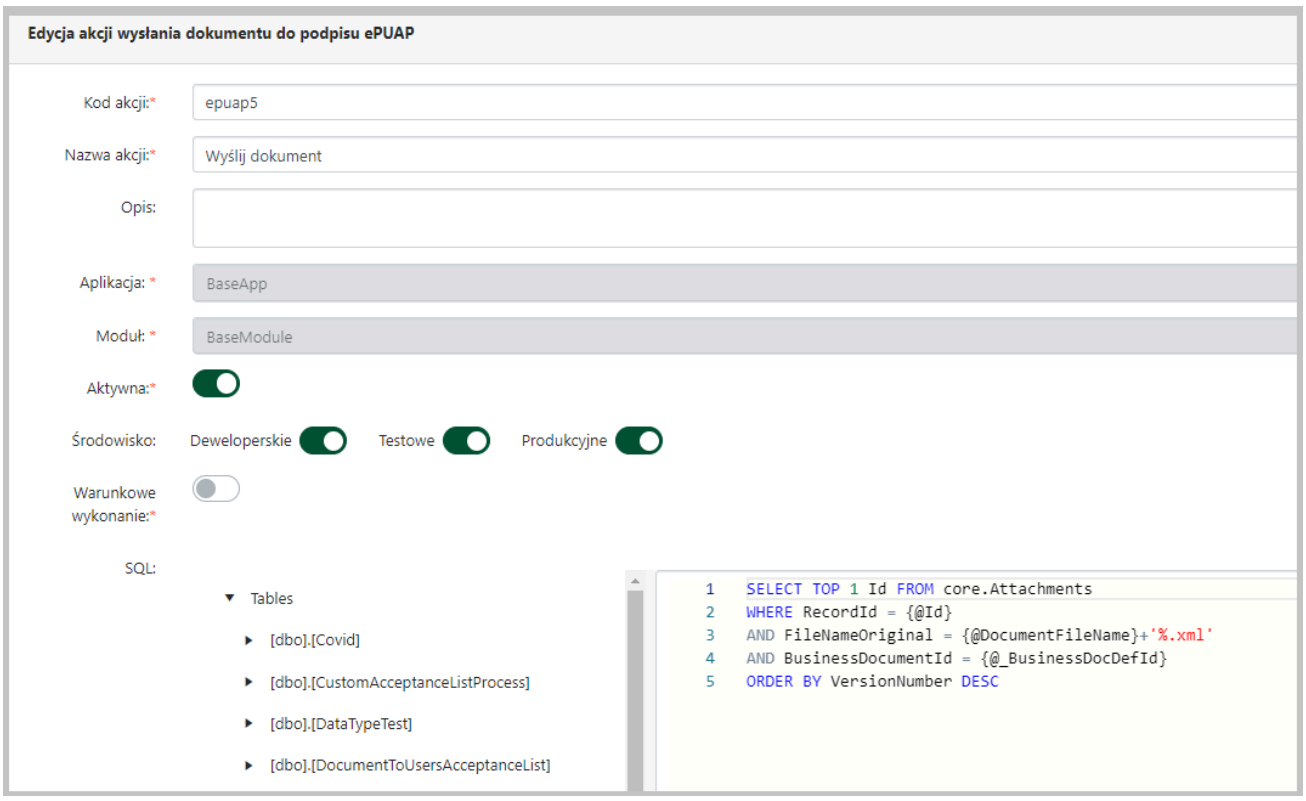


Po wykonaniu akcji w załącznikach pojawił się nowy plik XML oraz podgląd pisma.



WYSŁANIE PISMA

Aby wysłać pismo należy skonfigurować akcję wysyłki z zapytaniem SQL, które zwróci ID dokumentu XML do wysłania.



Pozostałe parametry to między innymi adres odbiorcy oraz nazwa pola, w którym zostanie zapisany ID dokumentu nadany przez ePUAP.

Nazwa podmiotu:* TXT {@EntityName}

Podgląd zapytania SQL

Przykład: `Faktura nr {@Code} z dnia {@CreateDate}`

Nazwa skrytki:* TXT {@MailboxName}

Podgląd zapytania SQL

Przykład: `Faktura nr {@Code} z dnia {@CreateDate}`

Adres skrytki:* TXT {@MailboxAddress}

Podgląd zapytania SQL

Przykład: `Faktura nr {@Code} z dnia {@CreateDate}`

Adres odbiorcy:* TXT {@ReceiverMailboxAddress}

Podgląd zapytania SQL

Przykład: `Faktura nr {@Code} z dnia {@CreateDate}`

Pole na ID dokumentu: DocumentId

Dane dodatkowe: TXT

Podgląd zapytania SQL

Przykład: `Faktura nr {@Code} z dnia {@CreateDate}`

POCZTA PRZYCHODZĄCA

Pobieraniem poczty przychodzącej zajmuje się usługa *NAX.Modules.ePUAP.Service*. Współpracuje ona ze skonfigurowaną skrytką w trybie *PULL*, co oznacza cykliczne pobieranie dokumentów w interwale nie częstszym niż 15 minut. Po odpytaniu kolejki o liczbę dokumentów w kolejce następuje ich indywidualne pobieranie i zapis w nAxiom z pakietem danych, według konfiguracji pobierania. Tylko w wypadku bezbłędnego dodania pisma w nAxiom następuje potwierdzenie odebrania pisma i pobranie następnego z kolejki. Błędy, które nie zatrzymają pobierania pisma to:

- Pobranie załączników z pisma — ePUAP potrafi zwrócić błąd podczas pobierania załączników (brak uprawnień do pliku), pobranie przez API się już nie powiedzie, jedyna możliwość pobrania to skorzystanie z URL do pliku, który znajduje się w XML lub na podglądzie pisma
- Generowanie podglądu pisma — w wypadku, gdy pismo nie zostało rozpoznane, podejmowana jest próba pobrania pliku XSL służącego do przekształcenia XML do HTML i później PDF, zdarzają się problemy z odpytaniem serwera crd.gov.pl.

NIEROZPOZNANE RODZAJE PISM

Za obsługę nierozpoznanych pism odpowiada sekcja *UnrecognizedDocumentConfiguration*.

```
"UnrecognizedDocumentConfiguration": {
  "SaveXmlAttachment": true,
  "StaticFieldsConfiguration": {
  },
  "BusinessDocumentConfiguration": {
    "DocumentDefinitionId": 3,
    "FormDefinitionId": 3,
    "AttachmentsCategoryId": 6,
    "XmlAttachmentCategoryId": 1,
    "XmlPreviewAttachmentCategoryId": 1,
    "SignaturePreviewAttachmentCategoryId": 1
  },
  "RequiredFieldsConfiguration": {
    "CaseId": "EpuapCaseId",
    "DocumentId": "EpuapDocumentId",
    "EntityName": "EntityName",
    "MailboxAddress": "MailboxAddress",
    "MailboxName": "MailboxName"
  },
  "SignatureValidationFieldsConfiguration": {
    "DocumentSignatureStatus": "StatusPodpisu"
  }
}
```

Wartości parametrów w powyższym przykładzie:

- `SaveXmlAttachment`: zapis pobranego dokumentu w formacie XML jak załącznik
- `DocumentDefinitionId`: definicja dokumentu biznesowego z ID 3
- `FormDefinitionId`: definicja formularza z ID 3
- `AttachmentsCategoryId`: załączniki z pisma zapisane w kategorii z ID 6
- `XmlAttachmentCategoryId`: kategoria załączników dla pliku XML pisma to ID 1
- `XmlPreviewAttachmentCategoryId`: podgląd pisma zapisany w kategorii z ID 1
- `SignaturePreviewAttachmentCategoryId`: podgląd podpisu pisma zapisany w kategorii z ID 1
- `CaseId`: zapisanie ID sprawy w polu formularza `EpuapCaseId`
- `DocumentId`: zapisanie ID dokumentu w polu formularza `EpuapDocumentId`

- `EntityName`: zapisanie nazwy podmiotu w polu formularza `EntityName`
- `MailboxAddress`: zapisanie adresu skrytki nadawcy w polu formularza `MailboxAddress`
- `MailboxName`: zapisanie nazwy skrytki nadawcy w polu formularza `MailboxName`
- `DocumentSignatureStatus`: zapisanie statusu podpisu w polu formularza `StatusPodpisu`

ROZPOZNANE RODZAJE PISM

Aby skonfigurować rozpoznawane typy pism należy dodać odpowiednią konfigurację do listy *RecognizedDocumentsConfigurations*. Przykład:

```
"RecognizedDocumentsConfigurations": [
  {
    "RecognitionConfiguration": {
      "ValuePath": "/Dokument/OpisDokumentu/Identyfikator/Wartosc",
      "Value": "cpi/Pismo ogólne do podmiotu publicznego"
    },
    "StyleTransformationPath":
       "C:\\TFS\\nAxiom\\core\\Modules\\ePUAP\\NAX.Modules.ePUAP.Service\\
                                                          bin\\Debug\\styl.xsl",
    "SaveXmlAttachment": true,
    "StaticFieldsConfiguration": {
    },
    "BusinessDocumentConfiguration": {
      "DocumentDefinitionId": 3,
      "FormDefinitionId": 3,
      "AttachmentsCategoryId": 6,
      "XmlAttachmentCategoryId": 1,
      "XmlPreviewAttachmentCategoryId": 1,
      "SignaturePreviewAttachmentCategoryId": 1
    },
    "RequiredFieldsConfiguration": {
      "CaseId": "EpuapCaseId",
      "DocumentId": "EpuapDocumentId",
      "EntityName": "EntityName",
      "MailboxAddress": "MailboxAddress",
      "MailboxName": "MailboxName"
    },
    "SignatureValidationFieldsConfiguration": {
```

```
        "DocumentSignatureStatus": "StatusPodpisu"
      },
      "XmlToFormFieldConfiguration": [
        {
          "FormField": "Tytul",
          "XmlPath": "/Dokument/TrescDokumentu/Tytul"
        },
        {
          "FormField": "Informacja",
          "XmlPath": "/Dokument/TrescDokumentu/Informacje/Informacja"
        },
        {
          "FormField": "RodzajPisma",
          "XmlPath": "/Dokument/TrescDokumentu/RodzajWnioskuRozszerzony@rodzaj"
        }
      ]
    }
  ]
```

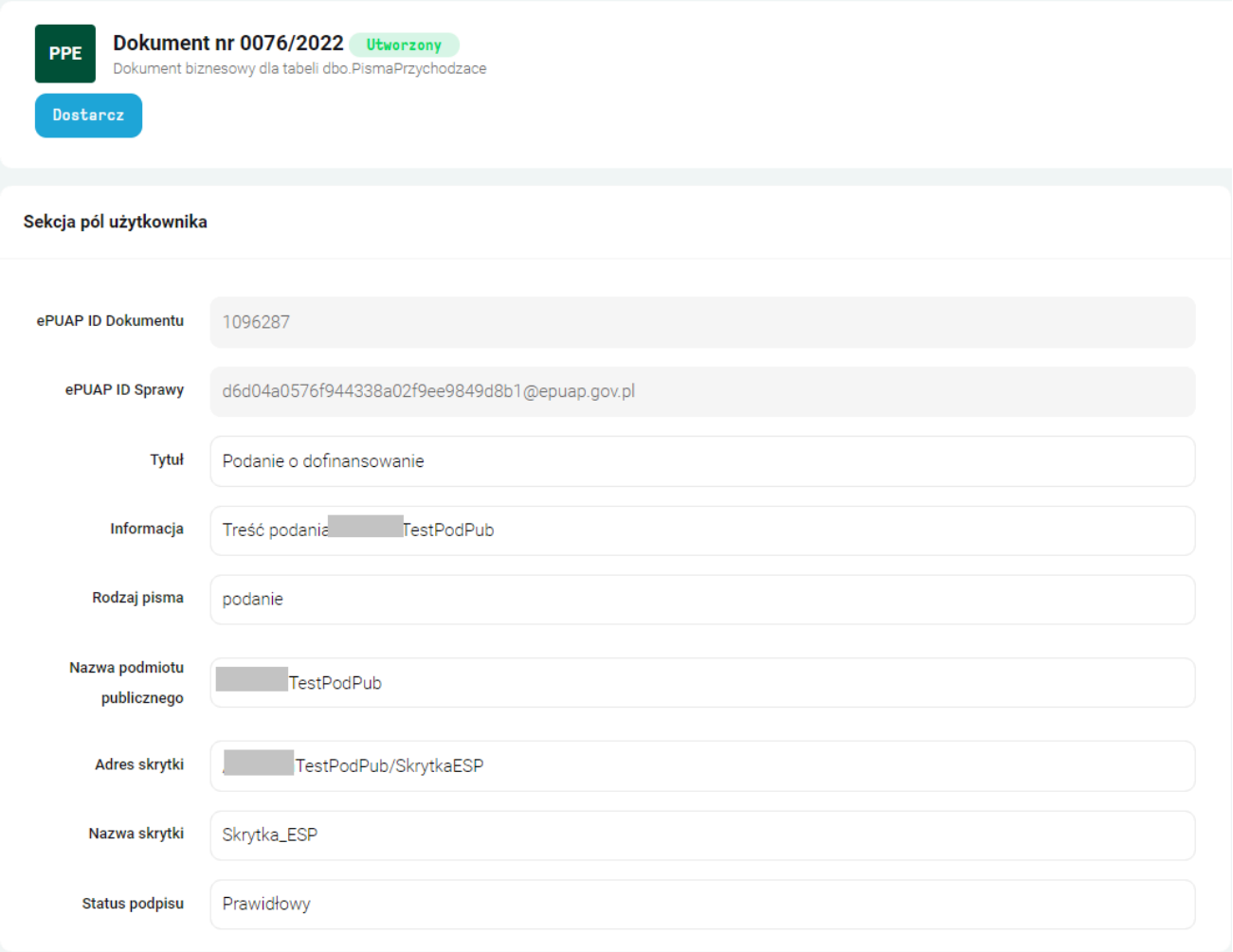
Wartości parametrów w powyższym przykładzie:

- `ValuePath`: ścieżka do wartości w pliku XML, która określa rodzaj pisma
- `Value`: poszukiwana wartość
- `StyleTransformationPath`: ścieżka do pliku XSL transformującego XML do HTML, jeśli plik nie zostanie odnaleziony to nastąpi próba wygenerowania podglądu na podstawie adresu z pliku XML i crd.gov.pl
- `SaveXmlAttachment`: zapisz pismo XML do załączników
- `DocumentDefinitionId`: definicja dokumentu biznesowego z ID 3
- `FormDefinitionId`: definicja formularza z ID 3
- `AttachmentsCategoryId`: załączniki z pisma zapisane w kategorii z ID 6
- `XmlAttachmentCategoryId`: kategoria załączników dla pliku XML pisma to ID 1
- `XmlPreviewAttachmentCategoryId`: podgląd pisma zapisany w kategorii z ID 1
- `SignaturePreviewAttachmentCategoryId`: podgląd podpisu pisma zapisany w kategorii z ID 1
- `CaseId`: zapisanie ID sprawy w polu formularza `EpuapCaseId`
- `DocumentId`: zapisanie ID dokumentu w polu formularza `EpuapDocumentId`
- `EntityName`: zapisanie nazwy podmiotu w polu formularza `EntityName`
- `MailboxAddress`: zapisanie adresu skrytki nadawcy w polu formularza

`MailboxAddress`

- `MailboxName`: zapisanie nazwy skrytki nadawcy w polu formularza `MailboxName`
- `DocumentSignatureStatus`: zapisane statusu podpisu w polu formularza `StatusPodpisu`
- `XmlToFormFieldConfiguration`: mapowanie pól z pisma XML na pola w nAxiom; do pola `Tytul` trafi wartość z węzła `/Dokument/TrescDokumentu/Tytul`, podobnie `Informacja`, `RodzajPisma` znajduje się w pliku XML w atrybucie węzła dlatego zastosowano zapis *@rodzaj*

Dane pobrane z pisma, tj. ID pisma, ID sprawy, tytuł, informacja, rodzaj pisma, dane skrytki nadawcy, status podpisu.



Podgląd pisma XML, na którym widać te dane:

[redacted] Testowy , 2022-02-08

PESEL: 30010121651

[redacted] SPÓŁKA AKCYJNA
Jasionka
Tajęcina XX

PODANIE

Podanie o dofinansowanie

Treść podania [redacted] TestPodPub

Załączniki:
1. Zaświadczenie.pdf

Dokument został podpisany, aby go zweryfikować należy użyć oprogramowania do weryfikacji podpisu
Data złożenia podpisu: 2022-02-08T10:22:54.621+01:00
Podpis elektroniczny

Załączniki, w tym pismo XML, jego podgląd, UPP i podgląd podpisu.

Sekcja załączników | Załączniki

.TXT .JPG .PNG .PDF .ZIP .DOC .DOCX .XLSX .XLS .PPT .PPTX .7Z .BAK .MP3 .MP4 .AVI .CSV .XML .JSON .RMVB .WAV .JS .CS .SQL
Przeciągnij i upuść plik lub wgraj z dysku

Pismo.pdf v1	Pismo.pdf 14 KB	system 08.02.2022, 10:23
UPP.xml v1	UPP.xml 7 KB	system 08.02.2022, 10:23
Pismo - Signature.pdf v1	Pismo - Signature.pdf 20 KB	system 08.02.2022, 10:23
Pismo.xml v1	Pismo.xml 8 KB	system 08.02.2022, 10:23

Podgląd podpisu:

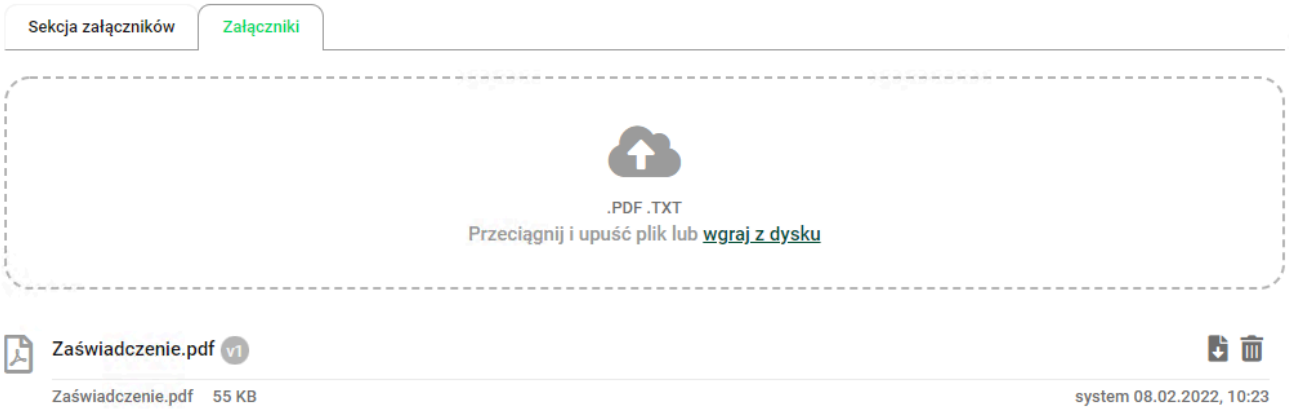
Status podpisu dokumentu: Prawidłowy

Data weryfikacji: 2022-02-08T10:23:16.748+01:00

Podpis Prawidłowy - [redacted], Testowy

Stan podpisu:	Prawidłowy
Stan certyfikatu:	Ważny
Nr seryjny certyfikatu:	717539058
Wystawca certyfikatu (CN):	CPI CA for epuap TEST2
Właściciel certyfikatu (CN):	epodpis_sign_int
Właściciel certyfikatu (O):	COI
Szczegóły:	• kwalifikowany
Identyfikator podpisu:	Signature-b623ffe8-fee9-40c8-9d99-7fa1d4cd1934
Czas podpisania:	2022-02-08 10:22:54 CET
Typ zobowiązania:	
Typ podpisu:	podpis zaufany
Dane Profilu Zaufanego:	[redacted] Testowy (30010121651)

W przypadku załączników do pisma skonfigurowano zapis w innej kategorii.



Podgląd pliku:

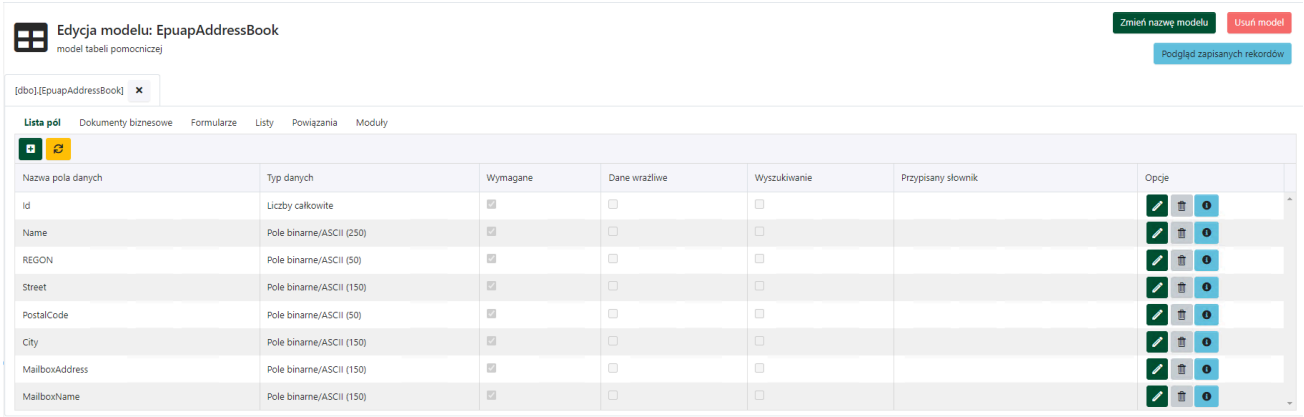
Zaświadczenie

Zaświadczenie, ABCD

KSIĄŻKA ADRESOWA EPUAP

Usługa pozwala także na skonfigurowanie cyklicznego pobierania aktualnej książki adresowej. Dostępne są jedynie adresy podmiotów publicznych, które wystąpiły o umieszczenie ich w książce adresowej.

Najpierw utwórz tabelę zwykłą dla książki adresowej.



Przykładowa konfiguracja dla środowiska testowego dla powyższej tabeli:

```
"AddressesConfiguration": {
  "RepeatInterval": 86400000,
  "Url": "https://int.epuap.gov.pl/LESP/LESP.csv",
  "TableSchema": "dbo",
  "TableName": "EpuapAddressBook",
  "KeyColumn": "Id",
  "NameColumn": "Name",
  "RegonColumn": "REGON",
  "StreetColumn": "Street",
  "PostalCodeColumn": "PostalCode",
  "CityColumn": "City",
```

```
    "MailboxAddressColumn": "MailboxAddress",
    "MailboxNameColumn": "MailboxName"
  }
```

Tabela jest czyszczona i uzupełniana nowymi danymi. Powodem czyszczenia jest brak parametru, który pozwala na porównywanie nowych i starych wartości w określonym czasie. Adresy skrytek powtarzają się, posiadają niejednokrotnie różnice jedynie w wielkości liter, co wymusza wyszukiwanie w bazie danych z uwzględnieniem tego faktu. Ze względu na duży wolumen danych wymagane jest wyszukiwanie tekstowe, co źle wpływa na wydajność.

DODATEK B. PRZYKŁAD PEŁNEJ KONFIGURACJI

```
{
  "AuthConfiguration": {
    "ClientId": "",
    "ClientSecret": "",
    "User": "",
    "Password": ""
  },
  "AuthUrl": "",
  "ServiceUrl": "",
  "PublicApiUrl": "",
  "LogResponseXml": true,
  "LogsPath": "C:\\Logs\\ePUAP",
  "AllowedIPs": "127.0.0.1;::1",
  "AddressesConfiguration": {
    "RepeatInterval": 86400000,
    "Url": "https://int.epuap.gov.pl/LESP/LESP.csv",
    "TableSchema": "",
    "TableName": "",
    "KeyColumn": "",
    "NameColumn": "",
    "RegonColumn": "",
    "StreetColumn": "",
    "PostalCodeColumn": "",
    "CityColumn": "",
    "MailboxAddressColumn": "",
    "MailboxNameColumn": ""
  },
  "IntegrationConfigurations": [
    {
```

```
    "Active": true,
    "ValidateSignature": true,
    "RepeatInterval": 900000,
    "EpuapDnsIdentity": "ePUAP-INT-WS-Sign",
    "PZDnsIdentity": "PZintWS",
    "ClientCertificateName": "",
    "EpuapServerCertificateName": "ePUAP-INT-WS-Sign",
    "PZServerCertificateName": "PZintWS",
    "EntityName": "",
    "MailboxName": "",
    "MailboxAddress": "",
    "EpuapPullServiceUrl":
              "https://int.epuap.gov.pl/pk_external_ws/services/pull",
    "EpuapPOBoxServiceUrl":
              "https://int.epuap.gov.pl/pk_external_ws/services/skrytka",
    "TpSigningServiceUrl":
              "https://int.pz.gov.pl/ep-services/tpSigning5",
    "FileRepoServiceUrl":
              "https://int.epuap.gov.pl/repo-ws-ext/FileRepoService",
    "RecognizedDocumentsConfigurations": [
      {
        "RecognitionConfiguration": {
          "ValuePath": "/Dokument/OpisDokumentu/Identyfikator/Wartosc",
          "Value": "cpi/Pismo ogólne do podmiotu publicznego"
        },
        "StyleTransformationPath": "",
        "SaveXmlAttachment": true,
        "StaticFieldsConfiguration": {
            "SubjectId": "",
            "PersonType": "",
            "FirstName": "",
            "LastName": "",
            "Pesel": "",
            "Nip": "",
            "Regon": "",
            "SenderUserName": "",
            "SenderSystemName": "",
            "DateOfPosting": "",
            "ResponseMailboxAddress": ""
        },
        "BusinessDocumentConfiguration": {
          "DocumentDefinitionId": 0,
```

```
          "FormDefinitionId": 0,
          "AttachmentsCategoryId": 0,
          "XmlAttachmentCategoryId": 0,
          "XmlPreviewAttachmentCategoryId": 0,
                      "SignaturePreviewAttachmentCategoryId": 0
        },
        "RequiredFieldsConfiguration": {
          "CaseId": "EpuapCaseId",
          "DocumentId": "EpuapDocumentId",
          "EntityName": "EntityName",
          "MailboxAddress": "MailboxAddress",
          "MailboxName": "MailboxName"
        },
        "SignatureValidationFieldsConfiguration": {
          "DocumentSignatureStatus": "StatusPodpisu"
        },
        "XmlToFormFieldConfiguration": [
          {
            "FormField": "Tytul",
            "XmlPath": "/Dokument/TrescDokumentu/Tytul"
          },
          {
            "FormField": "Informacja",
            "XmlPath": "/Dokument/TrescDokumentu/Informacje/Informacja"
          },
          {
            "FormField": "RodzajPisma",
            "XmlPath": "/Dokument/TrescDokumentu/RodzajWnioskuRozszerzony@rodzaj"
          }
        ]
      }
    ],
    "UnrecognizedDocumentConfiguration": {
      "SaveXmlAttachment": true,
      "StaticFieldsConfiguration": {
          "SubjectId": "",
          "PersonType": "",
          "FirstName": "",
          "LastName": "",
          "Pesel": "",
          "Nip": "",
          "Regon": "",
```

```
            "SenderUserName": "",
            "SenderSystemName": "",
            "DateOfPosting": "",
            "ResponseMailboxAddress": ""
        },
        "BusinessDocumentConfiguration": {
          "DocumentDefinitionId": 0,
          "FormDefinitionId": 0,
          "AttachmentsCategoryId": 0,
          "XmlAttachmentCategoryId": 0,
          "XmlPreviewAttachmentCategoryId": 0,
                  "SignaturePreviewAttachmentCategoryId": 0
        },
        "RequiredFieldsConfiguration": {
          "CaseId": "EpuapCaseId",
          "DocumentId": "EpuapDocumentId",
          "EntityName": "EntityName",
          "MailboxAddress": "MailboxAddress",
          "MailboxName": "MailboxName"
        },
        "SignatureValidationFieldsConfiguration": {
          "DocumentSignatureStatus": "StatusPodpisu"
        }
      }
    }
  ]
}
```


DODATEK C. PRZYKŁADOWE PISMA EPUAP


```
{
  "AuthConfiguration": {
    "ClientId": "aqE2NV1PExhVqdpVnrQKPg==",
    "ClientSecret": "veEzPBSsXgrmcRhztdbWUA==",
    "User": "In+1M/aYqS89fEoaoM9Y3A==",
    "Password": "2CBgr27KQE2Xbc7WDdcyig=="
  },
  "AuthUrl": "http://localhost:5000",
  "ServiceUrl": "http://localhost:8080",
  "PublicApiUrl": "https://localhost:5203",
```

```
  "LogResponseXml": true,
  "LogsPath": "C:\\Logs\\ePUAP",
  "AllowedIPs": "127.0.0.1;::1",
  "AddressesConfiguration": {
    "RepeatInterval": 86400000,
    "Url": "https://int.epuap.gov.pl/LESP/LESP.csv",
    "TableSchema": "dbo",
    "TableName": "EpuapAddressBook",
    "KeyColumn": "Id",
    "NameColumn": "Name",
    "RegonColumn": "REGON",
    "StreetColumn": "Street",
    "PostalCodeColumn": "PostalCode",
    "CityColumn": "City",
    "MailboxAddressColumn": "MailboxAddress",
    "MailboxNameColumn": "MailboxName"
  },
  "IntegrationConfigurations": [
    {
      "Active": true,
      "ValidateSignature": true,
      "RepeatInterval": 900000,
      "EpuapDnsIdentity": "ePUAP-INT-WS-Sign",
      "PZDnsIdentity": "PZintWS",
      "ClientCertificateName": "195.8.106.78",
      "EpuapServerCertificateName": "ePUAP-INT-WS-Sign",
      "PZServerCertificateName": "PZintWS",
      "EntityName": "ABCdefTestPodPub",
      "MailboxName": "Skład_ESP",
      "MailboxAddress": "/ABCdefTestPodPub/SkrytkaESP",
      "EpuapPullServiceUrl": "https://int.epuap.gov.pl/pk_external_ws/services/pull",
      "EpuapPOBoxServiceUrl": "https://int.epuap.gov.pl/pk_external_ws/services/skrytka",
      "TpSigningServiceUrl": "https://int.pz.gov.pl/ep-services/tpSigning5",
      "FileRepoServiceUrl": "https://int.epuap.gov.pl/repo-ws-ext/FileRepoService",
      "RecognizedDocumentsConfigurations": [
        {
          "RecognitionConfiguration": {
            "ValuePath": "/Dokument/OpisDokumentu/Identyfikator/Wartosc",
            "Value": "cpi/Pismo ogólne do podmiotu publicznego"
          },
          "StyleTransformationPath":
              "C:\\TFS\\nAxiom\\core\\Modules\\ePUAP\\NAX.Modules.ePUAP.Service\\
```

```
                                                                bin\\Debug\\styl.xsl",
      "SaveXmlAttachment": true,
      "StaticFieldsConfiguration": {
      },
      "BusinessDocumentConfiguration": {
        "DocumentDefinitionId": 3,
        "FormDefinitionId": 3,
        "AttachmentsCategoryId": 6,
        "XmlAttachmentCategoryId": 1,
        "XmlPreviewAttachmentCategoryId": 1,
        "SignaturePreviewAttachmentCategoryId": 1
      },
      "RequiredFieldsConfiguration": {
        "CaseId": "EpuapCaseId",
        "DocumentId": "EpuapDocumentId",
        "EntityName": "EntityName",
        "MailboxAddress": "MailboxAddress",
        "MailboxName": "MailboxName"
      },
      "SignatureValidationFieldsConfiguration": {
        "DocumentSignatureStatus": "StatusPodpisu"
      },
      "XmlToFormFieldConfiguration": [
        {
          "FormField": "Tytul",
          "XmlPath": "/Dokument/TrescDokumentu/Tytul"
        },
        {
          "FormField": "Informacja",
          "XmlPath": "/Dokument/TrescDokumentu/Informacje/Informacja"
        },
        {
          "FormField": "RodzajPisma",
          "XmlPath": "/Dokument/TrescDokumentu/RodzajWnioskuRozszerzony@rodzaj"
        }
      ]
    }
  ],
  "UnrecognizedDocumentConfiguration": {
    "SaveXmlAttachment": true,
    "StaticFieldsConfiguration": {
    },
```

```
        "BusinessDocumentConfiguration": {
          "DocumentDefinitionId": 3,
          "FormDefinitionId": 3,
          "AttachmentsCategoryId": 6,
          "XmlAttachmentCategoryId": 1,
          "XmlPreviewAttachmentCategoryId": 1,
          "SignaturePreviewAttachmentCategoryId": 1
        },
        "RequiredFieldsConfiguration": {
          "CaseId": "EpuapCaseId",
          "DocumentId": "EpuapDocumentId",
          "EntityName": "EntityName",
          "MailboxAddress": "MailboxAddress",
          "MailboxName": "MailboxName"
        },
        "SignatureValidationFieldsConfiguration": {
          "DocumentSignatureStatus": "StatusPodpisu"
        }
      }
    }
  ]
}
```

PISMO OGÓLNE OD OSOBY FIZYCZNEJ DO PODMIOTU

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<?xml-stylesheet type="text/xsl"
  href="http://crd.gov.pl/wzor/2013/12/12/1410/styl.xsl"?>
<wnio:Dokument xmlns:wnio="http://crd.gov.pl/wzor/2013/12/12/1410/"
    xmlns:adr="http://crd.gov.pl/xml/schematy/adres/2009/11/09/"
    xmlns:ev="http://www.w3.org/2001/xml-events"
    xmlns:inst="http://crd.gov.pl/xml/schematy/instytucja/2009/11/16/"
    xmlns:meta="http://crd.gov.pl/xml/schematy/meta/2009/11/16/"
    xmlns:oso="http://crd.gov.pl/xml/schematy/osoba/2009/11/16/"
    xmlns:str="http://crd.gov.pl/xml/schematy/struktura/2009/11/16/"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:schemaLocation="http://crd.gov.pl/wzor/2013/12/12/1410/
                              http://crd.gov.pl/wzor/2013/12/12/1410/schemat.xsd">
    <wnio:OpisDokumentu>
```

```
        <str:CID>@Guid.NewGuid().ToString("N")@("@")epuap.gov.pl</str:CID>
        <meta:Identyfikator typIdentyfikatora="idFormularza">
            <meta:Wartosc>cpi/Pismo ogólne do podmiotu publicznego</meta:Wartosc>
        </meta:Identyfikator>
        <meta:RodzajDokumentu>
            <meta:Kategoria>tekst</meta:Kategoria>
        </meta:RodzajDokumentu>
    </wnio:OpisDokumentu>
    <wnio:DaneDokumentu>
        <meta:Data typDaty="stworzony">
            <meta:Czas>@DateTime.Now.ToString("yyyy-MM-dd")</meta:Czas>
        </meta:Data>
        <str:Adresaci>
            <meta:Podmiot>
                <inst:Instytucja>
                    <inst:IdInstytucji>
                        <inst:NIP></inst:NIP>
                        <inst:REGON></inst:REGON>
                    </inst:IdInstytucji>
                    <inst:NazwaInstytucji></inst:NazwaInstytucji>
                    <adr:Adres>
                        <adr:KodPocztowy></adr:KodPocztowy>
                        <adr:Poczta></adr:Poczta>
                        <adr:Miejscowosc></adr:Miejscowosc>
                        <adr:Ulica></adr:Ulica>
                        <adr:Budynek></adr:Budynek>
                        <adr:Lokal></adr:Lokal>
                    </adr:Adres>
                </inst:Instytucja>
            </meta:Podmiot>
        </str:Adresaci>
        <str:Nadawcy>
            <meta:Podmiot>
                <oso:Osoba>
                    <oso:IdOsoby>
                        <oso:PESEL></oso:PESEL>
                        <oso:NIP></oso:NIP>
                    </oso:IdOsoby>
                    <oso:Imie></oso:Imie>
                    <oso:Nazwisko></oso:Nazwisko>
                    <adr:Adres>
                        <adr:KodPocztowy></adr:KodPocztowy>
```

```
                        <adr:Poczta></adr:Poczta>
                        <adr:Miejscowosc></adr:Miejscowosc>
                        <adr:Ulica></adr:Ulica>
                        <adr:Budynek></adr:Budynek>
                        <adr:Lokal></adr:Lokal>
                    </adr:Adres>
                    <adr:Kontakt>
                        <adr:Telefon></adr:Telefon>
                        <adr:Email></adr:Email>
                    </adr:Kontakt>
                </oso:Osoba>
            </meta:Podmiot>
        </str:Nadawcy>
    </wnio:DaneDokumentu>
    <wnio:TrescDokumentu format="text/xml" kodowanie="XML">
        <wnio:MiejscowoscDokumentu/>
        <wnio:Tytul>Podanie o dofinansowanie</wnio:Tytul>
        <wnio:RodzajWnioskuRozszerzony jakiInny="inne pismo" rodzaj="podanie"/>
        <wnio:Informacje>
            <wnio:Informacja>Treść podania @Model.FormContext.EntityName</wnio:Informacja>
        </wnio:Informacje>
        <wnio:Oswiadczenia>
            <wnio:Oswiadczenie/>
        </wnio:Oswiadczenia>
        <wnio:DoWiadomosci/>
        <str:Zalaczniki/>
    </wnio:TrescDokumentu>
</wnio:Dokument>
```

PISMO OGÓLNE OD PODMIOTU DO OSOBY FIZYCZNEJ

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<?xml-stylesheet type="text/xsl" href="http://crd.gov.pl/wzor/2013/12/12/1410/styl.xsl"?>
<wnio:Dokument xmlns:wnio="http://crd.gov.pl/wzor/2013/12/12/1410/"
               xmlns:adr="http://crd.gov.pl/xml/schematy/adres/2009/11/09/"
              xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
              xmlns:ev="http://www.w3.org/2001/xml-events"
              xmlns:inst="http://crd.gov.pl/xml/schematy/instytucja/2009/11/16/"
              xmlns:meta="http://crd.gov.pl/xml/schematy/meta/2009/11/16/"
              xmlns:oso="http://crd.gov.pl/xml/schematy/osoba/2009/11/16/"
```

```
            xmlns:str="http://crd.gov.pl/xml/schematy/struktura/2009/11/16/"
            xmlns:xades="http://uri.etsi.org/01903/v1.3.2#"
            xmlns:xs="http://www.w3.org/2001/XMLSchema"
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
            xsi:schemaLocation="http://crd.gov.pl/wzor/2013/12/12/1410/
                                http://crd.gov.pl/wzor/2013/12/12/1410/schemat.xsd">
    <wnio:OpisDokumentu>
        <str:CID>@Guid.NewGuid().ToString("N")@("@")epuap.gov.pl</str:CID>
        <meta:Identyfikator typIdentyfikatora="idFormularza">
            <meta:Wartosc>cpi/Pismo ogólne do podmiotu publicznego</meta:Wartosc>
        </meta:Identyfikator>
    </wnio:OpisDokumentu>
    <wnio:DaneDokumentu>
        <meta:Data typDaty="stworzony">
            <meta:Czas>@DateTime.Now.ToString("yyyy-MM-dd")</meta:Czas>
        </meta:Data>
        <str:Adresaci>
            <meta:Podmiot>
                <oso:Osoba>
                                    <oso:IdOsoby>
                        <oso:PESEL></oso:PESEL>
                        <oso:NIP></oso:NIP>
                    </oso:IdOsoby>
                    <oso:Imie></oso:Imie>
                    <oso:Nazwisko></oso:Nazwisko>
                    <adr:Adres>
                        <adr:KodPocztowy></adr:KodPocztowy>
                        <adr:Poczta></adr:Poczta>
                        <adr:Miejscowosc></adr:Miejscowosc>
                        <adr:Ulica></adr:Ulica>
                        <adr:Budynek></adr:Budynek>
                        <adr:Lokal></adr:Lokal>
                    </adr:Adres>
                    <adr:Kontakt>
                        <adr:Telefon></adr:Telefon>
                        <adr:Email></adr:Email>
                    </adr:Kontakt>
                                </oso:Osoba>
            </meta:Podmiot>
        </str:Adresaci>
        <str:Nadawcy>
            <meta:Podmiot>
```

```
                <inst:Instytucja>
                    <inst:IdInstytucji>
                        <inst:NIP></inst:NIP>
                        <inst:REGON></inst:REGON>
                    </inst:IdInstytucji>
                    <inst:NazwaInstytucji></inst:NazwaInstytucji>
                    <adr:Adres>
                        <adr:KodPocztowy></adr:KodPocztowy>
                        <adr:Poczta></adr:Poczta>
                        <adr:Miejscowosc></adr:Miejscowosc>
                        <adr:Ulica></adr:Ulica>
                        <adr:Budynek></adr:Budynek>
                        <adr:Lokal></adr:Lokal>
                    </adr:Adres>
                </inst:Instytucja>
            </meta:Podmiot>
        </str:Nadawcy>
    </wnio:DaneDokumentu>
    <wnio:TrescDokumentu format="text/xml" kodowanie="XML">
        <wnio:MiejscowoscDokumentu></wnio:MiejscowoscDokumentu>
        <wnio:Tytul></wnio:Tytul>
        <wnio:RodzajWnioskuRozszerzony jakiInny="" rodzaj="wniosek"/>
        <wnio:Informacje>
            <wnio:Informacja></wnio:Informacja>
        </wnio:Informacje>
        <str:Zalaczniki/>
    </wnio:TrescDokumentu>
</wnio:Dokument>
```

PISMO OGÓLNE OD PODMIOTU DO PODMIOTU

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?> <?xml-stylesheet type="text/xsl" href="http://crd.go
    <wnio:OpisDokumentu>
        <str:CID>@Guid.NewGuid().ToString("N")@("@")epuap.gov.pl</str:CID>
        <meta:Identyfikator typIdentyfikatora="idFormularza">
            <meta:Wartosc>cpi/Pismo ogólne do podmiotu publicznego</meta:Wartosc>
        </meta:Identyfikator>
    </wnio:OpisDokumentu>
    <wnio:DaneDokumentu>
        <meta:Data typDaty="stworzony">
```

```
            <meta:Czas>@DateTime.Now.ToString("yyyy-MM-dd")</meta:Czas>
        </meta:Data>
        <str:Adresaci>
            <meta:Podmiot>
                <inst:Instytucja>
                                        <inst:IdInstytucji>
                        <inst:NIP></inst:NIP>
                        <inst:REGON></inst:REGON>
                    </inst:IdInstytucji>
                    <inst:NazwaInstytucji></inst:NazwaInstytucji>
                    <adr:Adres>
                        <adr:KodPocztowy></adr:KodPocztowy>
                        <adr:Poczta></adr:Poczta>
                        <adr:Miejscowosc></adr:Miejscowosc>
                        <adr:Ulica></adr:Ulica>
                        <adr:Budynek></adr:Budynek>
                        <adr:Lokal></adr:Lokal>
                    </adr:Adres>
                </inst:Instytucja>
            </meta:Podmiot>
        </str:Adresaci>
        <str:Nadawcy>
            <meta:Podmiot>
                <inst:Instytucja>
                    <inst:IdInstytucji>
                        <inst:NIP></inst:NIP>
                        <inst:REGON></inst:REGON>
                    </inst:IdInstytucji>
                    <inst:NazwaInstytucji></inst:NazwaInstytucji>
                    <adr:Adres>
                        <adr:KodPocztowy></adr:KodPocztowy>
                        <adr:Poczta></adr:Poczta>
                        <adr:Miejscowosc></adr:Miejscowosc>
                        <adr:Ulica></adr:Ulica>
                        <adr:Budynek></adr:Budynek>
                        <adr:Lokal></adr:Lokal>
                    </adr:Adres>
                </inst:Instytucja>
            </meta:Podmiot>
        </str:Nadawcy>
    </wnio:DaneDokumentu>
    <wnio:TrescDokumentu format="text/xml" kodowanie="XML">
```

```
        <wnio:MiejscowoscDokumentu></wnio:MiejscowoscDokumentu>
        <wnio:Tytul></wnio:Tytul>
        <wnio:RodzajWnioskuRozszerzony jakiInny="" rodzaj="wniosek"/>
        <wnio:Informacje>
            <wnio:Informacja></wnio:Informacja>
        </wnio:Informacje>
        <str:Zalaczniki/>
    </wnio:TrescDokumentu>
</wnio:Dokument>
```

9.21. DEFINIOWANIE POMOCY W APLIKACJACH BIZNESOWYCH

Podsumowanie:
W aplikacjach biznesowych nAxiom można wyświetlać pomoc dotyczącą formularzy, list, definicji dokumentów biznesowych oraz statusów. Treść pomocy jest wyświetlana w zdefiniowanym przez konsultanta formularzu, kiedy użytkownik kliknie ikonę koła ratunkowego . Zależnie od ustawień, ikona pomocy jest wyświetlana w obszarze ikon systemowych (np. kontekst, audyt, odśwież) na formularzu i w widoku listy (także w sekcji typu lista).

Przykładowy scenariusz konfigurowania pomocy dla użytkownika może wyglądać następująco:

1. Dodać treść pomocy do elementów aplikacji.
2. Zdefiniować formularz do wyświetlania pomocy.
3. Zdefiniować akcje otwierania formularza z przekazywaniem identyfikatora formularza/statusu i listy.
4. Przypisać akcje wykonywane po kliknięciu przycisku pomocy:
    - w ustawieniach systemowych
    - indywidualnie dla dokumentu biznesowego
5. Skonfigurować opcje wyświetlania przycisku pomocy dla list i formularzy.

Poszczególne kroki tego scenariusza opisano w osobnych tematach.

Powiązane tematy:

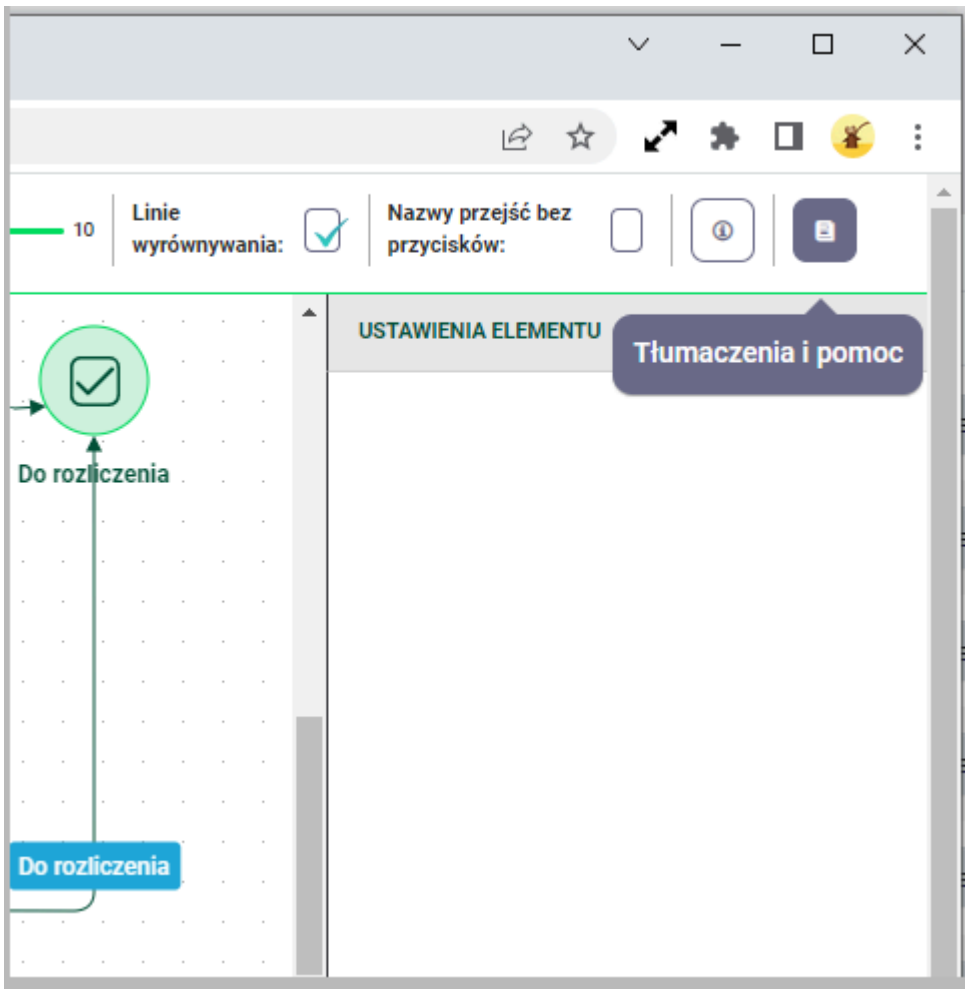
- Ustawienia systemu, sekcja Pomoc i dokumentacja
- Definicja dokumentu, karta Pomoc dla użytkownika
- Właściwości listy, karta Pomoc dla użytkownika
- Właściwości formularza, karta Pomoc dla użytkownika

9.21.1. Dodawanie treści pomocy

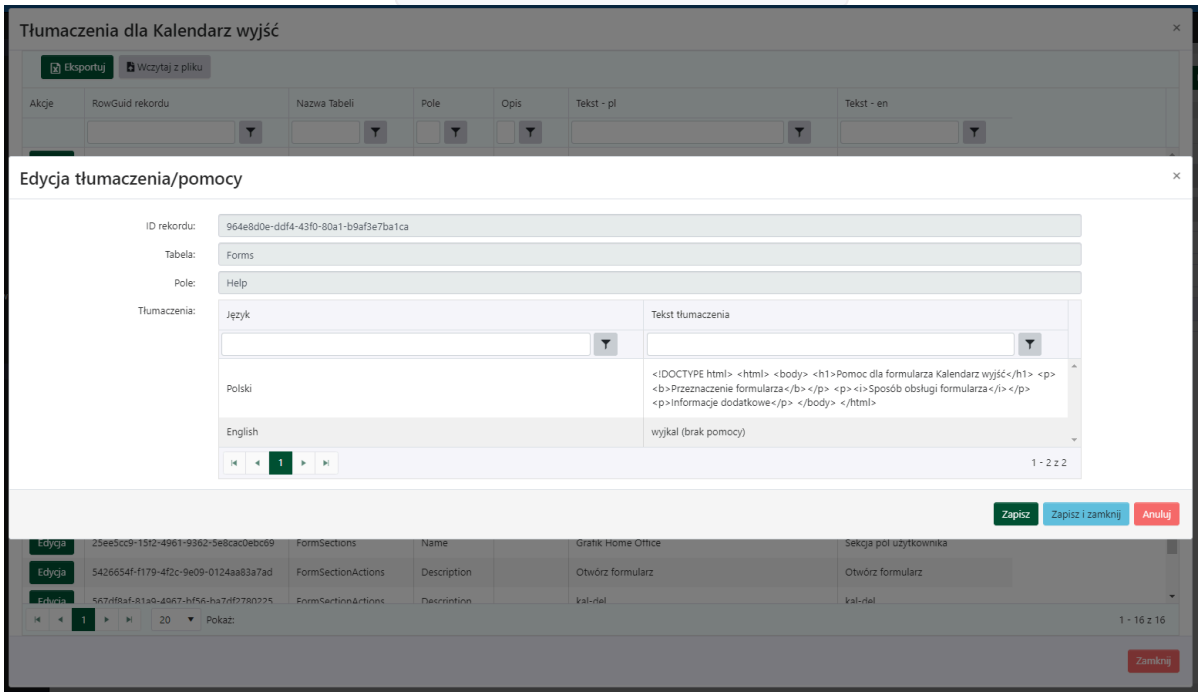
Podsumowanie:
Treść pomocy dla aplikacji biznesowych dodaje się, używając funkcjonalności `Tłumaczenia i pomoc`.

1. Wyświetl definicję listy, formularza, dokumentu biznesowego lub diagram procesu.
2. Kliknij przycisk `Tłumaczenia i pomoc`. Zostanie wyświetlona tabela z tekstami tłumaczeń i pomocy.

   WSKAZÓWKA
   Na diagramie procesu przycisk `Tłumaczenia i pomoc` jest reprezentowany przez ostatnią ikonę z prawej strony na pasku narzędzi u góry.



3. Kliknij przycisk `Edycja` w wierszu, w którym wartość w kolumnie `Pole` to `Help`. Zostanie wyświetlone okno `Edycja tłumaczenia/pomocy`.



4. Wpisz tekst pomocy w kolumnie `Tekst tłumaczenia` dla języka polskiego; w tym polu można wkleić przygotowany osobno kod HTML z tekstem pomocy

sformatowanym przy użyciu znaczników HTML; opcjonalnie podaj treść pomocy dla innych języków.

> **WSKAZÓWKA**
> Treść pomocy można także wprowadzić bezpośrednio w odpowiedniej komórce tabeli tłumaczeń; aby zmiana została zapisana, należy nacisnąć przycisk *Enter*.

Treść pomocy można również dodawać, używając polecenia `Tłumaczenia i pomoc` z menu `ADMINISTRACJA`. To polecenie wyświetla całą tabelę tłumaczeń, aby łatwiej znaleźć odpowiednie wiersze, należy w filtrze kolumny `Pole` wpisać wartość *Help*.

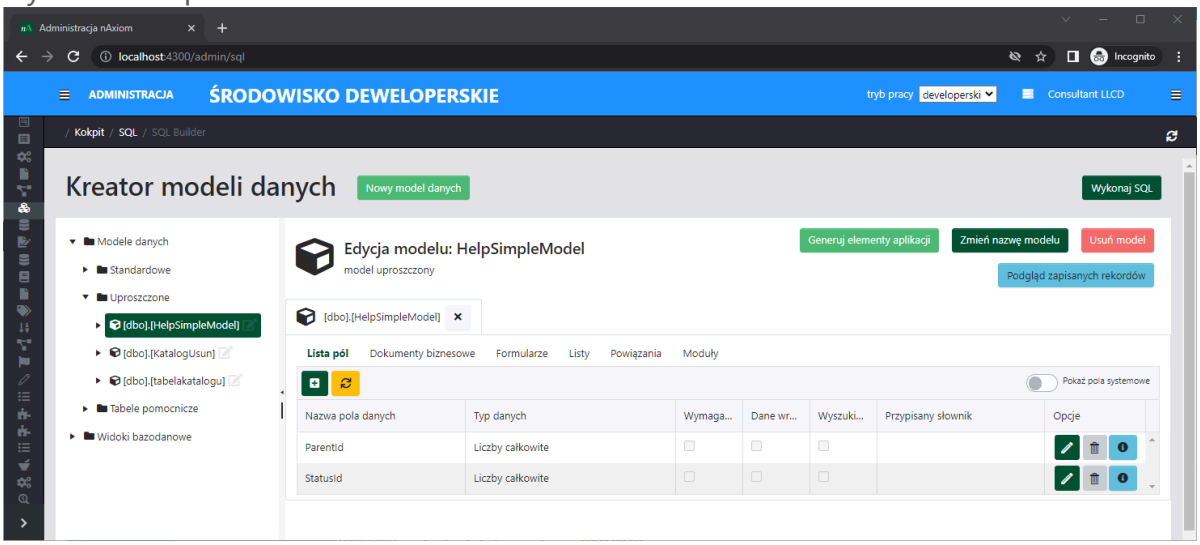
Treść pomocy jest zapisywana w bazie danych w tabeli *core.Translations*.

9.21.2. Definiowanie formularza pomocy

> **Podsumowanie:**
> Do wyświetlania treści pomocy potrzebny jest formularz. Najprościej go utworzyć poleceniem `Generuj elementy aplikacji` w `Kreatorze modeli danych`. Wystarczy użyć uproszczonego modelu danych (formularz będzie wyświetlał statyczne dane i nie będzie służył do przetwarzania dokumentów).

Formularz do wyświetlania treści pomocy można zdefiniować na kilka sposobów. Przedstawiony poniżej ma wyłącznie charakter poglądowy.

1. Zdefiniuj uproszczony model danych i w jego tabeli zdefiniuj pola *ParentId* i *StatusId* (typ *liczba całkowita*, anuluj zaznaczenie pole `Wymagane`). Będą to tzw. *pola techniczne* do przekazania identyfikatorów elementów, dla których ma zostać wyświetlona pomoc.



2. Uruchom polecenie `Generuj elementy aplikacji` dla utworzonego modelu danych.

3. Przejdź do projektu utworzonego formularza i usuń sekcje pól użytkownika, załączników i komentarzy (opcjonalnie), a dla sekcji pól systemowych wyłącz przełączniki `Aktywna` i `Aktywna w wersji mobile` oraz zmniejsz szerokość kolumny macierzystej do 1 kolumny.
4. Dodaj nową kolumnę, a w niej dodaj trzy sekcje o nazwach *Pomoc dla statusu*, *Pomoc dla formularza* i *Pomoc dla dokumentu biznesowego*; dla wszystkich wybierz `Sekcja HTML` jako typ sekcji.
5. Kliknij przycisk `Ustawienia` i skonfiguruj sekcje zgodnie z przykładami podanymi poniżej:
   Sekcja `Pomoc dla statusu`:
   `Zezwól na parsowanie HTML z pól formularza (unsecure)`: włącz ten przełącznik, aby obsłużyć treść pomocy wprowadzoną jako kod HTML.
   `Edytor SQL Query`:

```
SELECT [Value] as TranslatedHelp
  FROM   [core].[Translations] translations
  JOIN [core].[TranslationDefinitions] translationDefinitions
  ON translationdefinitions.Id = translations.TranslationDefinitionId
  JOIN [core].[BusinessStatuses] businessStatuses
  ON businessStatuses.Help = translationdefinitions.Guid
  WHERE  translations.LanguageId = {@_LangId}
  AND businessStatuses.Id = {@StatusId} -- wartość StatusId zostanie przekazana w akcji otwierania f
```

   `Edytor HTML`:

```
<!-- treść pomocy zwrócona przez zapytanie SQL -->
<QueryRowTemplate>{@Query_TranslatedHelp}</QueryRowTemplate>
```

Sekcja `Pomoc dla formularza`:
`Edytor SQL Query`:

```
SELECT [Value] as TranslatedHelp
  FROM   [core].[Translations] translations
  JOIN [core].[TranslationDefinitions] translationDefinitions
  ON translationdefinitions.Id = translations.TranslationDefinitionId
  JOIN [core].[Forms] forms
  ON forms.Help = translationdefinitions.Guid
  WHERE  translations.LanguageId = {@_LangId}
  AND forms.Id = {@ParentId}  -- wartość ParentId zostanie przekazana w akcji otwierania formularza
```

Pozostałe ustawienia, jak w poprzedniej sekcji.

Sekcja `Pomoc dla dokumentu biznesowego`:
`Edytor SQL Query`:

```
SELECT [Value] as TranslatedHelp
  FROM   [core].[Translations] translations
  JOIN [core].[TranslationDefinitions] translationDefinitions
  ON translationdefinitions.Id = translations.TranslationDefinitionId
  JOIN [core].[BusinessDocuments] businessDocuments
  ON businessDocuments.Help = translationdefinitions.Guid
  WHERE  translations.LanguageId = {@_LangId}
  AND businessDocuments.Id =
    (SELECT BusinessDocumentId
       FROM core.Forms
       WHERE Forms.Id = {@ParentId})  -- wartość ParentId zostanie przekazana w akcji otwierania formularza
```

Pozostałe ustawienia, jak w poprzedniej sekcji. Na koniec zapisz formularz.

Następnie utwórz kopię tego formularza w celu wyświetlania pomocy dla list. W tym formularzu możliwe będzie wyświetlanie pomocy dla list i dla dokumentu biznesowego, dlatego usuń sekcję *Pomoc dla statusu*, a nazwę sekcji pomocy dla formularza zmień na `Pomoc dla listy`. Konieczne będą także zmiany zapytania SQL pobierającego treść pomocy do wyświetlenia.

Sekcja `Pomoc dla list`:

```
SELECT [Value] as TranslatedHelp
  FROM   [core].[Translations] translations
  JOIN [core].[TranslationDefinitions] translationDefinitions
  ON translationdefinitions.Id = translations.TranslationDefinitionId
  JOIN [core].[Grids] grids
  ON forms.Help = translationdefinitions.Guid
  WHERE  translations.LanguageId = {@_LangId}
  AND forms.Id = {@ParentId}  -- wartość ParentId zostanie przekazana w akcji otwierania formularza
```

Sekcja `Pomoc dla dokumentu biznesowego`:

```
SELECT [Value] as TranslatedHelp
  FROM   [core].[Translations] translations
  JOIN [core].[TranslationDefinitions] translationDefinitions
  ON translationdefinitions.Id = translations.TranslationDefinitionId
  JOIN [core].[BusinessDocuments] businessDocuments
```

```
ON businessDocuments.Help = translationdefinitions.Guid
WHERE  translations.LanguageId = {@_LangId}
AND businessDocuments.Id =
  (SELECT BusinessDocumentId
     FROM core.Grids
     WHERE Grids.Id = {@ParentId})  -- wartość ParentId zostanie przekazana w akcji otwierania formularza
```

Jest możliwe obsłużenie pomocy za pomocą jednego tylko formularza z czterema sekcjami. Sekcje niepotrzebne w konkretnym przypadku byłyby ukrywane. Jednak takie rozwiązanie wydaje się bardziej skomplikowane.

Powiązane tematy:

- Kreator modeli danych
- Typy sekcji formularza

9.21.3. Akcje przycisku pomocy

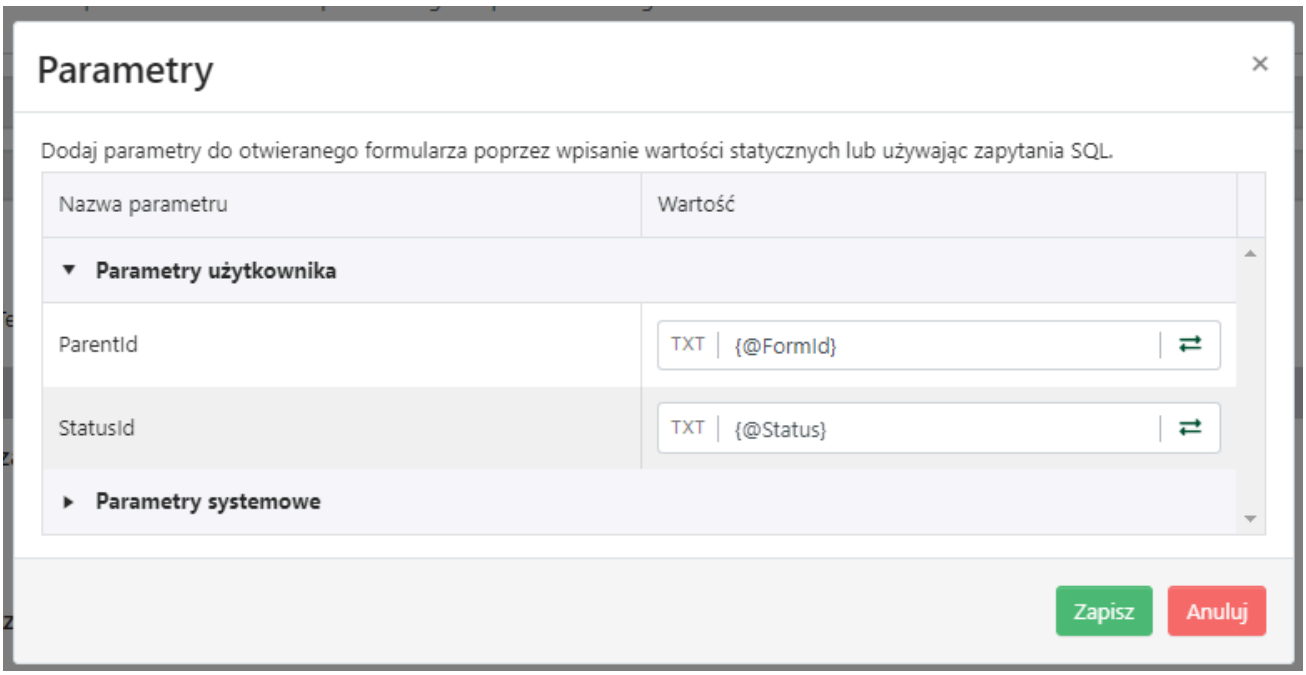
Podsumowanie:
Aby przycisk pomocy wyświetlał formularz, należy mu przypisać odpowiednią akcję otwierania formularza.

Osobno przypisuje się sekwencję akcji przyciskowi wyświetlanemu na listach, a osobno przyciskowi wyświetlanemu na formularzach.

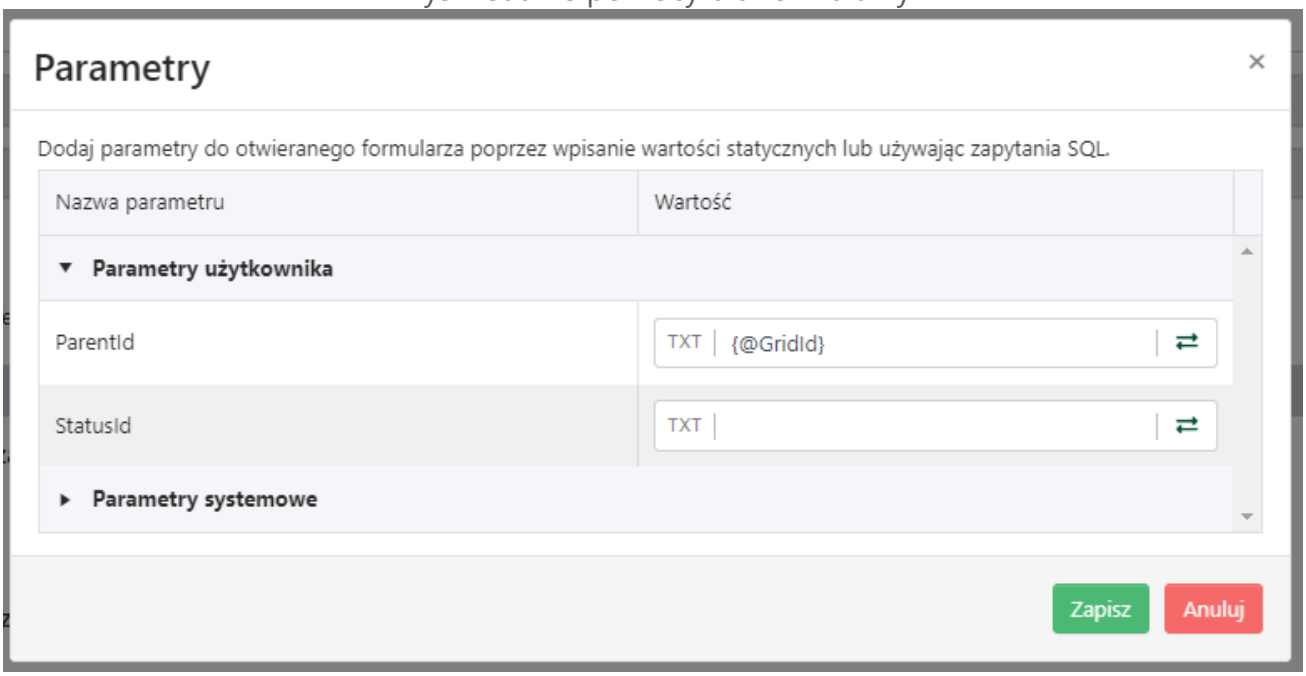
Akcje przycisku pomocy przypisuje się na poziomie środowiska nAxiom w ustawieniach systemowych lub indywidualnie na poziomie definicji dokumentu biznesowego. Przyciskowi można przypisać sekwencję akcji, jednak wymagane jest, aby w tej sekwencji znajdowała się akcja otwierania formularza.

W najprostszym przypadku wystarczy zdefiniować dwie akcje otwierania formularza pomocy, jedną dla formularzy, a drugą dla list.

Kliknij podmenu `Akcje` w menu `NARZĘDZIA`, a następnie kliknij polecenie `Otwieranie formularza`. Kliknij przycisk `Nowa akcja`. Skonfiguruj wymagane parametry. Kliknij przycisk `Wybierz z listy` i wybierz z listy odpowiedni formularz. Następnie kliknij przycisk `Wybierz parametry i wartości` i skonfiguruj przekazywanie identyfikatorów, zależnie od formularza docelowego, jak na poniższych ilustracjach.



Wyświetlanie pomocy dla formularzy



Wyświetlanie pomocy dla list

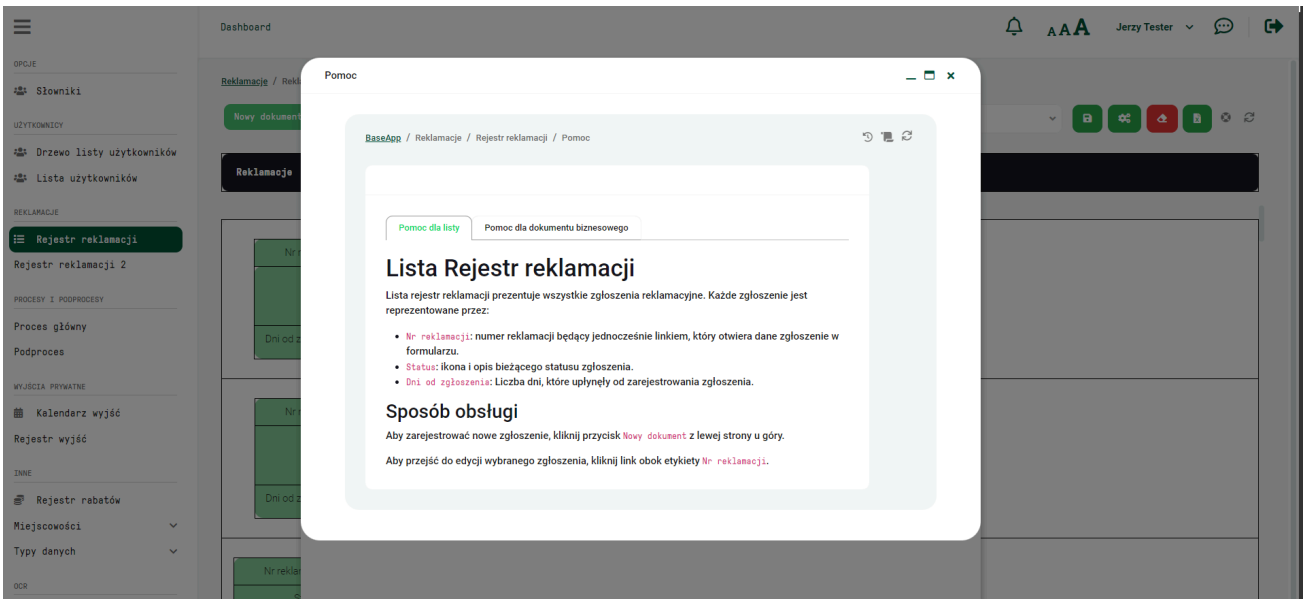
Na podstawie przekazanych wartości parametrów zapytania w sekcjach HTML zwrócą odpowiednią treść pomocy.

Powiązane tematy:

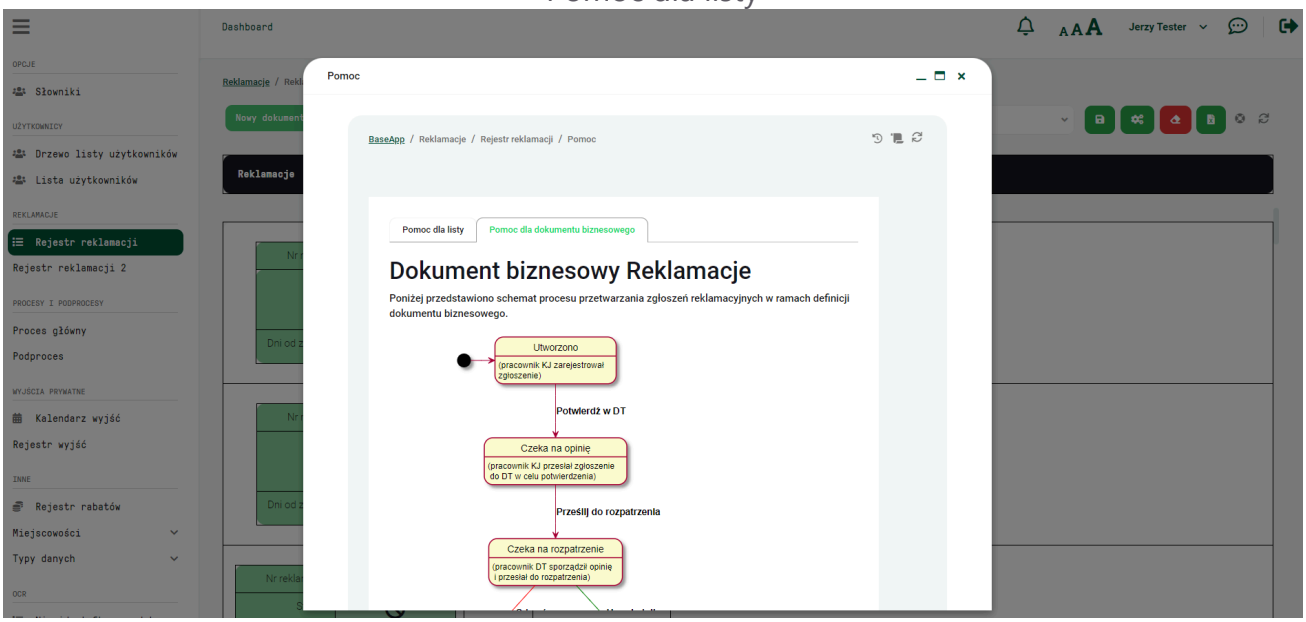
- Akcja otwierania formularza

9.21.4. Przykład pomocy w aplikacji

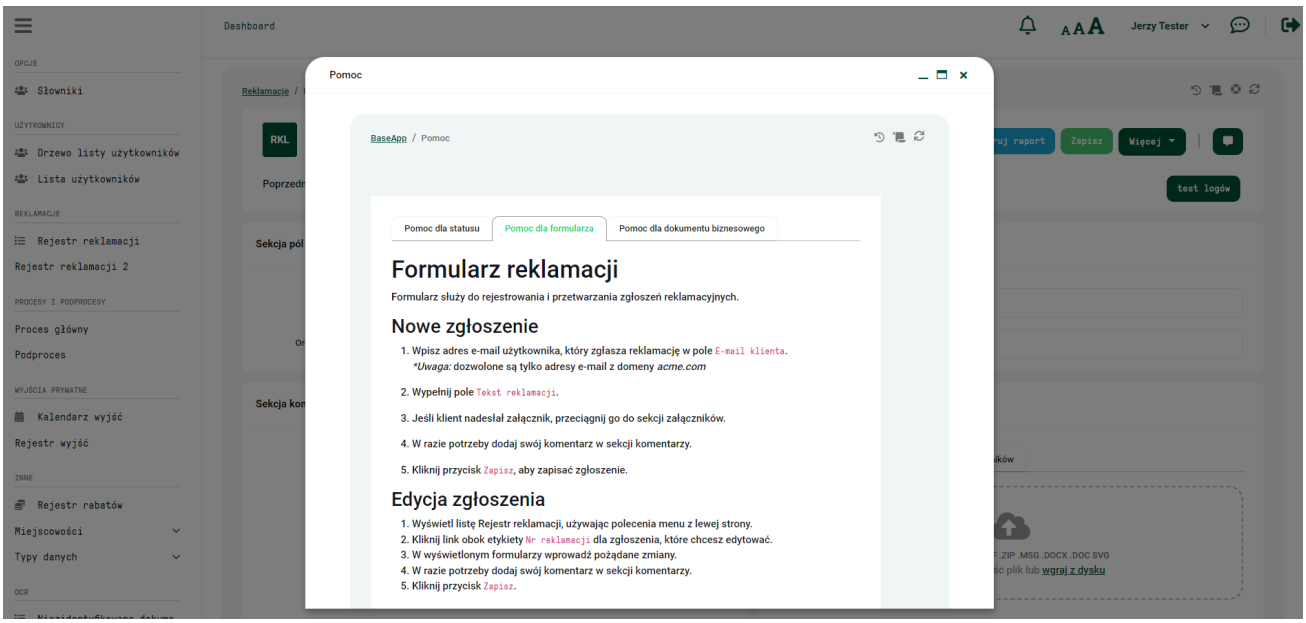
Poniżej przedstawiono ilustracje pomocy wyświetlanej dla równych elementów aplikacji po zastosowaniu konfiguracji opisanej w tym artykule. Przykłady dotyczą aplikacji do rejestrowania zgłoszeń reklamacyjnych, której tworzenie opisano w instrukcji Pierwsze kroki w nAxiom



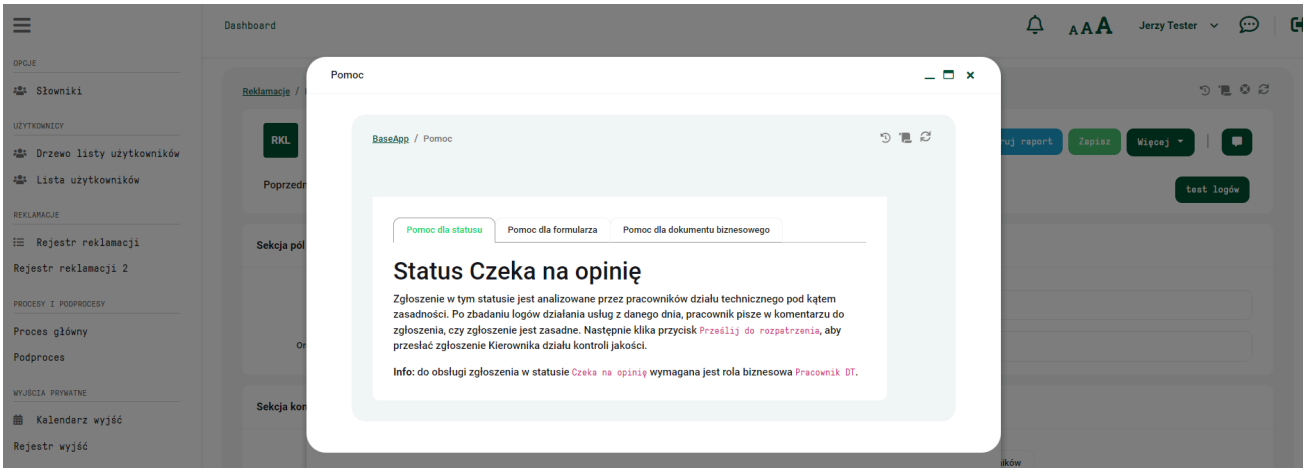
Pomoc dla listy



Pomoc dla dokumentu biznesowego



Pomoc dla formularza



Pomoc dla statusu

9.22. PREDEFINIOWANE FUNKCJE I PROCEDURY

Podsumowanie:
nAxiom oferuje szereg predefiniowanych funkcji i procedur, do których można się odwoływać w oknach edytora SQL, w definicjach różnych obiektów. Ten temat grupuje opisy tych funkcji i procedur.

Funkcje i procedury opisane w kolejnych tematach to:

- core.PrepareEmailToSend – procedura składowana; umieszcza w kolejce wiadomość e-mail do wysyłki.
- core.SubstitutionsPerformedByUser – funkcja tabelaryczna, zwraca wykaz zastępstw pełnionych przez użytkownika o podanym identyfikatorze.

- core.GetAllSubstitutions – funkcja tabelaryczna, zwraca listę wszystkich aktywnych zastępstw.
- core.CalculateOUId – funkcja skalarna, zwraca identyfikator jednostki organizacyjnej na podstawie podanych parametrów.
- JavaScript: showCustomMessage – funkcja JavaScript wyświetlająca własne komunikaty użytkownika.
- core.GetAllCurrentDocumentACLPermissions – funkcja tabelaryczna, zwraca informacje o uprawnieniach do dokumentu o podanym identyfikatorze.
- core.OUInstanceNameTranslations_View – widok bazodanowy z tłumaczeniami nazw jednostek organizacyjnych.

9.22.1. core.PrepareEmailToSend

Podsumowanie:
Procedura składowana *core.PrepareEmailToSend* dodaje do tabeli *core.Emails* rekord z parametrami wiadomości do wysłania. Procedury można używać w akcji SQL do systemowej bazy danych.

Procedura ma następujące parametry:

- *SenderEmail*: (wymagany) adres nadawcy; musi mieć wartość skonfigurowaną w ustawieniach systemu (`Email` > `Nazwa użytkownika powiązana z poświadczeniami` lub `Exchange - Konto użytkownika`).
- *ToEmails*: (wymagany) adresy odbiorców, w przypadku kilku należy je rozdzielic średnikami (;).
- *CcEmails*: (opcjonalny) adresy „do wiadomości".
- *BccEmails*: (opcjonalny) adresy „ukryte do wiadomości".
- *Title*: (obowiązkowy) temat wiadomości.
- *Body*: (obowiązkowy) treść wiadomości.

Przykład wywołania:

```
EXEC core.PrepareEmailToSend
   @SenderEmail = 'default_sender@company.com'
  ,@ToEmails = 'receipient@company.com'
  ,@CcEmails = ''
  ,@BccEmails = ''
  ,@Title = 'message subject'
```

```
  ,@Body = 'message body'
```

Standardowo procedura dodaje jeden rekord, czyli jedną wiadomość e-mail. Aby dodać kilka rekordów, należy przygotować dane źródłowe i użyć procedury w pętli albo z kursorem. Poniżej przedstawiono przykład użycia procedury w pętli.

```
-- deklaracja zmiennych do procedury:
DECLARE @CurrentId int, @CurrentTitle varchar(255), @CurrentPrice decimal(10,2), @CurrentBody nvarchar(max);

-- deklaracja zmiennej tabelarycznej do zasielnia - dane źródłowe
DECLARE @EmailSourceData TABLE (
    [Id] int,
    [Name] varchar(255),
    [Price] decimal(10,2)
);

-- zasilenie zmiennej tabelarycznej
INSERT INTO @EmailSourceData ([Id], [Name], [Price]) VALUES (11, 'GNSS Receiver 11', 511);
INSERT INTO @EmailSourceData ([Id], [Name], [Price]) VALUES (22, 'GNSS Receiver 22', 522);
INSERT INTO @EmailSourceData ([Id], [Name], [Price]) VALUES (33, 'GNSS Receiver 33', 533);

-- deklaracja zmiennych do obsługi pętli: licznik i iterator
DECLARE @RowCount INT = (SELECT COUNT(*) FROM @EmailSourceData);
DECLARE @Counter INT = 1;

-- start pętli
WHILE @Counter <= @RowCount
BEGIN
       -- obliczenie danych dla bieżącego rekordu
    SELECT
         @CurrentId = [Id]
       ,@CurrentTitle = [Name]
       ,@CurrentPrice = [Price]
    FROM (
        SELECT [Id], [Name], [Price], ROW_NUMBER() OVER (ORDER BY Id) AS RowNum
        FROM @EmailSourceData
    ) AS Subquery
    WHERE RowNum = @Counter;

       SET @CurrentBody = concat(@CurrentId, ' | ', @CurrentTitle, ' | ', @CurrentPrice);
```

```
        -- wywołanie procedury
    EXEC core.PrepareEmailToSend
        @SenderEmail = 'system@naxiom.com',
        @ToEmails = 'jmucha@naxiom.com',
        @CcEmails = '',
        @BccEmails = '',
        @Title = @CurrentTitle,
        @Body = @CurrentBody;

        -- przeliczenie iteratora
    SET @Counter = @Counter + 1;
END;
```

Powiązane tematy:

- Ustawienia systemu, sekcja Email

9.22.2. core.SubstitutionsPerformedByUser

Podsumowanie:
Funkcja tabelaryczna *core.SubstitutionsPerformedByUser* zwraca informacje o aktywnych zastępstwach, w których dany użytkownik jest zastępującym. Funkcja zwraca jeden rekord na każde zastępstwo, a w nim informacje o zastępowanym, jego rolach i jednostkach organizacyjnych, czas trwania zastępstwa oraz zakres zastępstwa (właściciel zadania, rola i rola w jednostce). Funkcji można użyć na przykład w widżecie typu sekcja HTML lub na formularzu do prezentowania informacji o zastępstwach bieżącego użytkownika.

Parametry wywołania funkcji:

- *UserId*: identyfikator (GUID) użytkownika zastępującego; jeśli nie zostanie podany, zostanie zwrócona pusta tabela.
- *LangId*: (wymagany) identyfikator języka na potrzeby obsługi tłumaczeń zwracanych informacji.

Przykład wywołania:

```
SELECT *
FROM core.SubstitutionsPerformedByUser ({@UserId}, {@LangId})
```

Funkcja zwraca tabelę, w której każdy rekord odpowiada każdemu aktywnemu (trwającemu) zastępstwu, dla którego podany użytkownik odgrywa rolę zastępującego. Kolumny zwracanej tabeli to:

- *DelegationId*: identyfikator zastępstwa.
- *AbsentUserId*: identyfikator użytkownika zastępowanego.
- *AbsentUserLogin*: nazwa logowania użytkownika zastępowanego.
- *AbsentUserFirstName*: imię użytkownika zastępowanego.
- *AbsentUserLastName*: nazwisko użytkownika zastępowanego.
- *AbsentUserEmail*: adres e-mail użytkownika zastępowanego.
- *AbsentUserGlobalBusinessRoles*: lista rozdzielanych przecinkami nazw globalnych ról biznesowych użytkownika zastępowanego (to jest przypisanych bez związku z przynależnością do jednostek organizacyjnych).
- *AbsentUserBusinessRolesOrgUnit*: lista rozdzielanych przecinkami nazw ról biznesowych i nazw jednostek organizacyjnych użytkownika zastępowanego w formacie *NazwaRoli* ***w*** *NazwaJO*.
- *AbsenceFromDate*: data rozpoczęcia zastępstwa (godzina rozpoczęcia jest automatycznie ustawiana na *00:00:00.0000000*).
- *AbsenceToDate*: data zakończenia zastępstwa (godzina zakończenia jest automatycznie ustawiana na *23:59:59.0000000*).

UWAGA

- Podczas wykonywania tej funkcji nie są sprawdzane uprawnienia PBA danego użytkownika do zastępstw.
- Wywołanie funkcji z parametrem systemowym *{@_UserId}* podczas pracy w trybie zastępstwa zwróci listę zastępstw zdefiniowanych dla zastępowanego.

Powiązane tematy:

- Dodaj zastępstwo
- Edytuj zastępstwo
- Usuń zastępstwo
- core.GetAllSubstitutions

9.22.3. core.GetAllSubstitutions

Podsumowanie:
Funkcja tabelaryczna *core.GetAllSubstitutions* zwraca informacje o wszystkich zastępstwach z tabeli *core.PermissionDelegations*. Funkcja zwraca jeden rekord dla każdego zastępstwa. Funkcji można użyć na przykład w niestandardowej funkcjonalności do obsługi zastępstw w połączeniu z akcjami dodawania, edycji i usuwania zastępstw.

Funkcja nie ma parametrów. Przykład wywołania:

```
SELECT *
FROM core.GetAllSubstitutions ()
```

Funkcja zwraca tabelę rekordów z tabeli *core.PermissionDelegations* rozszerzoną o informacje z tabeli *core.UserProfiles*. Kolumny zwracanej tabeli to:

- *SubstitutionId*: identyfikator zastępstwa.
- *AbsentUserId*: identyfikator użytkownika zastępowanego.
- *AbsentUserFirstName*: imię użytkownika zastępowanego.
- *AbsentUserLastName*: nazwisko użytkownika zastępowanego.
- *AbsentUserEmail*: adres e-mail użytkownika zastępowanego.
- *SubstituteUserId*: identyfikator użytkownika zastępującego.
- *SubstituteUserFirstName*: imię użytkownika zastępującego.
- *SubstituteUserLastName*: nazwisko użytkownika zastępującego.
- *SubstituteUserEmail*: adres e-mail użytkownika zastępującego.
- *BeginDateTime*: data rozpoczęcia zastępstwa (godzina rozpoczęcia jest automatycznie ustawiana na *00:00:00.0000000*).
- *EndDateTime*: data zakończenia zastępstwa (godzina zakończenia jest automatycznie ustawiana na *00:00:00.0000000*).
- *ActivityStatus*: stan flagi aktywności zastępstwa.

UWAGA
Podczas wykonywania tej funkcji nie są sprawdzane uprawnienia PBA danego użytkownika do zastępstw.

Powiązane tematy:

- Dodaj zastępstwo
- Edytuj zastępstwo
- Usuń zastępstwo
- core.SubstitutionsPerformedByUser

9.22.4. core.CalculateOUId

Podsumowanie:
Funkcja skalarna *[core].[calculateOUId]* zwraca identyfikator jednostki organizacyjnej na podstawie przekazanych parametrów. Funkcji można używać wszędzie w definicji zapytań SQL w aplikacji *AdminSPA*, w szczególności w akcji przypisywania roli w jednostce organizacyjnej do grupy interesariuszy.

Parametry wejściowe do funkcji:

- **kod jednostki organizacyjnej** (wymagany): wskazuje jednostkę odniesienia i określa kontekst wyszukiwania żądanej jednostki organizacyjnej; parametr wymagany.
- **identyfikator użytkownika** (opcjonalny): jeśli jest podany, zwracany jest identyfikator jednostki, do której należy użytkownik (kod jednostki odniesienia musi określać jednostkę, do której należy użytkownik lub jej jednostkę nadrzędną dowolnego poziomu).
- **poziom względem jednostki odniesienia** (opcjonalny): wartość liczbowa określająca poziom struktury organizacyjnej względem jednostki odniesienia; dopuszczalne wartości:
    - *0* (lub *NULL*): funkcja zwraca id jednostki, do której należy użytkownik, uwzględniając tylko podaną jednostkę główną i jej podjednostki.
    - *1*: funkcja zwraca id jednostki głównej gałęzi struktury organizacyjnej, w której znajduje się jednostka odniesienia.
    - *-n* (liczba ujemna): zwraca identyfikator jednostki nadrzędnej dla jednostki odniesienia na *n*-tym poziomie w stronę jednostki głównej; np. dla -1 zwraca identyfikator jednostki nadrzędnej dla jednostki odniesienia na następnym poziomie struktury organizacyjnej.

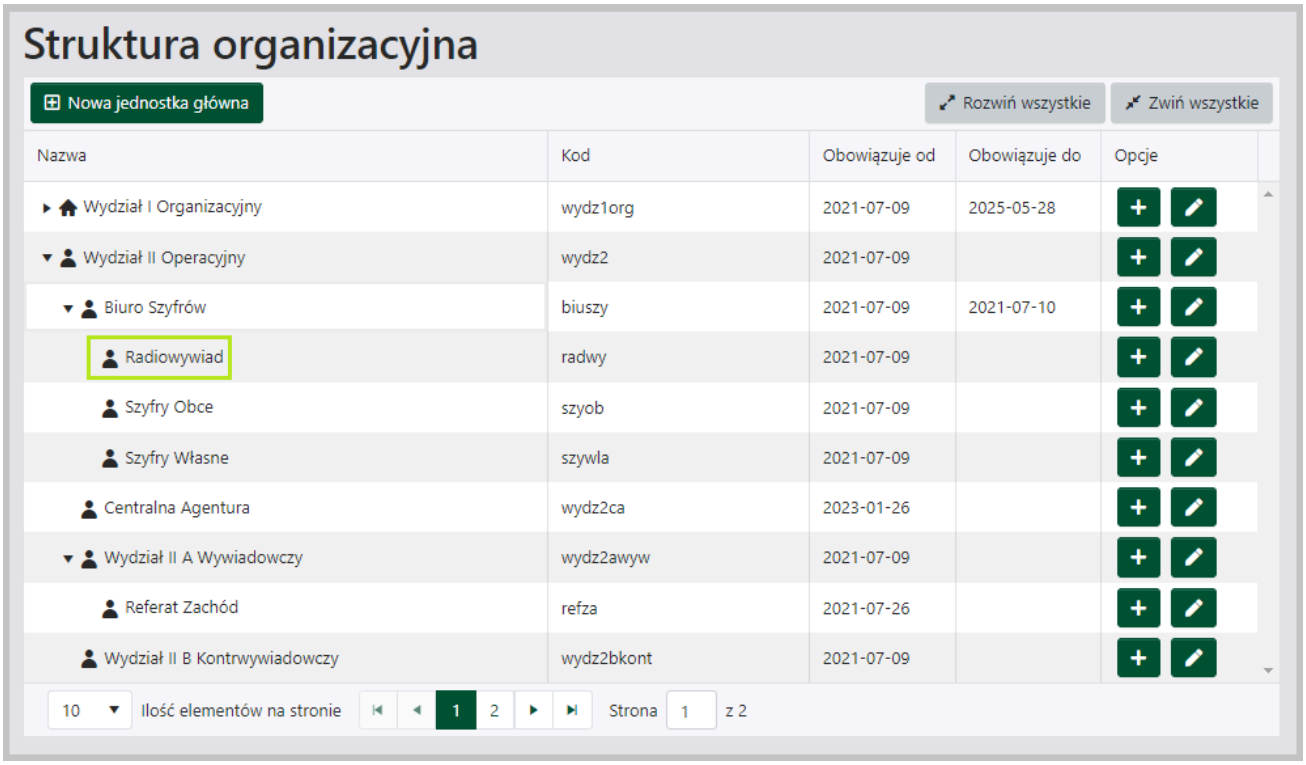
Funkcja zwraca następujące kody błędów (wartości ujemne):

- *-1000*: kod jednostki organizacyjnej odniesienia (parametr *ouRootCode*) nie został podany lub jest pusty.
- *-1001*: jednostka odniesienia o podanym kodzie nie istnieje lub użytkownik nie jest

przypisany do tej jednostki.

- *-1002*: istnieje więcej niż jedna jednostka o podanym kodzie.
- *-2000*: użytkownik o podanym identyfikatorze nie istnieje.
- *-3000*: parametr *orgUnitLevel* ma nieprawidłową wartość (np. >1).
- *-9999*: wystąpił nieobsługiwany wyjątek.

PRZYKŁADY



Przykłądowa struktura organizacyjna

1. Dla użytkownika należącego do jednostki oznaczonej na ilustracji ramką, aby obliczyć kod tej jednostki, należy podać kod tej jednostki, jej jednostki nadrzędnej lub jednostki głównej oraz identyfikator użytkownika (GUID). Trzy zapytania przedstawione poniżej zwrócą ten sam wynik:

```
SELECT core.CalculateOUId('radwy','4A5C7CDA-E44A-4ACC-B5F6-08DBA4950BA2','') AS OUId
SELECT core.CalculateOUId('biuszy','4A5C7CDA-E44A-4ACC-B5F6-08DBA4950BA2','') AS OUId
SELECT core.CalculateOUId('wydz2','4A5C7CDA-E44A-4ACC-B5F6-08DBA4950BA2','') AS OUId
```

   Zamiast pustego trzeciego argumentu można wpisać *0*.

2. Aby obliczyć identyfikator jednostki głównej, mając tylko kod dowolnej z jej jednostek podrzędnych, należy użyć zapytania:

```
SELECT core.CalculateOUId('radwy',null,1) AS OUId
```

> **UWAGA**
> Jeśli identyfikator użytkownika nie jest podawany, konieczne jest użycie literału *null*.

3. Aby obliczyć identyfikator jednostki nadrzędnej dla jednostki odniesienia, należy użyć zapytania:

```
SELECT core.CalculateOUId('radwy',null,-1) AS OUId
```

> **Powiązane tematy:**
> - Struktura organizacyjna

9.22.5. JavaScript: showCustomMessage

> **Podsumowanie:**
> Funkcja JavaScript *showCustomMessage* służy do wyświetlania własnych komunikatów za pomocą akcji JavaScript lub z poziomu elementów interfejsu użytkownika, w których obsługiwany jest kod JavaScript. Na przykład blok HTML, sekcja HTML itp.

Funkcja *showCustomMessage* ma następującą składnię:

```
showCustomMessage('message body', message-type, position)
```

gdzie:

- *'message body'*: treść komunikatu
- *message-type*: rodzaj komunikatu (opcjonalny), dostępne wartości:
  - *'success'*
  - *'warning'*
  - *'error'*
  - *'message'* (dawniej *'info'*, wartość domyślna)
- *position*: położenie komunikatu (opcjonalny, w razie braku stosowane ustawienie systemowe), dostępne wartości:
  - *'top-center'*
  - *'top-right'*
  - *'top-left'*
  - *'bottom-center'*
  - *'bottom-right'*

- *'bottom-left'*

Funkcja *showCustomMessage* została wprowadzona w wersji 1.14 w miejsce wycofanej biblioteki *alertify.js*. Aby znaleźć obiekty, w których odwoływano się do tej biblioteki w celu generowania niestandardowych komunikatów, można użyć poniższych zapytań SQL w programie MS SQL Server Management Studio:

```
SELECT *
  FROM [core].[FormSections]
  WHERE Settings LIKE '%alertify%'

SELECT *
  FROM [core].[Actions]
  WHERE Command LIKE '%alertify%'

SELECT *
  FROM [core].[Forms]
  WHERE ContextChangeScript LIKE '%alertify%'

SELECT *
  FROM [core].[FormFields]
  WHERE ControlValue LIKE '%alertify%'
```

Powiązane tematy:

- Akcja Javascript

9.22.6. core.GetCurrentDocumentACLPermissions

Podsumowanie:
Funkcja tabelaryczna *GetCurrentDocumentACLPermissions(ACLId, LangId)* zwraca informacje o uprawnieniach ACL i ich źródłach dla dokumentu o podanym identyfikatorze *ACLId*. Funkcja jest przeznaczona do diagnostyki schematu uprawnień ACL używanego dla danej definicji dokumentu.

Parametry wywołania funkcji:

- *ACLId* (*uniqueidentifier*): (wymagany) identyfikator dokumentu z kolumny *ACLId* w tabeli dokumentu.
- *LangId* (*int*): (wymagany) identyfikator języka używany do pobrania nazw statusów i grup interesariuszy z tabeli tłumaczeń.

Przykład wywołania:

```
SELECT *
FROM core.GetCurrentDocumentACLPermissions ('82211A86-2622-4124-8565-5C627D0570BD', 1)
```

Funkcja zwraca tabelę rekordów z następującymi kolumnami:

- *IsFromSubstitution*: flaga informująca, czy uprawnienie pochodzi z zastępstwa
- *DocumentOrStatus*: zakres szablonu uprawnień; wartości to:
    - *Document*: uprawnienie na podstawie szablonu instancji
    - *Status: {opis_statusu}*: uprawnienie na podstawie szablonu statusu; podany opis statusu
- *PermissionSource*: źródło uprawnień, dostępne wartości:
    - *Stakeholder: {nazwa_grupy_interesariuszy}*: szablon dla grupy interesariuszy z podaną nazwą grupy
    - *DocumentCreatorTemplate or Action or Administrator operation*: uprawnienie przypisywane indywidualnie na podstawie szablonu twórcy dokumentu, akcją przypisania uprawnień lub przez nadanie użytkownikowi uprawnień na formularzu (ikona kluczyka)
    - *GlobalRoleTemplate or PermissionAction or Administrator operation*: uprawnienie przypisywane roli globalnej na podstawie szablonu roli globalnej, akcją przypisania uprawnień lub przez nadanie roli uprawnień na formularzu (ikona kluczyka)
    - *UserProfile: {Imię i nazwisko (e-mail)}*: imię, nazwisko i e-mail użytkownika zastępowanego w przypadku uprawnień pochodzących z zastępstwa
- *Permission_UserProfile: {Imię i nazwisko (e-mail)}*: imię, nazwisko i e-mail użytkownika w przypadku uprawnień przyznanych indywidualnie
- *Permission_GlobalRoles*: nazwa roli globalnej dla uprawnień przyznawanych posiadaczom roli
- *Permission_OURole_OU*: nazwa jednostki organizacyjnej w przypadku uprawnień wynikających z przypisania roli w jednostce do grupy interesariuszy
- *Permission_OURole_Role*: nazwa roli w jednostce organizacyjnej w przypadku uprawnień wynikających z przypisania roli w jednostce do grupy interesariuszy
- *Permission_OURole_SubOrgUnits*: flaga informująca, czy szablon dla roli miał włączoną opcję podrzędnych jednostek organizacyjnych
- *FullPermissionScope*: lista rozdzielanych spacjami uprawnień, np. *Read Update*
- *Read*: flaga informująca, czy uprawnienie obejmuje odczyt (*R*)
- *Update*: flaga informująca, czy uprawnienie obejmuje aktualizację (*U*)
- *Delete*: flaga informująca, czy uprawnienie obejmuje usunięcie (*D*)
- *Admin*: flaga informująca, czy uprawnienie obejmuje administrowanie uprawnieniami

(*A*)

	Id	IsFromS...	DocumentOrS...	PermissionSource	Permission_UserProfile	Permissi...	Permi...	Permissi...	Permi...	FullPermission...	Read	Update	Delete	Admin
1	1103	0	Document	DocumentCreatorTemplate or Action or A...	Jerzy Tester (dboratyn@naxiom.c...	NULL	NULL	NULL	0	Read Update	1	1	0	0
2	1104	0	Document	Stakeholder: Akceptant w procesie	Anna Kajot-Kierownik (akajot@a...	NULL	NULL	NULL	0	Read Update	1	1	0	0
3	1105	0	Document	Stakeholder: Akceptant w procesie	Sławomir Techniczny (stechnicz...	NULL	NULL	NULL	0	Read Update	1	1	0	0
4	1106	0	Document	Stakeholder: Akceptant w procesie	Anna Kajot-Kierownik (akajot@a...	NULL	NULL	NULL	0	Read Update	1	1	0	0
5	1107	0	Document	Stakeholder: Akceptant w procesie	Sławomir Techniczny (stechnicz...	NULL	NULL	NULL	0	Read Update	1	1	0	0
6	1110	0	Document	Stakeholder: Akceptant w procesie	Anna Kajot-Kierownik (akajot@a...	NULL	NULL	NULL	0	Read Update	1	1	0	0
7	1111	0	Document	Stakeholder: Akceptant w procesie	Sławomir Techniczny (stechnicz...	NULL	NULL	NULL	0	Read Update	1	1	0	0
8	1104	1	Document	UserProfile: Anna Kajot-Kierownik (akajot...	Eugeniusz Neustein (euneu@ac...	NULL	NULL	NULL	0	Read Update	1	1	0	0
9	1106	1	Document	UserProfile: Anna Kajot-Kierownik (akajot...	Eugeniusz Neustein (euneu@ac...	NULL	NULL	NULL	0	Read Update	1	1	0	0
10	1110	1	Document	UserProfile: Anna Kajot-Kierownik (akajot...	Eugeniusz Neustein (euneu@ac...	NULL	NULL	NULL	0	Read Update	1	1	0	0

Informacje o uprawnieniach ACL dokumentu

Powiązane tematy:

- Mechanika uprawnień ACL (dokumenty)

9.22.7. core.OUInstanceNameTranslations_View

Podsumowanie:
Widok bazodanowy *core.OUInstanceNameTranslations_View* zawiera tłumaczenia nazw jednostek organizacyjnych i ma na celu poprawę wydajności list zwracających nazwy jednostek organizacyjnych.

Widok zawiera trzy kolumny:

- *RecordRowGuid*: identyfikator jednostki organizacyjnej (wartość *RowGuid* z tabeli *core.OUInstances*)
- *LanguageId*: identyfikator języka
- *Value*: nazwa JO w danym języku

Na kolumnach widoku *RecordRowGuid* i *Value* utworzono indeks, który przyspiesza wyszukiwanie.

10. FrontSPA — przewodnik użytkownika

Podsumowanie:
FrontSPA (czasem także *front-office*) to ta część witryny nAxiom, w której użytkownicy końcowi korzystają z aplikacji biznesowych. Serwis *FrontSPA* jest dostępny pod adresem *adresUrlWitrynyNaxiom/**front***. Na ten adres automatycznie przekierowywani są logujący się użytkownicy, którzy mają rolę systemową *Klient* i/lub *Klient uproszczony*.

Użytkownicy, którzy pracują w aplikacjach stworzonych przez konsultantów, mają do czynienia z pewnymi standardowymi elementami i funkcjonalnościami, które zostaną opisane w tym dodatku. Po stronie front-office dostępne są także pewne elementy służące do administrowania witryną i konfigurowania jej, które również zostaną tu opisane.

10.1. TRYB POJEDYNCZEJ APLIKACJI

Podsumowanie:
Tryb pojedynczej aplikacji pozwala ograniczyć sekcje menu, które widzi użytkownik końcowy we FrontSPA, tylko do tych, które przypisano do wybranej aplikacji. Ten tryb musi zostać włączony przez konsultanta w ustawieniach systemu. Tam też określa się aplikację domyślną dla trybu pojedynczej aplikacji w przypadku, gdy takiej aplikacji nie ustawiono w profilu użytkownika.
Oprócz sekcji przypisanych do wyświetlanej aplikacji wyświetlane są też pozycje menu w sekcji `USTAWIENIA SYSTEMOWE` oraz pozycje menu innych aplikacji, dla których zaznaczono przełącznik `Wyświetlaj w sekcji współdzielonej`.

UWAGA
Aplikacje, które użytkownik może wybierać w trybie pojedynczej aplikacji oraz ustawiać w swoim profilu jako domyślna, zależą od posiadanych przez niego ról. Patrz opis definicji aplikacji (link na dole strony).

Po zalogowaniu do FrontSPA użytkownik może kliknąć symbol menu aplikacji w lewym górnym rogu (1), aby rozwinąć listę aplikacji i wybrać aplikację do pracy. Po wybraniu aplikacji, pod logo nAxiom zostaną wyświetlone akronim i nazwa (2). Natomiast w menu będą dostępne tylko sekcje menu powiązane z tą aplikacją (3). Dodatkowo wyświetlane będą sekcja współdzielona (4) oraz sekcja `USTAWIENIA SYSTEMOWE` (5). W sekcji współdzielonej, która nie ma nazwy, wyświetlane są pozycje menu, w których definicji włączono przełącznik `Wyświetlaj w sekcji współdzielonej`. Natomiast sekcja `USTAWIENIA SYSTEMOWE` zawiera pozycje menu umożliwiające wykonywanie czynności administracyjnych, odpowiednio do uprawnień PBA, które posiada bieżący użytkownik.

Aplikację wyświetlaną domyślnie można przypisać indywidualnie każdemu użytkownikowi w profilu użytkownika. Użytkownicy, którzy mają uprawnienie PBA do edycji własnego profilu mogą sami ustawić sobie aplikację domyślną. Dla użytkowników, którzy nie mają zdefiniowanej tej właściwości, aplikacją domyślną będzie aplikacja wskazana w ustawieniach systemu, gdzie można włączyć lub wyłączyć tryb pojedynczej aplikacji.

Powiązane tematy:

- Definicja aplikacji, karta Ogólne
- Definicja pozycji menu
- Ustawienia systemu, Tryb pojedynczej aplikacji

10.2. PULPITY UŻYTKOWNIKA

Podsumowanie:
Funkcjonalność pulpitów pozwala użytkownikom końcowym definiować wygląd strony głównej witryny nAxiom wyświetlanej po zalogowaniu się w aplikacji *FrontSPA* (oraz po kliknięciu w logo w aplikacji *FrontSPA*). Pulpity składają się z widżetów definiowanych przez konsultantów.

Oprócz pulpitów użytkownika w nAxiom można definiować także pulpity aplikacji.

Aby utworzyć pulpit użytkownika, w aplikacji *FrontSPA* kliknij wyboru z lewej strony nagłówka (z nazwą bieżącego pulpitu lub poleceniem `wybierz pulpit`) i wybrać z niej polecenie `Lista pulpitów`. W wyświetlonym oknie dialogowym kliknij przycisk `Nowy pulpit`.

Następnie zdefiniuj nowy pulpit, podając następujące właściwości:

- `Nazwa`
- `Opis`
- `Aktywny`: pulpit nieaktywny nie jest dostępny na liście wyboru.
- `Domyślny`: flaga decydująca, czy dany pulpit ma być wyświetlany domyślnie po zalogowaniu; może być ustawiana tylko przez właściciela pulpitu.
- `Dostęp`: lista wyboru określająca dostępność pulpitu dla innych użytkowników; dostępne pozycje:
  - `Prywatny`: inni użytkownicy nie widzą pulpitu na liście i nie mogą go wybierać.
  - `Publiczny`: wszyscy użytkownicy widzą pulpit na liście i mogą go wybierać.
  - `Grupa`: tylko użytkownicy z wybraną rolą widzą pulpit na liście i mogą go wybierać.
- `Dostępny dla ról`: lista wielowyboru widoczna tylko w przypadku, gdy z listy `Dostęp` wybrano pozycję `Grupa`; pozwala wybrać jedną lub kilka ról biznesowych w celu udostępnienia pulpitu użytkownikom z wybraną rolą.

Po zapisaniu nowy pulpit będzie wyświetlany na liście. Użytkownik, który utworzył pulpit, może go ustawić jako domyślny, ma także prawo do jego edycji. Użytkownicy, którym udostępniono pulpit, mogą utworzyć jego kopię jako własny pulpit, z uprawnieniami do jej edycji.

UKŁAD PULPITU

Właściciel pulpitu wyświetlanego na liście może przejść do edycji układu widżetów na tym tego pulpicie, wybierając z listy pulpitów polecenie `Edytuj bieżący pulpit` lub klikając przycisk `Designer` w rekordzie jednego ze swoich pulpitów.

Spowoduje to wyświetlenie pulpitu w trybie edycji. W tym trybie u góry pulpitu są wyświetlane trzy polecenia:

- `Wyjdź z trybu edycji`: zamyka tryb edycji i zapamiętuje wprowadzone zmiany.
- `Edytuj układ kolumn`: wyświetla okienko, w którym można wybrać układ pulpitu: jedno-, dwu lub trójkolumnowy.
- `Dodaj widżet`: wyświetla listę widżetów; aby dodać widżet do pulpitu, kliknij znak + z prawej strony.

Widżety można przeciągać między kolumnami oraz ustawiać jeden pod drugim w tej samej kolumnie.

Powiązane tematy:

- Widżety

10.3. POWIADOMIENIA I ALERTY

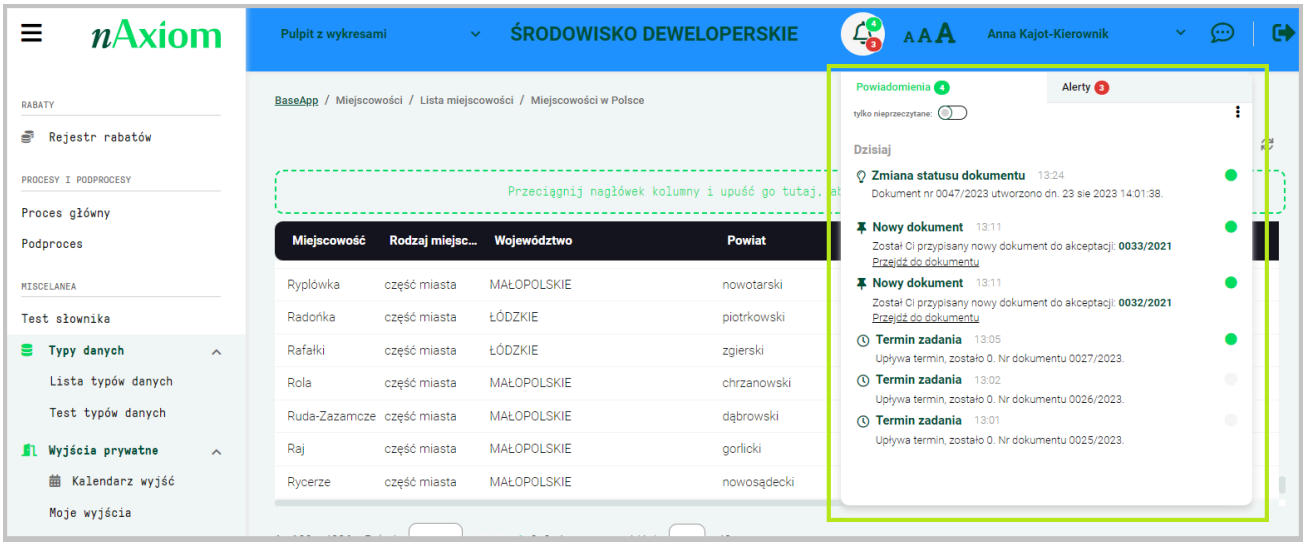
Podsumowanie:
Funkcjonalność powiadomień pozwala przesyłać do użytkowników aplikacji *FrontSPA* następujące rodzaje komunikatów:

- eskalatory, czyli powiadomienia o zbliżającym się lub przekroczonym terminie wykonania kroku procesu w przetwarzaniu dokumentu,
- powiadomienia o dokumentach oczekujących na akceptację,
- powiadomienia wysyłane w wyniku akcji wysyłania powiadomień.

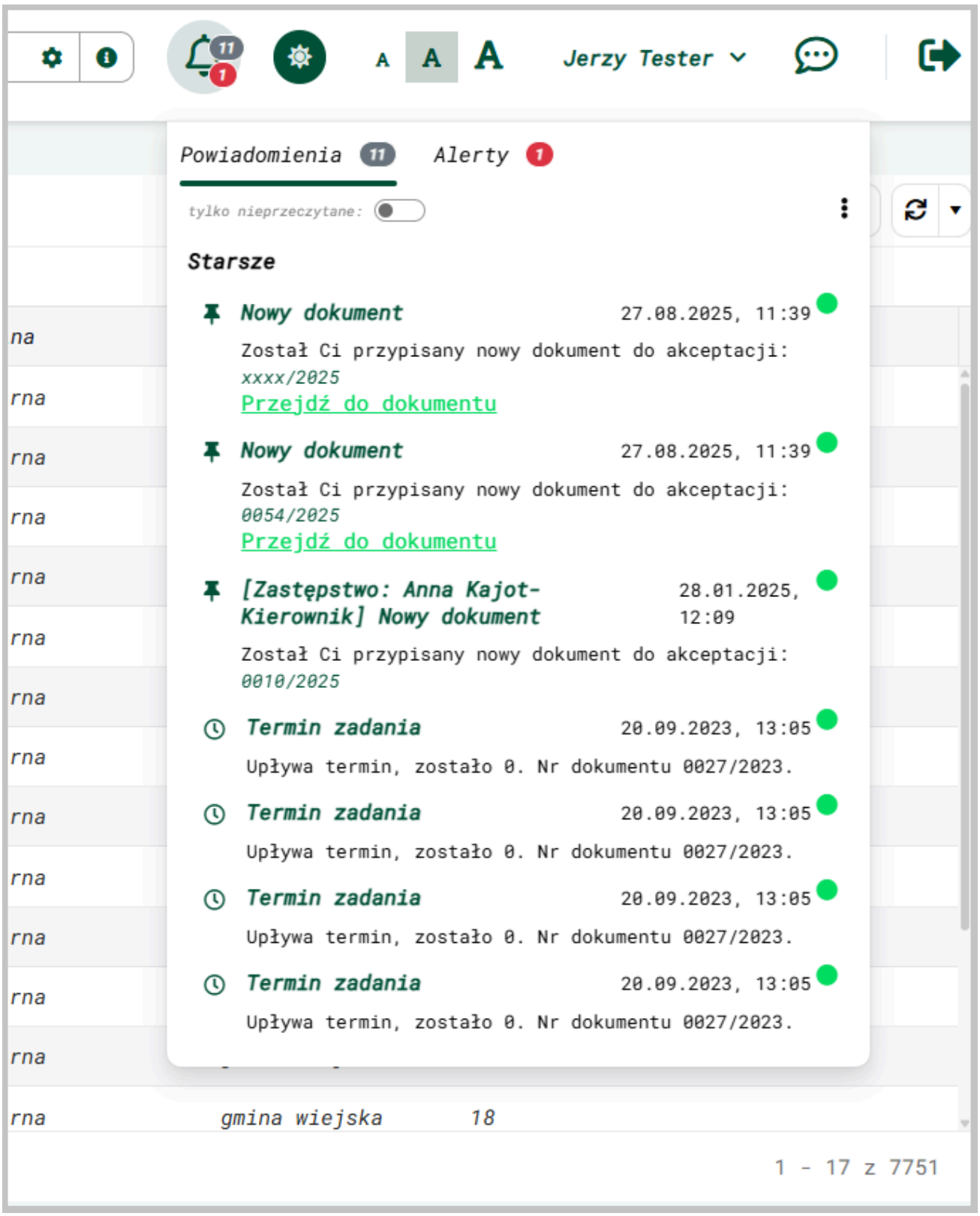
Dodatkowo, panel powiadomień wykorzystano do rejestrowania informacji o błędach, które wystąpiły podczas pracy użytkownika. Ta funkcjonalność pozwala przesyłać treść komunikatu i ewentualne uwagi użytkownika do osoby odpowiedzialnej za techniczną stronę działania aplikacji.

OSTRZEŻENIE
Funkcje czatu i powiadomień korzystają z biblioteki **SignalR**. Za jej włączenie odpowiadają flagi `"useSignalR": true` i `"chat": true` w pliku *front\assets\config\config.production.json*. Flagi te są domyślnie włączone przez instalator nAxiom. Ustawienie dla jednej z tych flag wartości *false* spowoduje wyłączenie zarówno czatu, jak i powiadomień.

W nagłówku pulpitu aplikacji *FrontSPA* znajduje się ikona dzwonka, której kliknięcie wyświetla panel z dwoma kartami: `Powiadomienia` i `Alerty`. Wraz z ikoną wyświetlane są liczniki powiadomień (zielone tło) i alertów (czerwone tło).



Panel powiadomień



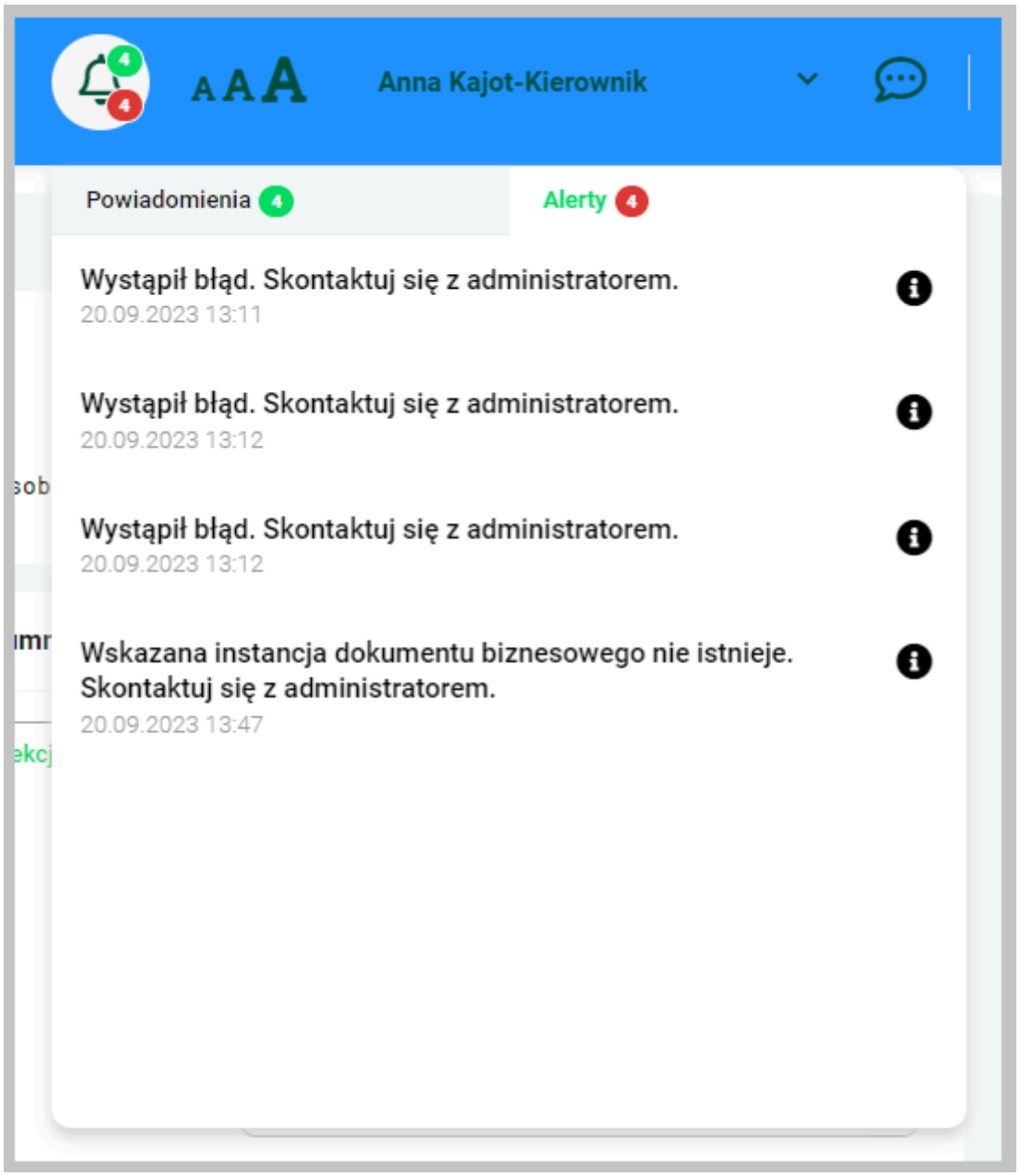
Panel powiadomień

Na panelu powiadomień wyświetlana jest lista powiadomień o dokumentach oczekujących na akceptację, zbliżających sie lub przekroczonych terminach oraz innych, które można dowolnie konfigurować w akcji powiadomień. Powiadomienia nieprzeczytane są oznaczone zieloną kropką z prawej strony. Ponadto u góry panelu jest dostępny przełącznik `tylko nieprzeczytane`, który pozwala ograniczyć listę wyświetlanych powiadomień.

W prawym górnym rogu panelu powiadomień znajduje się przycisk z wielokrokiem, który wyświetla menu z poleceniami do zarządzania powiadomieniami:

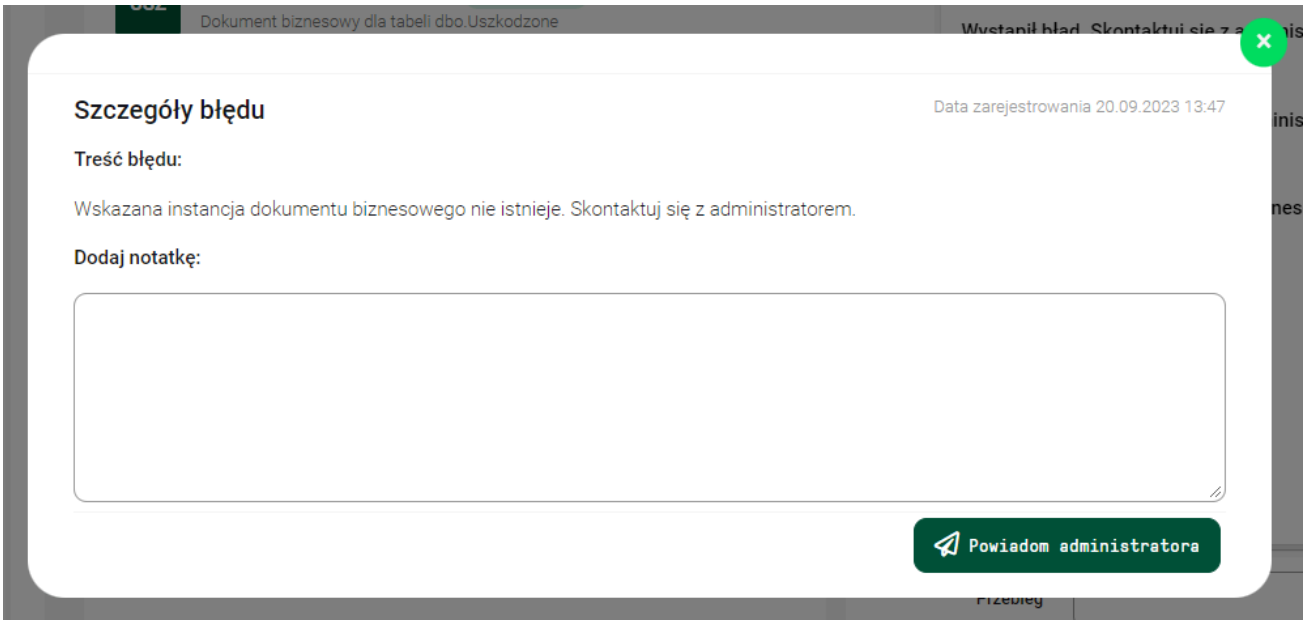
- `oznacz wszystkie jako przeczytane`
- `oznacz wszystkie jako nieprzeczytane`
- `usuń przeczytane`
- `usuń wszystkie`

Na panelu `Alerty` są wyświetlane informacje o błędach, które wystąpiły podczas pracy użytkownika. Są to tylko błędy, które wystąpiły po stronie serwera. Najczęściej oryginalny komunikat o błędzie (zapisywany w logach systemu) jest opakowywany w ogólny komunikat: *„Wystąpił błąd. Skontaktuj się z administratorem"*.



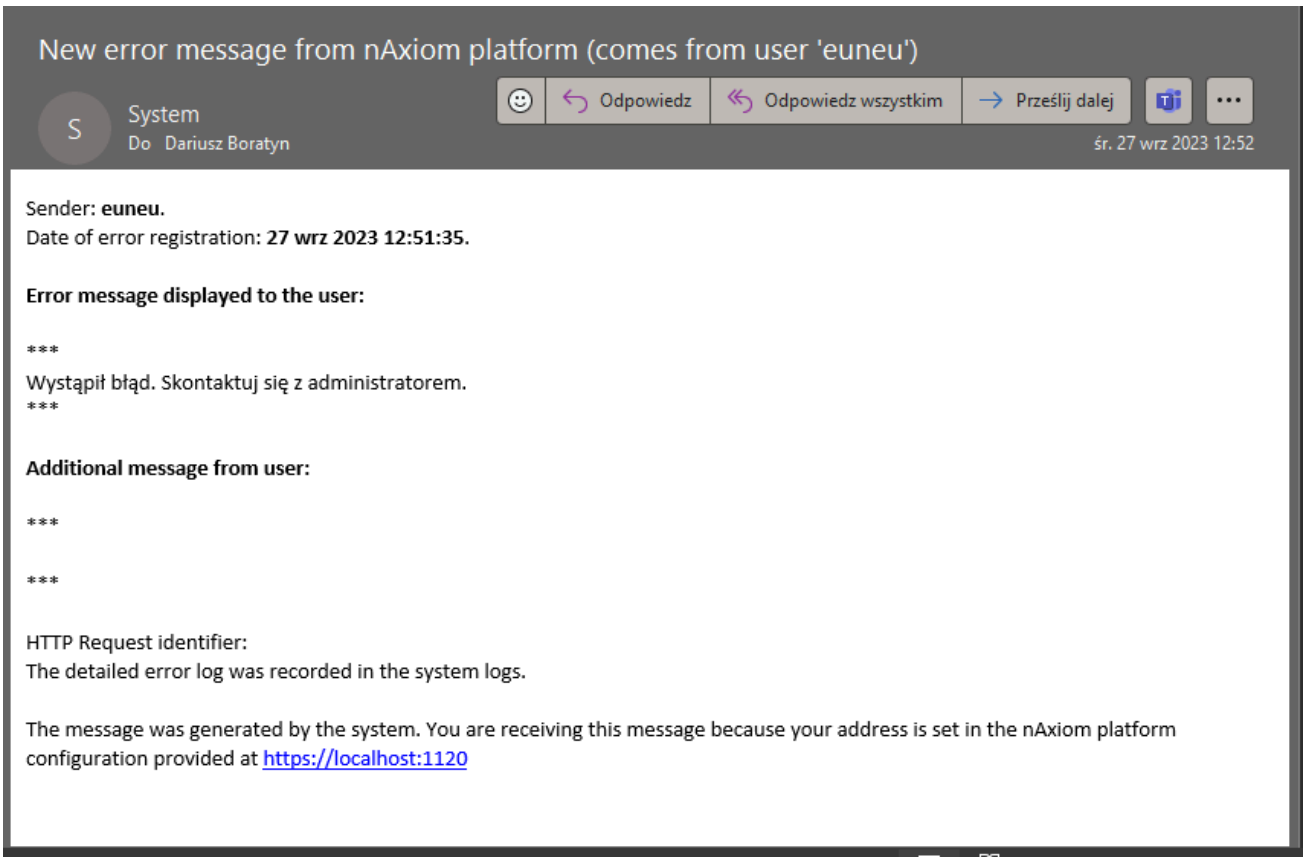
Panel alertów

Kliknięcie ikony z symbolem *i* z prawej strony wpisu wyświetla okno dialogowe szczegółów błędu.



Szczegóły błędu

Obszar tekstowy `Dodaj notatkę` i przycisk `Powiadom administratora` są dostępne pod warunkiem, że w ustawieniach systemu podano adres osoby odpowiedzialnej (`Ustawienia systemu` > `Notyfikacje` > `Adres email do wysyłki powiadomień o błędach od użytkowników`). W wiadomości do administratora, oprócz daty i godziny wystąpienia błędu, treści komunikatu i uwag użytkownika, wysyłany jest identyfikator żądania HTTP, na podstawie którego można odnaleźć odpowiedni wpis w logach systemu.



Wiadomość e-mail z powiadomieniem o błędzie

Panel alerty zawiera tylko informacje o błędach, które wystąpiły w bieżącej sesji. Po wylogowaniu użytkownika i ponownym zalogowaniu jego zawartość zostanie wyczyszczona.

Powiązane tematy:

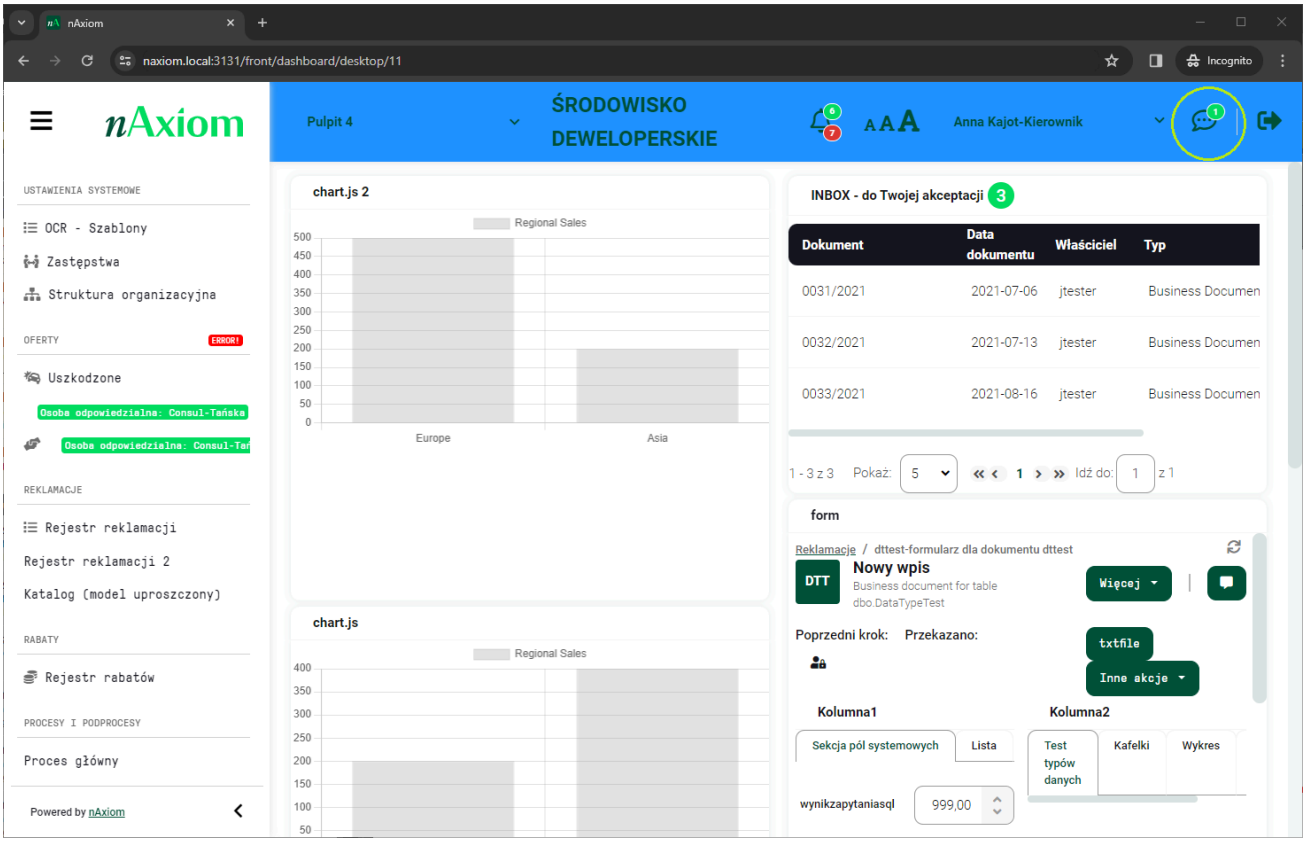
- Akcja wysyłania powiadomień
- Termin i eskalatory w zadaniu ręcznym
- Konfiguracja eskalatorów w definicji dokumentu
- Adres e-mail administratora

10.4. CZAT

Podsumowanie:
Funkcjonalność czatu umożliwia przesyłanie wiadomości tekstowych do innych użytkowników. Jeśli użytkownicy są jednocześnie zalogowani, mogą komunikować się ze sobą w czasie rzeczywistym. Funkcjonalność obejmuje tylko komunikację dwustronną. Konwersacje w większej grupie nie są obsługiwane.

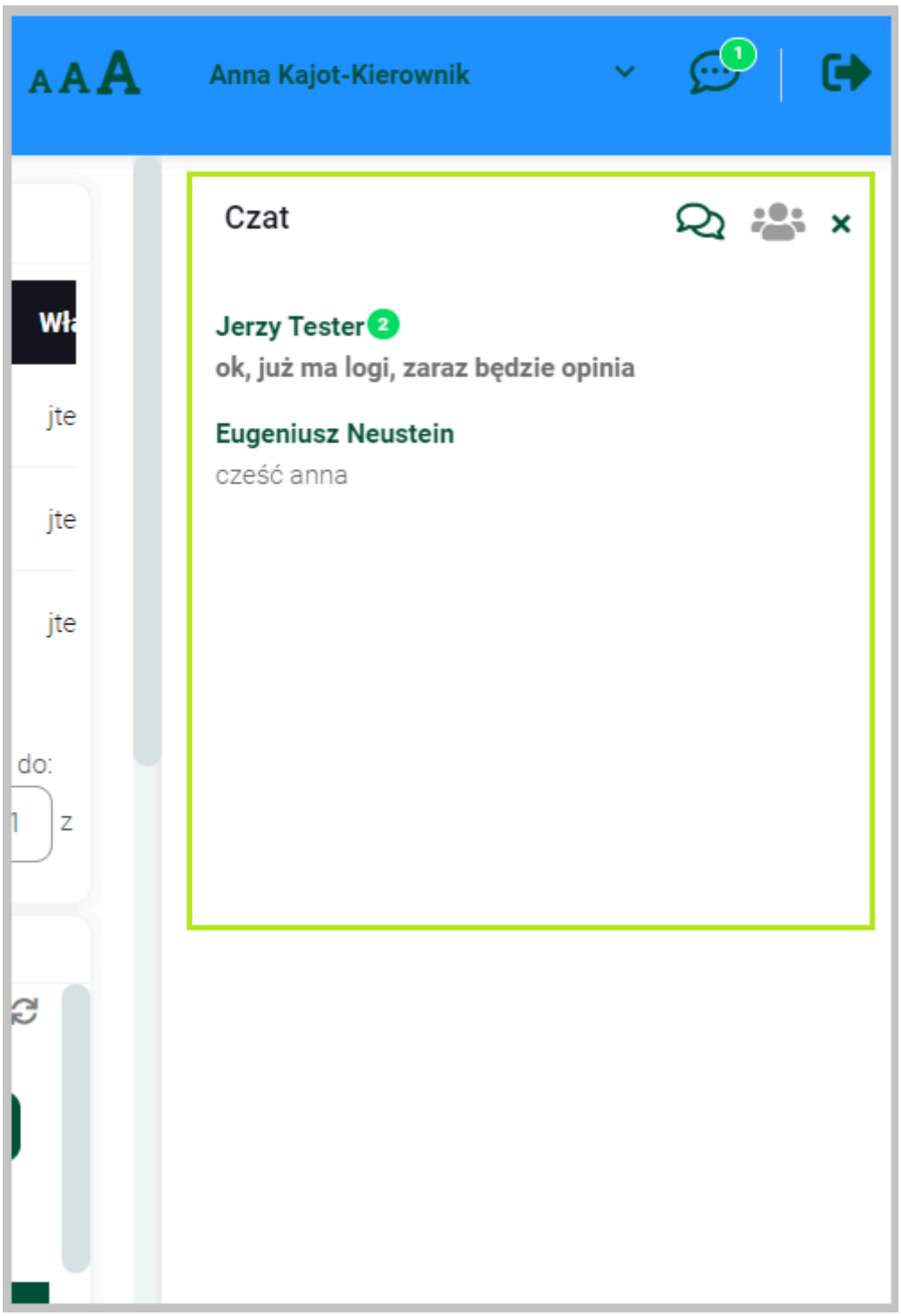
OSTRZEŻENIE

Funkcje czatu i powiadomień korzystają z biblioteki **SignalR**. Za jej włączenie odpowiadają flagi `"useSignalR": true` i `"chat": true` w pliku *front\assets\config\config.production.json*. Flagi te są domyślnie włączone przez instalator nAxiom. Ustawienie dla jednej z tych flag wartości *false* spowoduje wyłączenie zarówno czatu, jak i powiadomień.



Ikona czatu

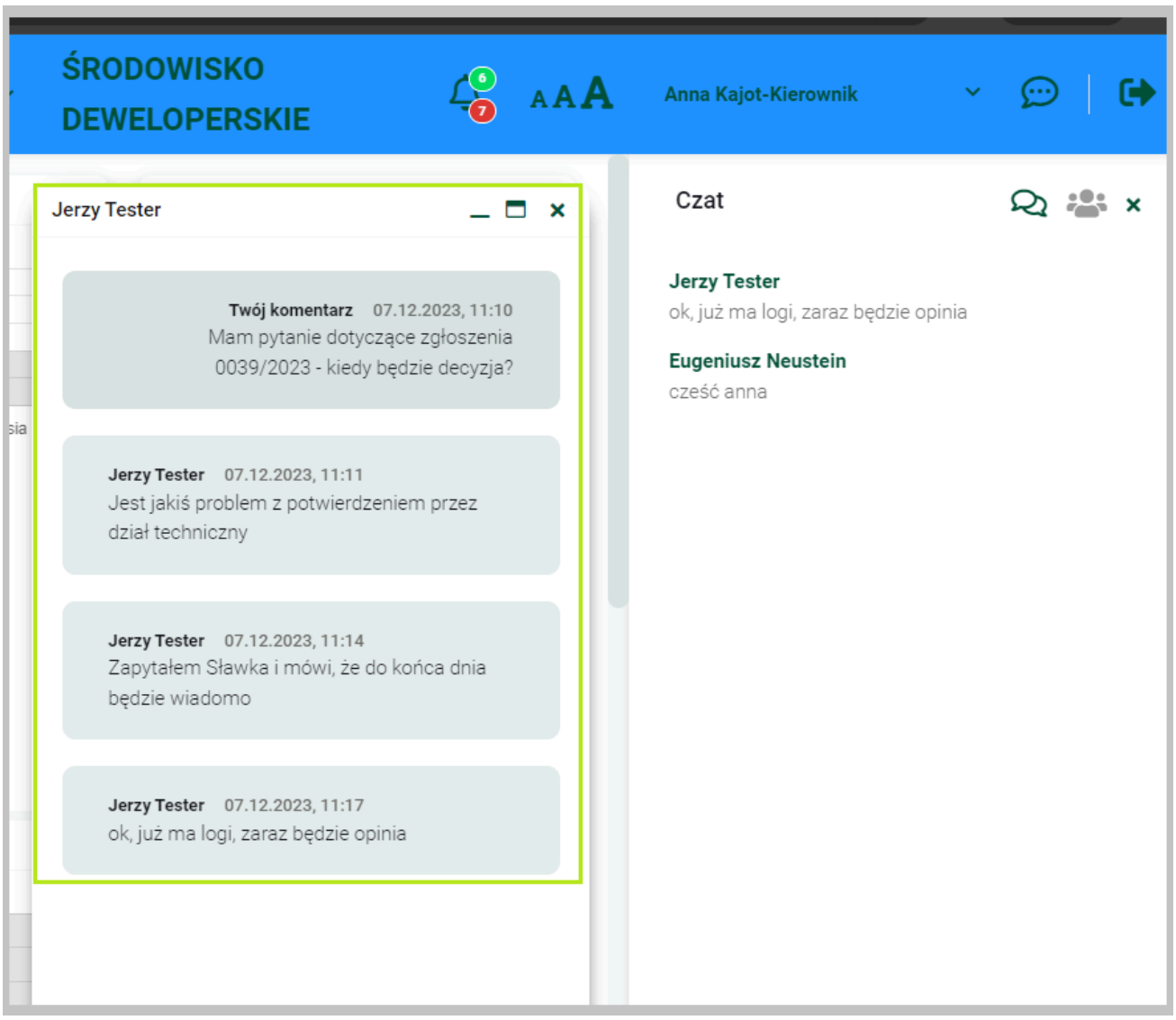
W nagłówku pulpitu aplikacji *FrontSPA* znajduje się ikona czatu (dymek z wielokropkiem), której kliknięcie wyświetla panel czatu. Wraz z ikoną wyświetlany jest licznik konwersacji, w których są nieprzeczytane wiadomości.



Panel czatu

Panel czatu zawiera listę konwersacji oraz ikony do przełączania między listą kontaktów i listą konwersacji. Aby rozpocząć konwersację, należy przejść na listę kontaktów i kliknąć wybrany kontakt. Jeśli kontaktu nie ma na liście, należy go wyszukać, wpisując 3 pierwsze litery loginu, imienia lub nazwiska. Kliknięcie kontaktu, otworzy panel konwersacji.

Panel czatu zawiera listę konwersacji oraz pole do wyszukiwania listą kontaktów. Aby rozpocząć konwersację, należy kliknąć ostatnią rozmowę z danym kontaktem lub znaleźć kontakt, wpisując co najmniej 3 litery loginu, imienia lub nazwiska. Kliknięcie kontaktu, otworzy panel konwersacji.



Panel konwersacji

Powiadomienia są zapisywane w tabeli *core.ChatMessages*. Dodatkowo do obsługi czatu używane są tabele *core.ChatRooms* i *core.ChatRoomUsers*, które zawierają informacje o konwersacjach (pokojach czatu) i ich uczestnikach.

10.5. WIDOK LISTY

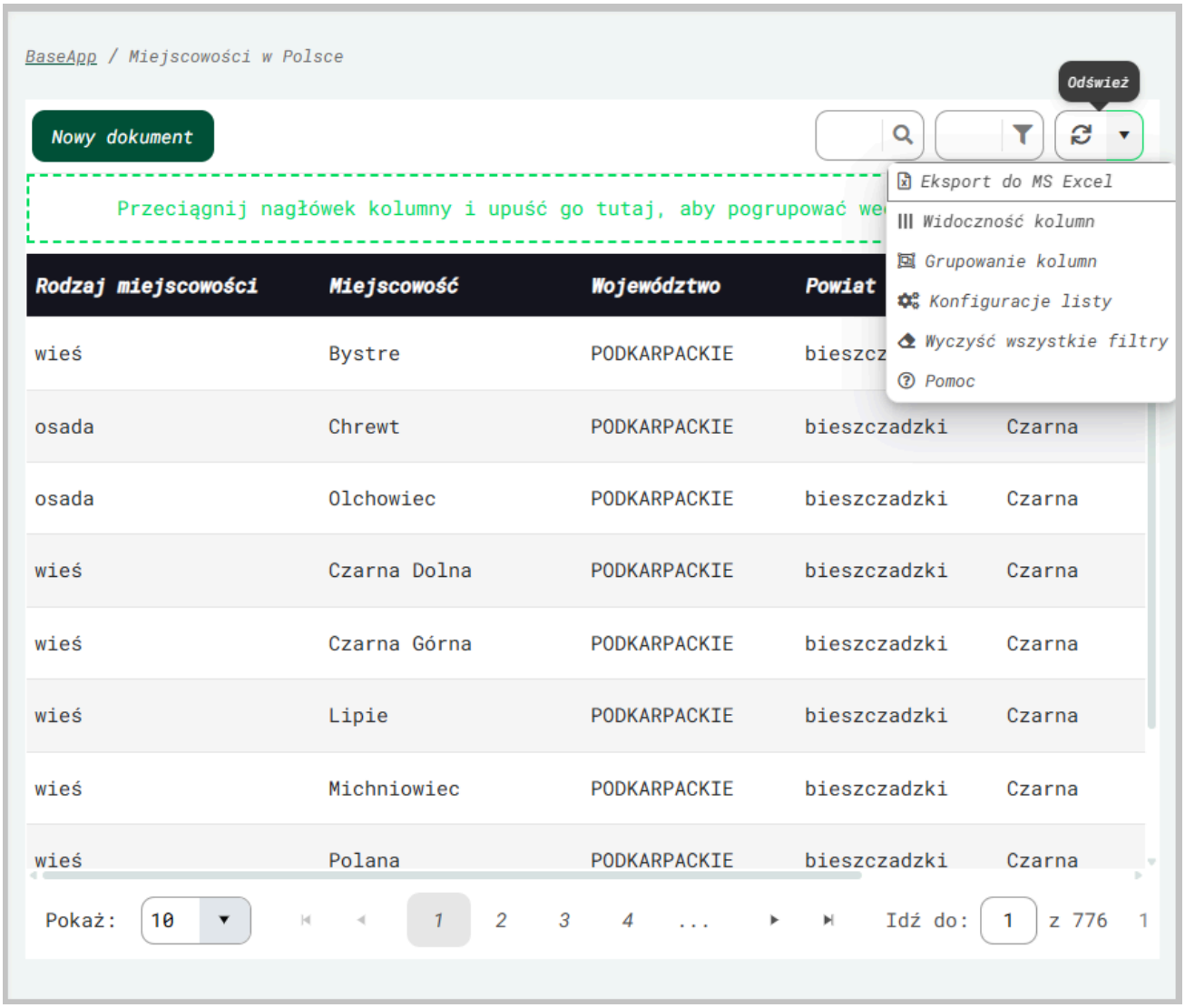
Podsumowanie:
Widok listy to, oprócz formularza, podstawowy sposób prezentacji danych w aplikacjach biznesowych budowanych na platformie nAxiom. Widok listy może być wyświetlany poleceniem menu, może być także osadzony jako sekcja na formularzu.

W TYM TEMACIE

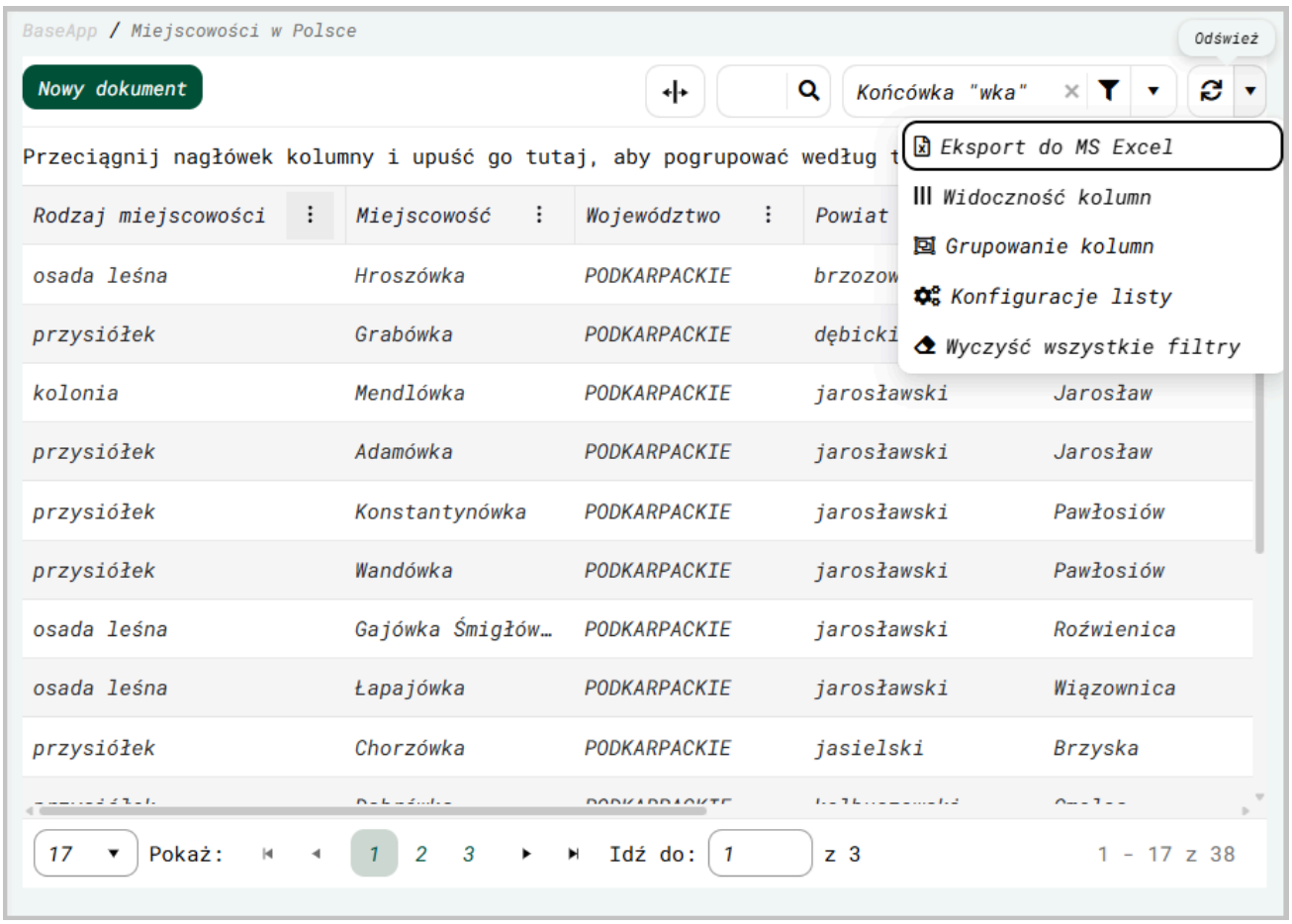
- Elementy widoku listy
- Filtry zaawansowane

ELEMENTY WIDOKU LISTY

Poniższa ilustracja przedstawia przykładowy widok listy w aplikacji *FrontSPA*.



Przykładowy widok listy w aplikacji FrontSPA



Przykładowy widok listy w aplikacji FrontSPA

Na górze listy znajduje się *panel górny*, który zawiera następujące elementy (od lewej):

- przyciski akcji; przycisk `Nowy dokument` jest standardowo dodawany na listach tworzonych w procesie automatycznego generowania elementów aplikacji; przycisk otwiera formularz powiązany z daną listę, w którym można utworzyć nowy dokument. Konsultant może dodawać inne przyciski na panelu górnym.

- przyciski akcji; przycisk `Nowy dokument` jest standardowo dodawany na listach tworzonych w procesie automatycznego generowania elementów aplikacji; przycisk otwiera formularz powiązany z daną listę, w którym można utworzyć nowy dokument. Konsultant może dodawać inne przyciski na panelu górnym.
- przycisk trybu zmiany szerokości kolumn, po kliknięciu umożliwia wybór jednego z dwóch trybów:
  - `Rozszerzanie kolumny kosztem sąsiedniej`: zmiana szerokości kolumny powoduje powoduje zmianę szerokości kolumny znajdującej się po prawej stronie.
  - `Dowolne rozszerzanie kolumn`: zmiana szerokości kolumny powoduje powoduje przesunięcie kolumn znajdujących się po prawej stronie.

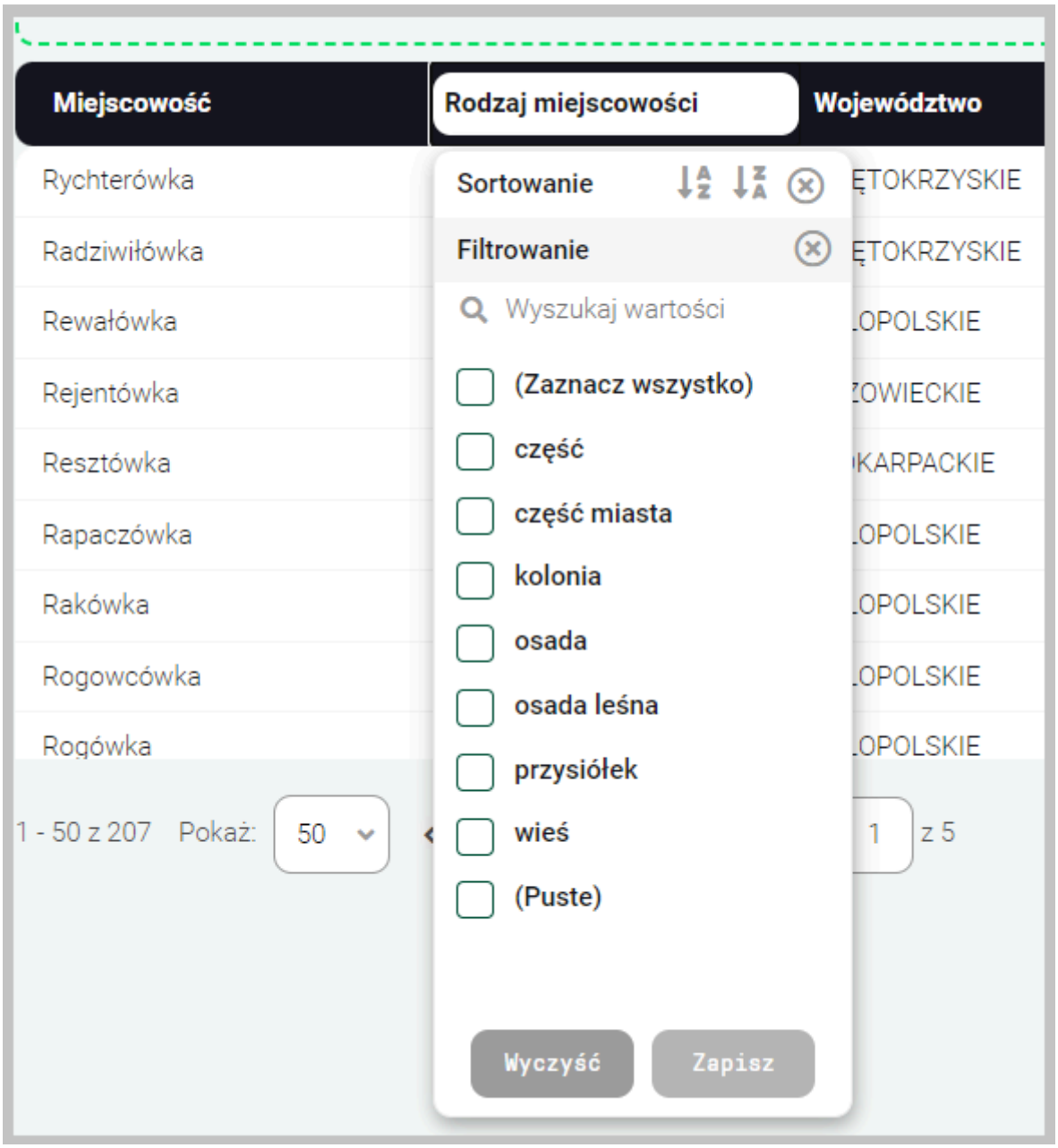
- pole wyszukiwania; służy do wyszukiwania wpisanego tekstu w kolumnach listy z danymi typu tekstowego; działa jak filtr, tj. wyświetla tylko rekordy, w których występuje szukana fraza; znalezione wystąpienia sa podświetlone na żółto.

> **WSKAZÓWKA**
> Wyszukiwarka jest dostępna, o ile w definicji listy na karcie `Wygląd` włączono przełącznik `Wyświetlaj pole wyszukiwania`.

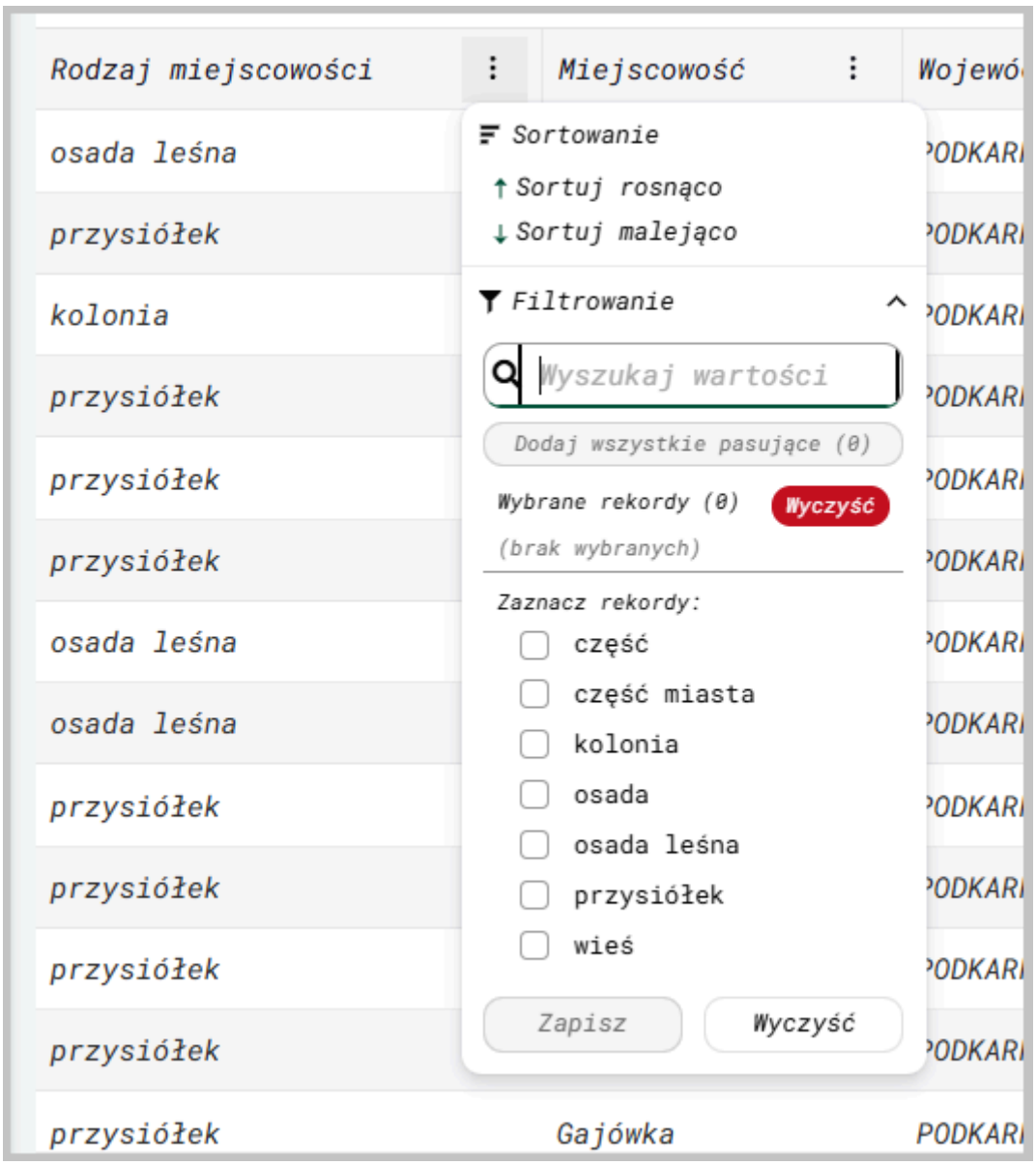
- pole filtrowania: w polu tekstowym jest wyświetlana nazwa zastosowanego filtru, na liście rozwijanej znajdują się zdefiniowane filtry oraz polecenie `Ustawienia`, które wyświetla panel `Filtrowanie zaawansowane`. Na panelu można zarządzać zdefiniowanymi filtrami i definiować nowe, pod warunkiem posiadania uprawnienia PBA `Tworzenie i edycja filtrów SQL`.
- przycisk odświeżania z listą rozwijaną, z której można wybrać następujące polecenia:
  - `Eksport do MS Excel`: powoduje zapisanie w folderze pobierania przeglądarki pliku w formacie *xlsx* z rekordami wyświetlanymi na liście.
  - `Widoczność kolumn`: wyświetla panel z listą aktywnych kolumn i umożliwia zarządzanie ich widocznością. Ustawienia widoczności można zapisać w ramach konfiguracji użytkownika. Opcja jest dostępna dla list, dla których w Kreatorze listy na karcie `Ustawienia listy` włączono przełącznik `Widoczność listy`.
  - `Grupowanie kolumn`: wyświetla panel do zarządzania grupowaniem rekordów przy użyciu klawiatury; uzupełnia możliwość grupowania rekordów przez przeciągnięcie nagłówka kolumny do ramki z tekstem `Przeciągnij nagłówek kolumny i upuść go tutaj, aby pogrupować według tej kolumny` wyświetlanego poniżej panelu górnego. Grupowanie jest możliwe tylko wtedy, gdy w definicji listy włączono opcję grupowania według wartości danej kolumny.
  - `Konfiguracje listy`: wyświetla panel `Konfiguracje listy`, na którym można zarządzać konfiguracjami listy. Konfiguracja obejmuje ustawienia szerokości kolumn, filtrowania, sortowania i grupowania.
  - `Wyczyść wszystkie filtry`: resetuje wszystkie zastosowane ustawienia filtrowania.
  - `Pomoc`: wyświetla tekst pomocy dla listy, o ile został przygotowany przez konsultanta; patrz także Leksykon nAxiom .

Pod panelem górnym i ramką do grupowania wyświetlane są nagłówki kolumn z nazwami kolumn. Domyślnie nazwy kolumn widoku listy są takie same jak nazwy kolumn zwracane zapytaniem, które jest źródłem danych dla tej listy, jednak można to zmienić w trybie projektowania listy. Jeśli dla listy włączono obsługę sortowania i/lub filtrowania, kliknięcie

nagłówka kolumny powoduje wyświetlenie panelu sortowania/filtrowaniakliknięcie nagłówka kolumny powoduje posortowanie wartości w kolumnie, a kliknięcie ikony z trzema kropkami wyświetla panel sortowania/filtrowania.



Panel filtrowania/sortowania



Panel filtrowania/sortowania

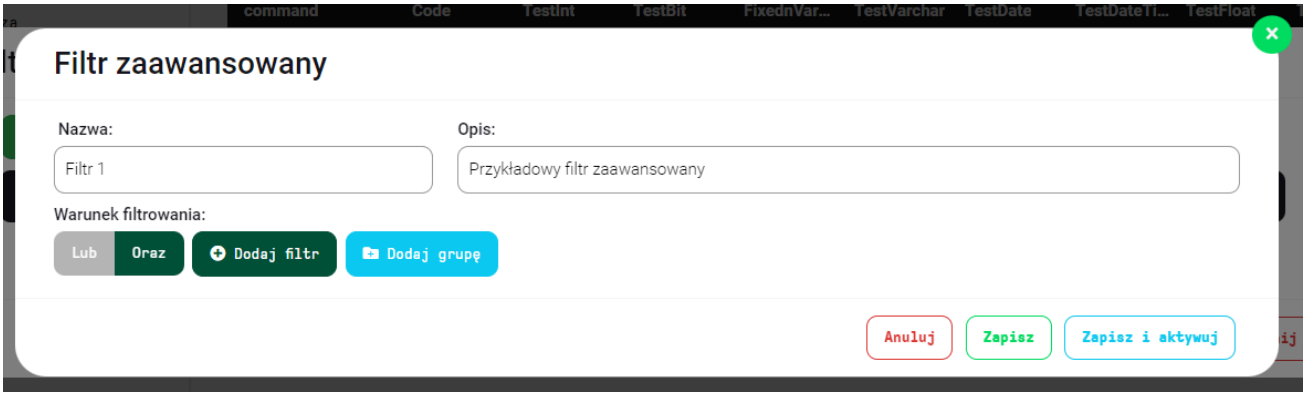
Ustawienia filtrowania zależą od typu danych i konfiguracji filtrowania określonej przez konsultanta w definicji widoku listy. Po określeniu kryteriów filtrowania należy kliknąć przycisk `Zapisz` u dołu panelu, aby zastosować filtr. Przycisk `Wyczyść` służy do usunięcia filtru.

Pod nagłówkami kolumn znajduje się tzw. *grid*, czyli sama lista z danymi. Oprócz danych lista może także zawierać tzw. *kolumnę akcji*, to jest kolumnę zawierającą przyciski akcji, w tym na przykład przycisk edycji, który wyświetla bieżący rekord danych w formularzu. Kolumna akcji można także zawierać przycisk usuwania rekordu oraz przyciski do edycji danych *inline*, to jest bezpośrednio w widoku listy.

Na samym dole znajduje się *stopka listy*. W stopce znajdują się przyciski przechodzenia do: pierwszej strony listy, poprzedniej strony listy, następnej strony listy i ostatniej strony listy, dodatkowo wyświetlany jest numer bieżącej strony. Z prawej strony wyświetlane są numery

rekordów na bieżącej stronie listy. Ponadto w stopce mogą być wyświetlane wartości funkcji agregujących.

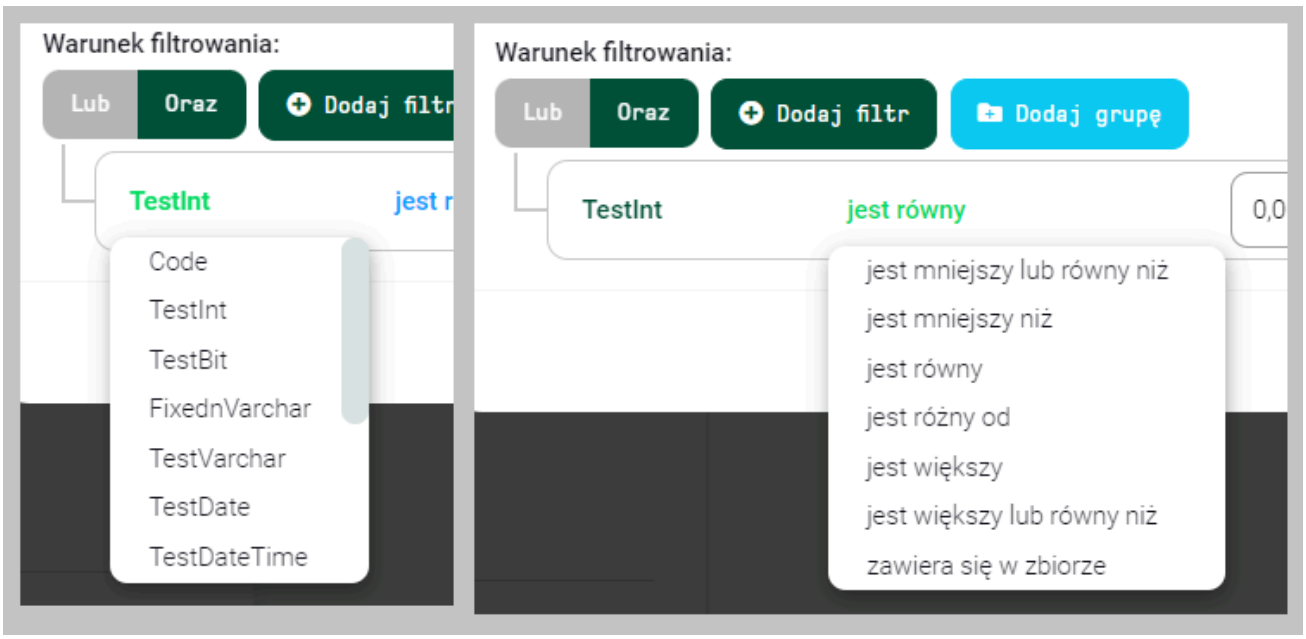
Po wybraniu z listy predefiniowanych filtrów pozycji `Ustawienia` i kliknięciu w wyświetlonym oknie przycisku `dodaj nowy filtr` zostaje wyświetlone okno dialogowe `Filtr zaawansowany` `Nowy filtr`, w którym można skonfigurować nowy filtr.



Filtr zaawansowany

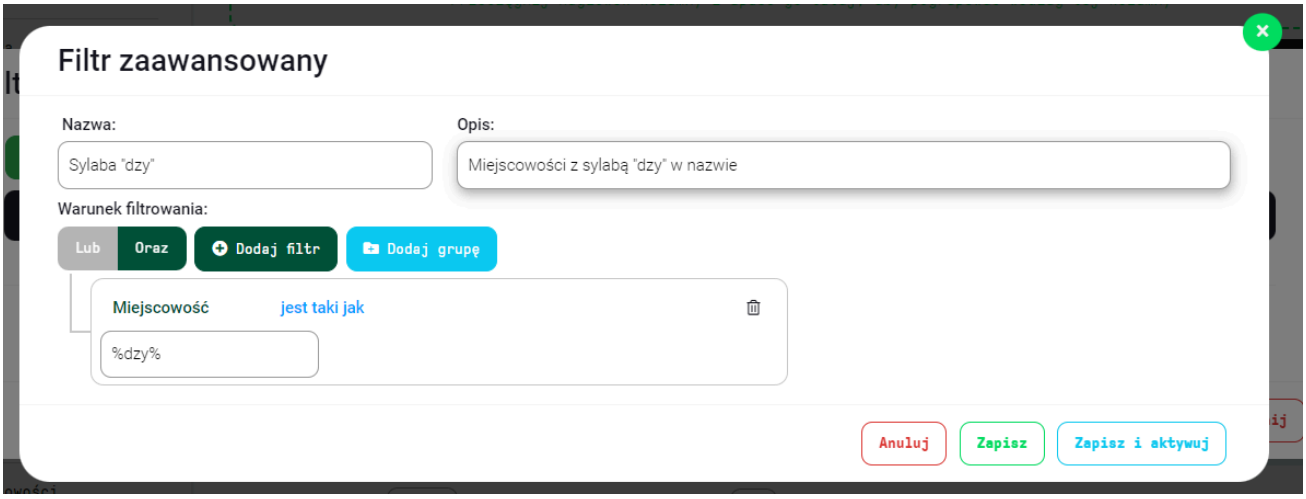
Filtr zaawansowany

Po wpisaniu nazwy i opisu należy kliknąć przycisk `Dodaj filtr` i skonstruować warunek filtrowania. Konstrukcja polega na wybraniu nazwy kolumny, operatora i wpisaniu wartości.



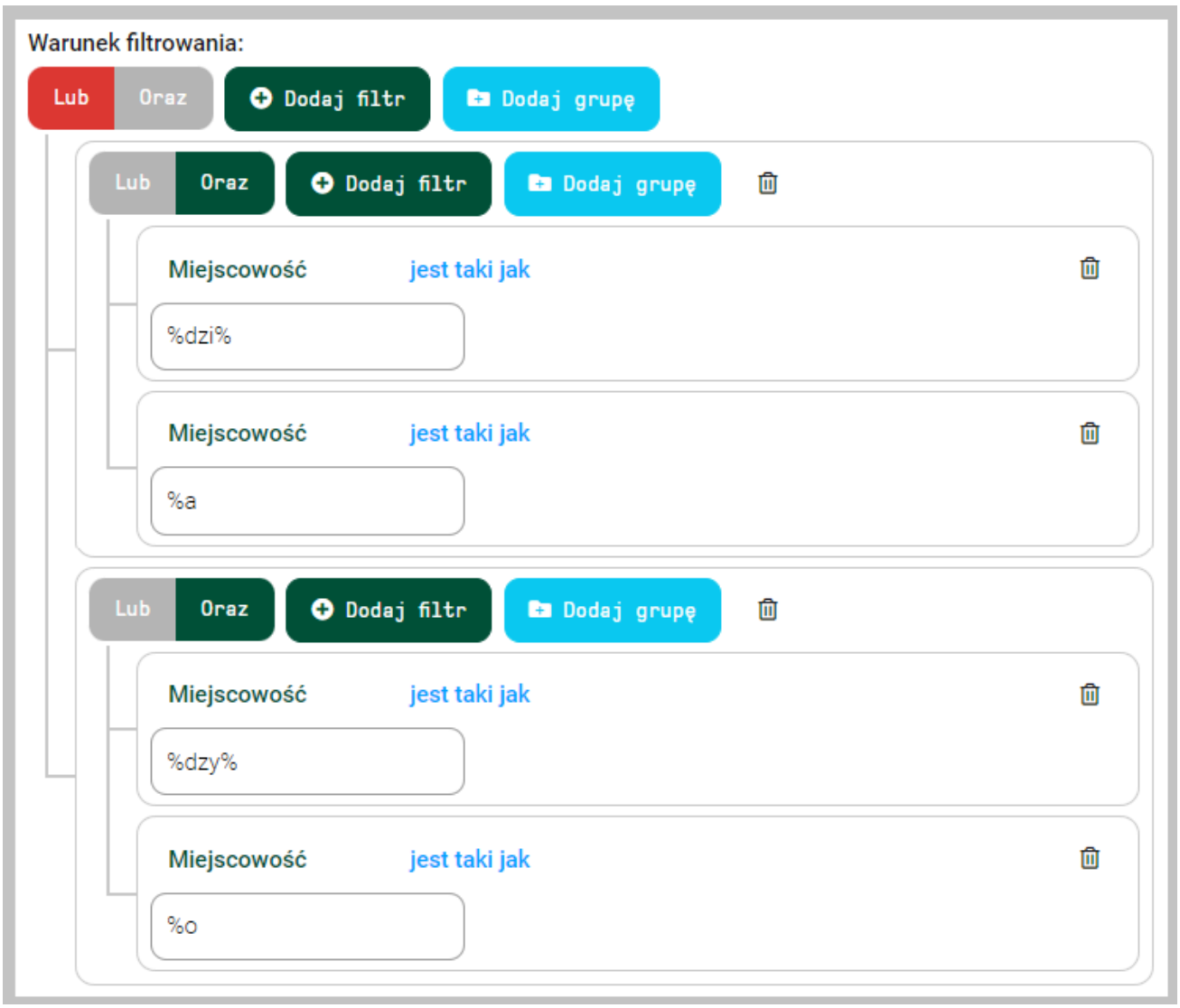
Definiowanie filtru

Dostępne operatory zależą od typu danych wybranej kolumny. Dla danych tekstowych dostępny jest operator `Jest taki jak`. Pozwala on filtrować wartości tekstowe, które zawierają podany ciąg znaków. w takim przypadku jako symbolu zastępczego pozostałej części tekstu należy użyć symbolu %.



Filtrowanie wartości tekstowych

Filtry zaawansowane mogą się składać z kilku warunków połączonych operatorem logicznym *LUB* albo *ORAZ*, a także tworzyć grupy warunków również połączonych jednym z dostępnych operatorów.



Filtrowanie wartości tekstowych

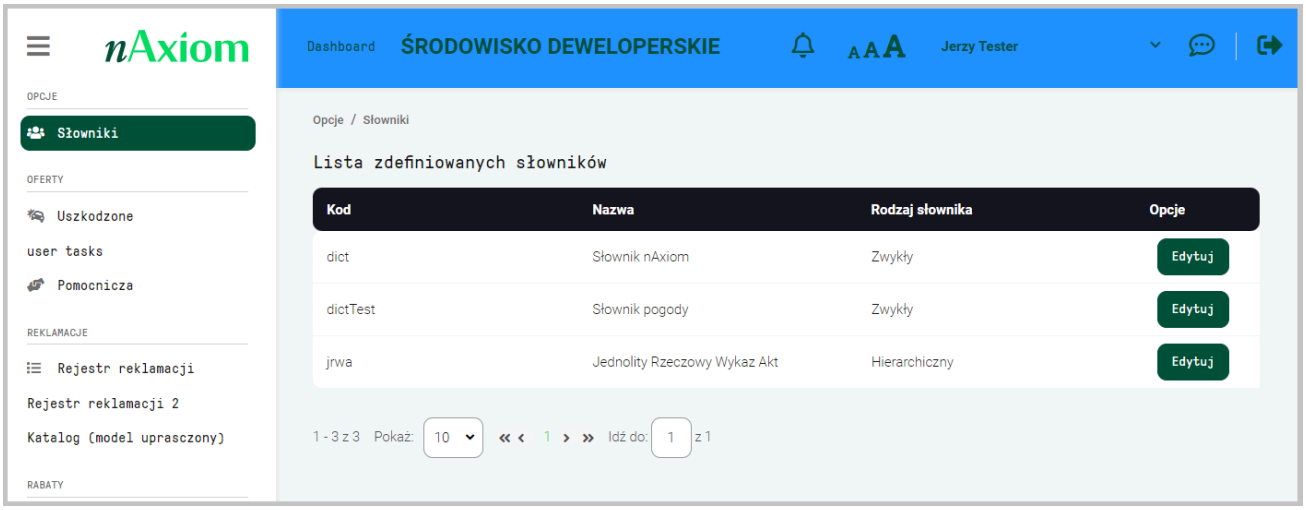
10.6. OBSŁUGA SŁOWNIKÓW

Podsumowanie:
Słowniki to zbiory predefiniowanych wartości tekstowych, które są używane jako źródło danych dla kontrolek typu lista wyboru. Ponadto pozycji słownika można używać do filtrowania danych w widokach listy. Ważną cechą słowników jest to, że ich zawartość może być definiowana przez uprawnionych użytkowników aplikacji biznesowych po stronie *FrontSPA* bez udziału konsultantów.

☛ *FrontSPA* > `OPCJE` > `Słowniki`

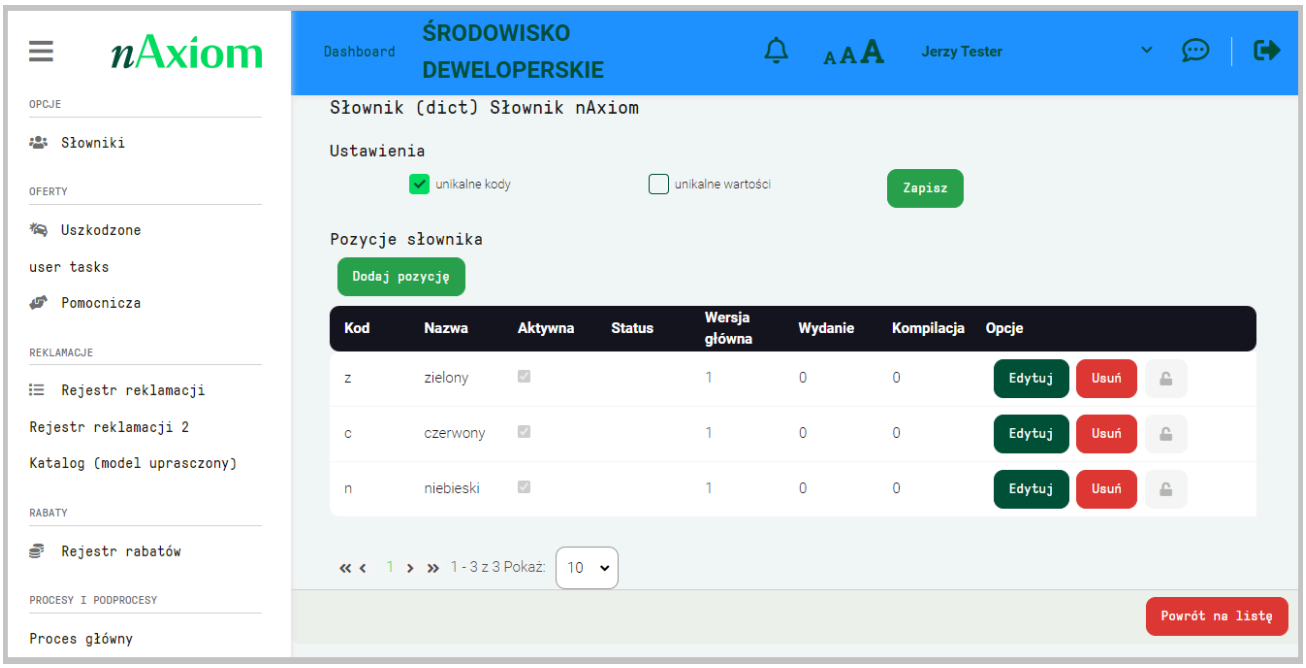
Aby użytkownik końcowy miał dostęp do obsługi pozycji słownika, musi mieć przypisaną rolę biznesową, która będzie miała dostęp do pozycji menu `Słowniki` (*AdminSPA* > `ADMINISTRACJA` > `Role biznesowe` > *NazwaRoli* > `Uprawnienia`).

Po kliknięciu polecenia `Słowniki` w sekcji `OPCJE` w menu głównym *FrontSPA* zostanie wyświetlona lista słowników.



Lista słowników

Kliknięcie przycisku `Edytuj` obok wybranego słownika wyświetla listę pozycji tego słownika.



Lista pozycji słownika

W wierszu pozycji są dostępne przyciski `Edytuj` i `Usuń` oraz ikona kłódki, która służy do włączania/wyłączania ochrony danej pozycji przed nadpisywaniem w przypadku zaimportowania nowej wersji aplikacji do bieżącego środowiska.

Widoczny u góry przycisk `Dodaj pozycję` powoduje dodanie nowego wiersza na liście pozycji, w którym w trybie inline należy wpisać:

- `Kod`: kod tekstowy pozycji (tylko znaki alfanumeryczne),

- `Nazwa`: wartość tekstowa pozycji słownika,
- `Aktywna`: status pozycji (domyślnie włączone),
- *Nazwa kolumny*: opisowe kolumny (maks. 3) dla pozycji słownika podane w definicji słownika
- *Pole numeryczne*: pomocnicze kolumny z wartością liczbową (maks. 3) dla pozycji słownika podane w definicji słownika

Aby dodać pozycję do słownika, kliknij przycisk `Dodaj` z prawej strony.

Nad listą pozycji są dostępne dwie opcje:

- `Unikalne kody`: zaznacz to pole wyboru, aby dla słownika nie można było dodać dwóch pozycji o takim samym kodzie („kod" jest systemowym atrybutem pozycji słownika — kolumna *Code* w tabeli *core.DictionaryValues*).
- `Unikalne wartości`: zaznacz to pole wyboru, aby dla słownika nie można było dodać dwóch pozycji o takiej samej wartości („wartość" jest systemowym atrybutem pozycji słownika — kolumna *Name* w tabeli *core.DictionaryValues*). button-cancel: |- `Anuluj`: kliknij, aby zamknąć okno i porzucić zmiany.

Do ich zmiany wymagane jest uprawnienie PBA: `Aktualizacja definicji aplikacji oraz jej obiektów/modułów` (*AdminSPA* > `ADMINISTRACJA` > `Lista aplikacji` > *NazwaAplikacji* > `Edytuj` > `Uprawnienia`).

Powyższy opis dotyczył słownika zwykłego (prostego). W nAxiom można także definiować słowniki hierarchiczne, to jest takie, których pozycje mogą zawierać pozycje podrzędne. W przypadku słownika hierarchicznego w wierszu pozycji jest dostępny przycisk z symbolem +, którym można dodać pozycję podrzędną. Ponadto dla pozycji słownika hierarchicznego nie jest obsługiwany status aktywności.

Lista pozycji słownika hierarchicznego

10.7. STRUKTURA ORGANIZACYJNA

Podsumowanie:
Obsługa struktury organizacyjnej odbywa się w aplikacji *FrontSPA*. Do tej czynności uprawnieni są posiadacze roli, w której uprawnieniach włączono pozycję menu `Struktura organizacyjna` (`ADMINISTRACJA` > `Role biznesowe` > `Uprawnienia`).

Tworzenie struktury organizacyjnej należy rozpocząć od utworzenia nowej jednostki głównej. W strukturze organizacyjnej może występować wiele jednostek głównych, a każda z nich może mieć wiele jednostek podrzędnych.

NOWA JEDNOSTKA GŁÓWNA

1. Kliknij polecenie `Struktura organizacyjna` w sekcji `USTAWIENIA SYSTEMOWE` we FrontSPA.
   Zostanie wyświetlona lista `Struktura organizacyjna`.

2. Kliknij przycisk `Nowa jednostka główna` z lewej strony u góry.
   Zostanie wyświetlony pusty wiersz na liście z możliwością edycji pól:
   - `Nazwa`: nazwa jednostki.
   - `Kod` (`Kod jednostki`): ciąg znaków identyfikujących jednostkę organizacyjną; kod jednostki może składać się ze znaków alfanumerycznych (cyfry i litery alfabetu łacińskiego) oraz wybranych znaków specjalnych: kropka (.), podkreślenie (_), myślnik (-) i ukośnik (/); znaków specjalnych

można używać w kodach liczących co najmniej trzy znaki; dozwolone jest definiowanie kodów jedno- i dwuznakowych złożonych ze znaków alfanumerycznych.
    - `Obowiązuje od`: data obowiązywania jednostki organizacyjnej w strukturze; domyślnie ustawiana jest data bieżąca, ale można ją zmienić na późniejszą.
    - `Obowiązuje do`: data końca obowiązywania jednostki organizacyjnej w strukturze; to pole jest wypełniane automatycznie po usunięciu jednostki organizacyjnej.

3. Wypełnij poszczególne pola i kliknij przycisk (`Zapisz`). Jednostka główna zostanie dodana.

W wierszu dodanej jednostki znajdują się przyciski (`Nowa podrzędna jednostka`) i (`Edycja`), które umożliwiają odpowiednio dodanie podrzędnej jednostki organizacyjnej oraz przejście do edycji jednostki organizacyjnej.

NOWA JEDNOSTKA PODRZĘDNA

Aby dodać jednostkę podrzędną, kliknij przycisk z symbolem + obok jednostki nadrzędnej, a następnie wpisz dane, tak samo jak dla jednostki głównej.

Po utworzeniu jednostki w strukturze organizacyjnej przejdź do jej edycji i uzupełnij dane tej jednostki. W oknie dialogowym `Edycja jednostki organizacyjnej` właściwości jednostki podzielono na trzy karty:

- `Dane podstawowe`: dane identyfikacyjne i daty obowiązywania.
- `Dane dodatkowe`: karta utworzona z formularza, którego identyfikator podano w ustawieniach systemu; patrz temat podanego wygenerowanego automatycznie utworzonego podczas generowania elementów aplikacji dla tabeli jednostek organizacyjnych.
- `Użytkownicy`: lista użytkowników przypisanych do danej jednostki organizacyjnej.

Ustawienia na tych kartach opisano poniżej. Wypełnij pola na poszczególnych kartach zgodnie z opisem w kolejnych sekcjach i kliknij przycisk (`Zapisz i zamknij`). Okno dialogowe zostanie zamknięte a zmiany jednostki organizacyjnej zostaną zapisane.

W oknie dialogowym `Edycja jednostki organizacyjnej` jest również dostępny przycisk `Usuń`. Po jego kliknięciu i potwierdzeniu wyświetlonego komunikatu status jednostki organizacyjnej zmieni się z `Aktywna` na `Usunięta`, a w pole `Obowiązuje do` zostanie wpisana data bieżąca. Taka jednostka będzie widoczna w strukturze, ale nie będzie

można jej edytować, przypisywać do niej użytkowników ani definiować szablonów uprawnień.

DANE PODSTAWOWE JEDNOSTKI ORGANIZACYJNEJ

Na karcie `Dane podstawowe` można określić następujące dane jednostki organizacyjnej:

- `Kod` (`Kod jednostki`): ciąg znaków identyfikujących jednostkę organizacyjną; kod jednostki może składać się ze znaków alfanumerycznych (cyfry i litery alfabetu łacińskiego) oraz wybranych znaków specjalnych: kropka (.), podkreślenie (_), myślnik (-) i ukośnik (/); znaków specjalnych można używać w kodach liczących co najmniej trzy znaki; dozwolone jest definiowanie kodów jedno- i dwuznakowych złożonych ze znaków alfanumerycznych.
- `Nazwa jednostki`: pełna nazwa jednostki.
- `OUID`: identyfikator jednostki organizacyjnej (ang. **O**rganizational **U**nit) w usłudze katalogowej.
- `Data od`: data włączenia jednostki organizacyjnej w strukturę organizacyjną; jeśli w tym polu zostanie wpisana data przyszła, jednostka organizacyjna będzie mieć status `Nieaktywna`.
- `Data do`: data końcowa funkcjonowania danej jednostki organizacyjnej w strukturze organizacyjnej; w przypadku usunięcia jednostki organizacyjnej w tym polu automatycznie wpisywana jest data bieżąca.
- `Status`: status jednostki organizacyjnej; dostępne wartości:
  - `Nowa`
  - `Aktywna`
  - `Nieaktywna`
  - `Archiwalna`
  - `Usunięta`

DANE DODATKOWE JEDNOSTKI ORGANIZACYJNEJ

Karta `Dane dodatkowe` to formularz o identyfikatorze podanym w ustawieniach systemu. Na tym formularzu można wyłączyć pasek toolbar, ponieważ jego przyciski są nieaktywne, zapis danych na tej karcie obsługują przyciski u dołu formularza edycji jednostki organizacyjnej.

UŻYTKOWNICY JEDNOSTKI ORGANIZACYJNEJ

Na karcie `Użytkownicy` są wyświetlani użytkownicy dodani do tej jednostki. Dane pochodzą z tabeli *core.OUInstanceUsers*. Na tej karcie znajdują się następujące pola i przyciski:

- `Dodaj`: wyświetla okno dialogowe `Lista użytkowników`, w którym można wybierać użytkowników, zaznaczając pola wyboru z lewej strony poszczególnych wierszy; po zaznaczeniu użytkowników kliknij przycisk `Wybierz`, aby przypisać ich do danej jednostki organizacyjnej.
- `Usuń`: (przycisk z prawej strony w wierszu użytkownika) pozwala usunąć użytkownika z jednostki organizacyjnej.

WSKAZÓWKA
W ustawieniach systemu w aplikacji *AdminSPA* można skonfigurować własne zapytania dla listy dostępnych użytkowników, własne zapytanie dla listy użytkowników w bieżącej jednostce organizacyjnej oraz włączyć własne ustawienia listy wyświetlanej na karcie `Użytkownicy`.

Powiązane tematy:
- Struktura organizacyjna, informacje ogólne
- Dane dodatkowe jednostek struktury organizacyjnej
- Ustawienia systemu, Struktura organizacyjna

10.8. AKCJE ZDARZEŃ SYSTEMOWYCH

Podsumowanie:
W witrynie nAxiom można definiować zestawy akcji wykonywanych po zalogowaniu. Takie scenariusze można definiować dla indywidualnych użytkowników i/lub dla ról biznesowych.

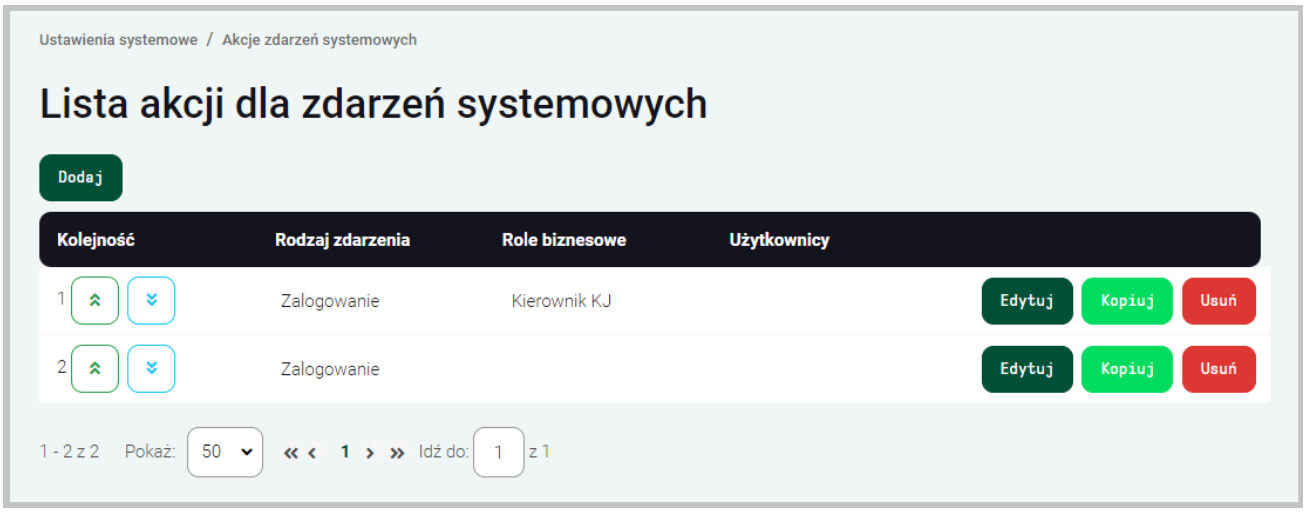
☛ *FrontSPA* > `Ustawienia systemowe` > `Akcje zdarzeń systemowych`

Aby użytkownik mógł konfigurować akcje wykonywane po zalogowaniu musi mieć przypisane uprawnienia PBA:

- `Akcje zdarzeń systemowych`: *FrontSPA* > `Uprawnienia` > *użytkownik/rola* > `Użytkowanie aplikacji` > `Przegląd aplikacji w module AdminSPA`
- `Przegląd aplikacji w module AdminSPA`: *FrontSPA* > `Uprawnienia` >

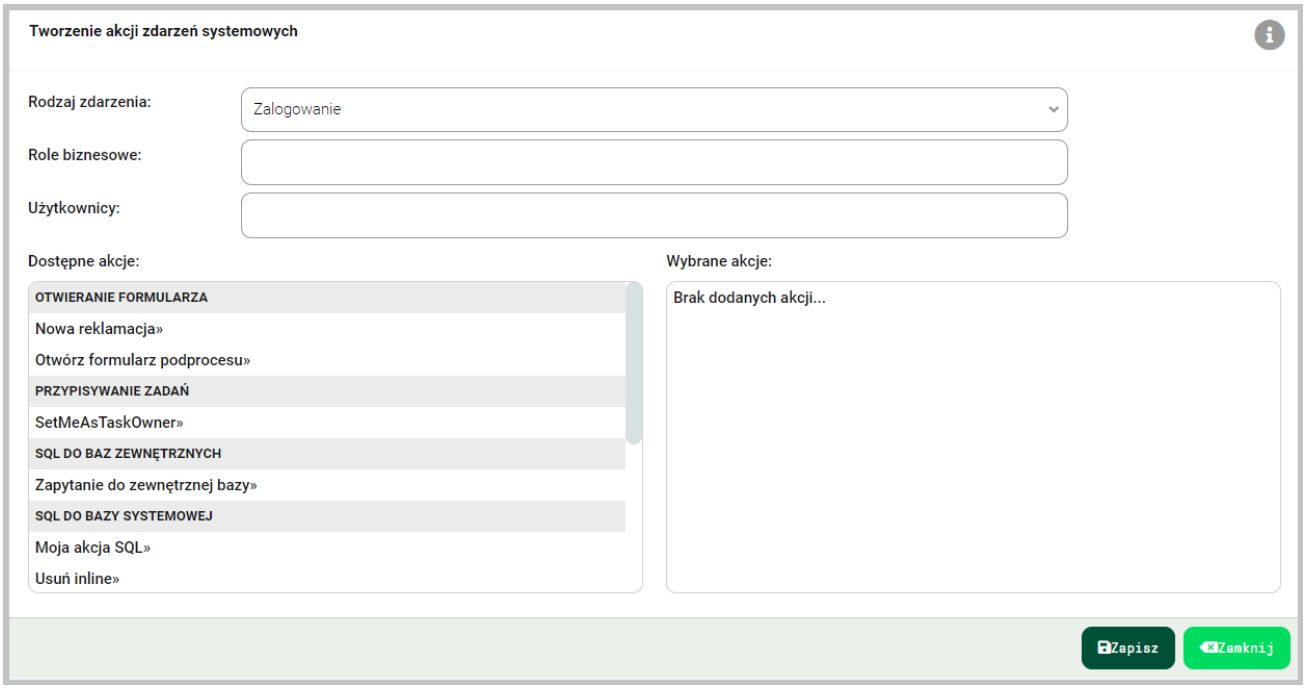
użytkownik/rola > `Budowanie aplikacji` > `Przegląd aplikacji w module AdminSPA`.

Po wybraniu polecenia wyświetlana jest lista akcji, na której można edytować, kopiować i usuwać zdefiniowane akcje (sekwencje akcji), a także zmieniać ich kolejność.



Lista akcji dla zdarzeń systemowych

Kliknij przycisk `Dodaj`, aby dodać nowy scenariusz. Zostanie wyświetlony formularz `Tworzenie akcji zdarzeń systemowych`.



Lista akcji dla zdarzeń systemowych

Wypełnij kolejno pola:

- `Rodzaj zdarzenia`: jedyne obsługiwane obecnie zdarzenie to `Logowanie`.
- `Role biznesowe`: wybierz role biznesowe (jedną lub kilka), dla których będzie

wykonywany scenariusz.

- `Użytkownicy`: wybierz użytkowników (jednego lub kilku), dla których będzie wykonywany scenariusz.
- `Dostępne akcje`: kliknij dwukrotnie akcję, aby przenieść ją na listę `Wybrane akcje`; na liście dostępne są następujące rodzaje akcji: SQL do bazy systemowej i do baz zewnętrznych, otwieranie formularza, wysyłanie/odbieranie e-mail, przypisywanie zadań.
- `Wybrane akcje`: akcje dodane do tej listy zostaną wykonane po zalogowaniu wskazanych użytkowników; aby usunąć akcję z tej listy, kliknij ją dwukrotnie.

W oknie definiowania scenariusza nie ma możliwości edytowania akcji, ani tworzenia nowych.

Zasady wykonywania scenariuszy są następujące:

- wykonywany jest zawsze tylko jeden scenariusz
- wykonywany jest scenariusz „imienny" dla logującego się użytkownika, o ile został zdefiniowany
- wykonywany jest pierwszy na liście scenariusz zdefiniowany dla jednej z ról użytkownika; kolejność można zmieniać strzałkami w kolumnie `Kolejnosć`
- scenariusze zdefiniowane dla kilku ról dotyczą użytkowników z przypisaną dowolną z tych ról
- jeśli dla logującego się użytkownika nie ma zdefiniowanego scenariusza, po zalogowaniu jest wyświetlany pulpit

Dla każdego scenariusza dostępne są przyciski `Edytuj`, `Kopiuj` i `Usuń`. Skopiowany scenariusz jest wyświetlany na końcu listy.

Powiązane tematy:

- Uprawnienia PBA, obszar Użytkowanie aplikacji
- Uprawnienia PBA, obszar Budowanie aplikacji

10.9. ZASTĘPSTWA

Podsumowanie:
Funkcjonalność zastępstw pozwala wyznaczyć użytkownika, który będzie zastępował innego użytkownika podczas jego nieobecności. Podjęcie pracy w zastępstwie powoduje zmianę wartości atrybutu *UserId* bieżącego użytkownika na odpowiadającą zastępowanemu. Rzeczywistą tożsamość zastępującego określa atrybut *NativeUserId*. Pozwala to zachować kontrolę nad pracą w kontekście innego użytkownika, a jednocześnie przenieść na zastępującego wszelkie uprawnienia zastępowanego, które są weryfikowane na podstawie atrybutu *UserId*.

Praca w zastępstwie oznacza pracę wyłącznie z uprawnieniami zastępowanego. Obejmuje to:

- role biznesowe — globalne i w jednostkach organizacyjnych
- dostęp do menu i aplikacji
- przynależność do grup interesariuszy
- tytuł twórcy dokumentu
- uprawnienia do akceptacji
- uprawnienia ACL do dokumentów
- uprawnienia ACL formularzy i ustawienia dostępności oparte na rolach
- uprawnienia PBA.

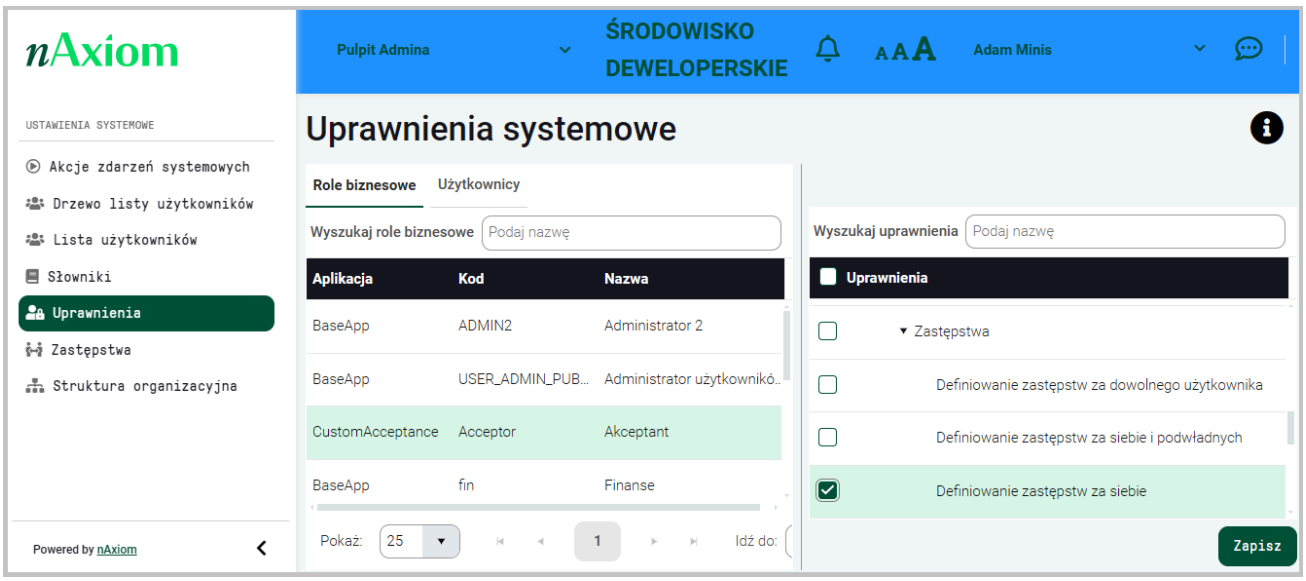
☛ *FrontSPA* > `Zastępstwa`

W TYM TEMACIE

- Zarządzanie zastępstwami
- Praca w zastępstwie

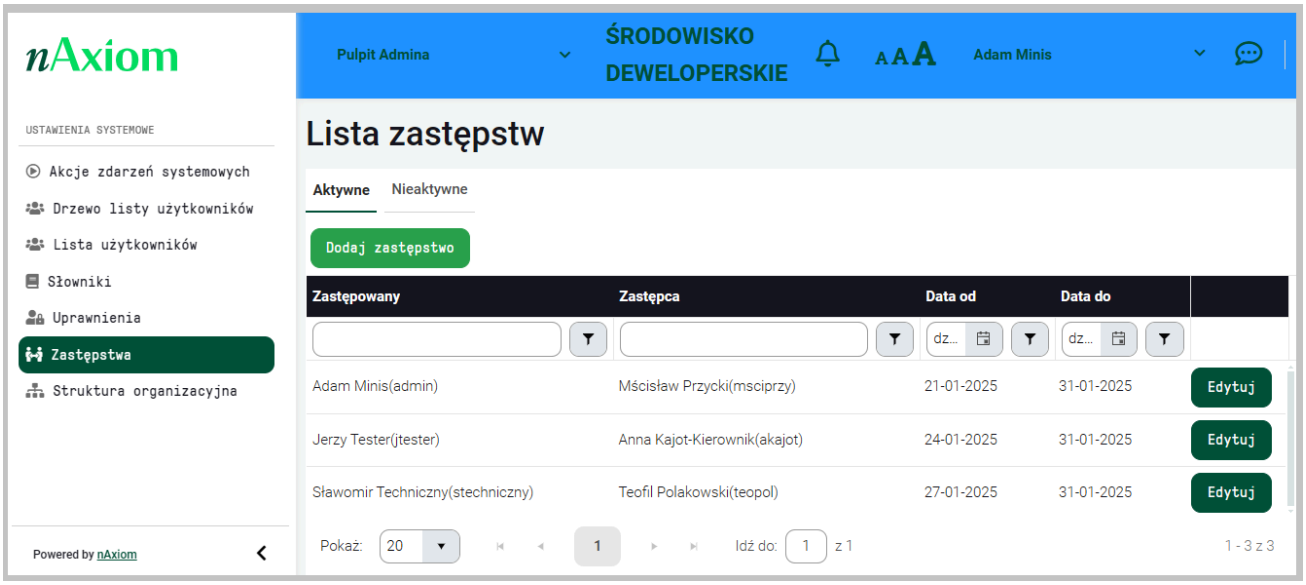
ZARZĄDZANIE ZASTĘPSTWAMI

Zastępstwa definiuje się w aplikacji *FrontSPA* po wybraniu polecenia `Zastępstwa` z menu `USTAWIENIA SYSTEMOWE`. Dostęp do tego polecenia zależy od posiadanych uprawnień PBA. Domyślnie nowo tworzony użytkownik nie ma żadnych uprawnień do definiowania zastępstw.



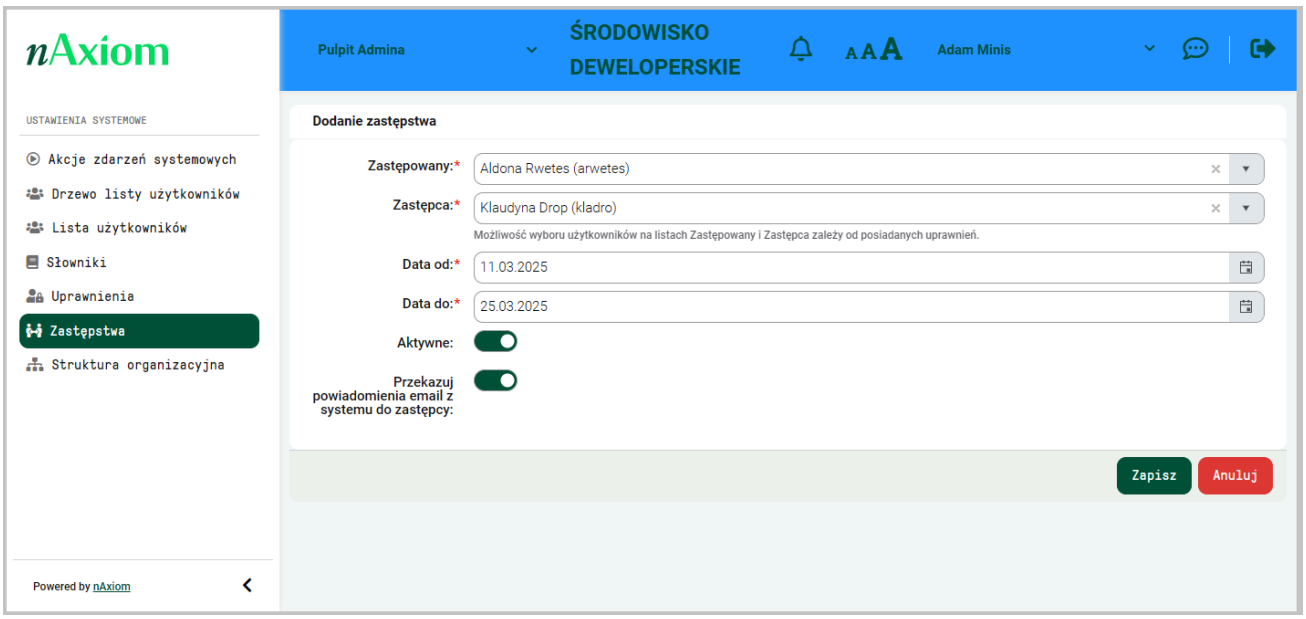
Uprawnienia PBA do zastępstw

Po wybraniu polecenia `Zastępstwa` jest wyświetlana lista aktywnych zastępstw odpowiednio do posiadanych uprawnień (zastępstwa za siebie, za podwładnych, wszystkie zastępstwa). Aktywne zastępstwa to zastępstwa, które trwają lub mają przyszłą datę rozpoczęcia. Na karcie `Nieaktywne` są wyświetlane zastępstwa zakończone i takie, które zostały dezaktywowane w trakcie trwania.



Lista zastępstw

Aby dodać nowe zastępstwo, kliknij przycisk `Dodaj zastępstwo` i wypełnij odpowiednie pola w wyświetlonym oknie dialogowym.



Dodanie zastępstwa

`Zastępowany`: (domyślnie bieżący użytkownik) wybierz z listy osobę zastępowaną.

`Zastępujący`: wybierz z listy osobę zastępującą.

> **WSKAZÓWKA**
> Na listach wyboru użytkowników nie ma użytkowników z dezaktywowanym lub zablokowanym profilem użytkownika. Zastępstwa za użytkowników, dla których upłynął termin ważności profilu, można definiować zależnie od ustawień systemowych (sekcja `Zastępstwa`).

`Data od`: data rozpoczęcia zastępstwa (od początku doby); może być datą przeszłą lub przyszłą (domyślnie bieżąca).

`Data do`: data zakończenia zastępstwa (do końca doby); może być datą bieżącą lub przyszłą (domyślnie bieżąca).

`Aktywne`: domyślnie włączone; pozwala dezaktywować zastępstwo; w przypadku dezaktywacji trwającego zastępstwa wyświetlany jest komunikat ostrzegawczy.

> **WAŻNE**
> Dezaktywacja zastępstwa w przypadku, gdy użytkownik pracuje w tym zastępstwie spowoduje, że zniknie informacja o pracy w zastępstwie oraz przycisk kończenia pracy w zastępstwie w menu rozwijanym profilu użytkownika. Jednak w dalszym ciągu będzie to praca w zastępstwie. Jeśli zastępstwo zostanie dezaktywowane przez użytkownika, który właśnie pracuje w tym zastępstwie, tryb pracy w zastępstwie zostanie zakończony automatycznie.

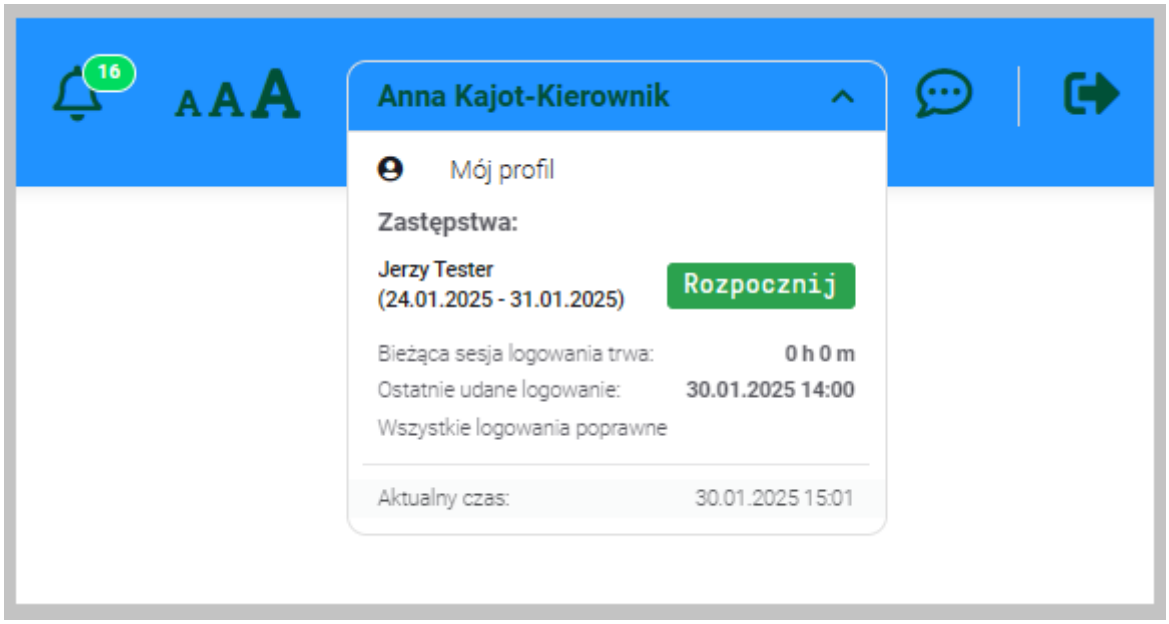
`Przekazuj powiadomienia email z systemu do zastępcy`: domyślnie włączone; powoduje przekazywanie kopii wiadomości e-mail wysyłanych z nAxiom (akcja wysyłki e-mail, eskalatory) na adres e-mail osoby zastępującej. Z przekazywania wyłączone są wiadomości e-mail z linkiem do zresetowania hasła. W przypadku akcji e-mail, w jej definicji musi być włączony przełącznik `Obsługa zastępstw`.

Zastępstwa można edytować. W przypadku zastępstw, które są w toku (upłynęła data rozpoczęcia), można edytować tylko datę zakończenia, tj. przedłużyć zastępstwo. Dla przyszłych zastępstw można edytować wszystkie dane. W ramach edycji można także dezaktywować lub aktywować zastępstwo.

> **WAŻNE**
> Podczas definiowania zastępstwa nie jest sprawdzane, czy w danym okresie zastępujący będzie obecny (czy nie jest w tym samym czasie zastępowany) ani czy nie zastępuje on w tym okresie innego użytkownika. System nie obsługuje „dziedziczenia" zastępstw.

PRACA W ZASTĘPSTWIE

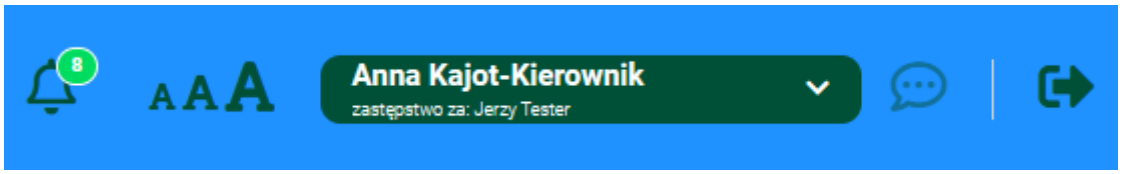
Aby podjąć pracę w zastępstwie, należy rozwinąć menu profilu użytkownika i kliknąć przycisk `Rozpocznij` obok imienia i nazwiska zastępowanej osoby. Po potwierdzeniu komunikatu następuje przełączenie wartości *UserId* i przełączenie na pulpit osoby zastępowanej.



Podejmowanie pracy w zastępstwie

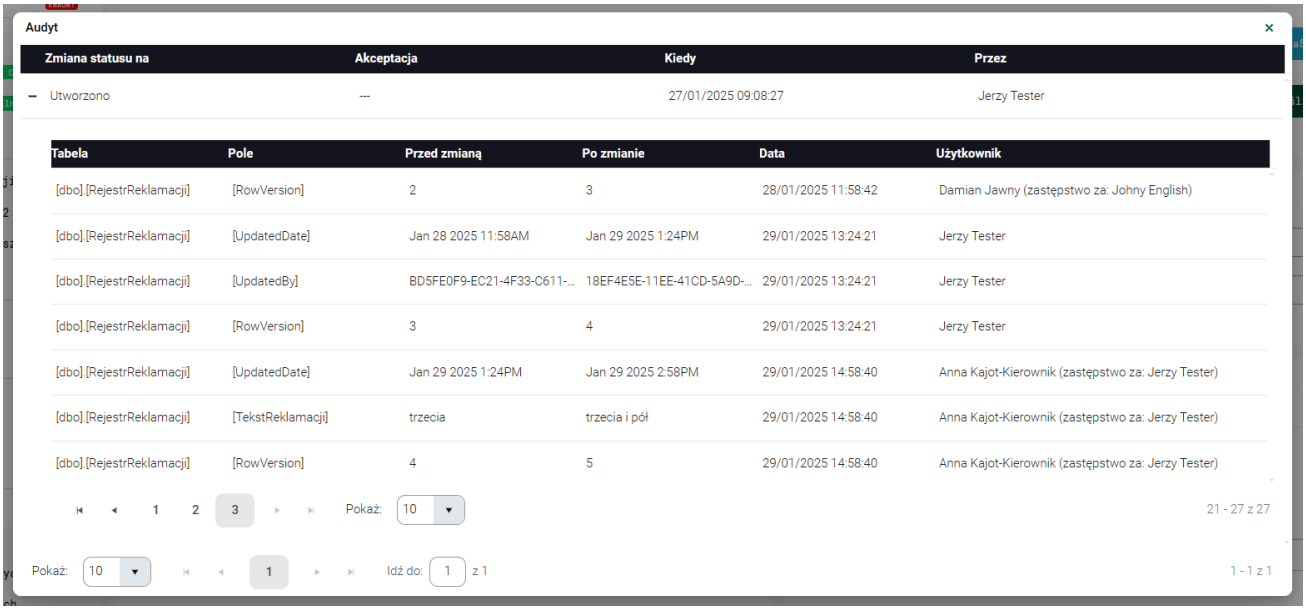
Widoczne menu i aplikacje odpowiadają rolom i uprawnieniom PBA zastępowanego, a dostęp do dokumentów na listach i formularzach jest wyliczany na podstawie uprawnień ACL zastępowanego. Całe środowisko witryny nAxiom wygląda tak, jak je widzi użytkownik

zastępowany. Jedyne widoczne różnice to dopisek o pracy w kontekście innego użytkownika pod danymi zalogowanego użytkownika oraz brak dostępu do funkcji czatu.



Praca w zastępstwie

Poza tym w audycie danych biznesowych widoczne są informacje o pracy w zastępstwie. Odpowiednie adnotacje są także widoczne w komentarzach dodawanych do dokumentów.



Audyt

Zmiana statusu na	Akceptacja	Kiedy	Przez
Utworzono	---	27/01/2025 09:08:27	Jerzy Tester

Tabela	Pole	Przed zmianą	Po zmianie	Data	Użytkownik
[dbo].[RejestrReklamacji]	[RowVersion]	2	3	28/01/2025 11:58:42	Damian Jawny (zastępstwo za: Johny English)
[dbo].[RejestrReklamacji]	[UpdatedDate]	Jan 28 2025 11:58AM	Jan 29 2025 1:24PM	29/01/2025 13:24:21	Jerzy Tester
[dbo].[RejestrReklamacji]	[UpdatedBy]	BD5FE0F9-EC21-4F33-C611-...	18EF4E5E-11EE-41CD-5A9D-...	29/01/2025 13:24:21	Jerzy Tester
[dbo].[RejestrReklamacji]	[RowVersion]	3	4	29/01/2025 13:24:21	Jerzy Tester
[dbo].[RejestrReklamacji]	[UpdatedDate]	Jan 29 2025 1:24PM	Jan 29 2025 2:58PM	29/01/2025 14:58:40	Anna Kajot-Kierownik (zastępstwo za: Jerzy Tester)
[dbo].[RejestrReklamacji]	[TekstReklamacji]	trzecia	trzecia i pół	29/01/2025 14:58:40	Anna Kajot-Kierownik (zastępstwo za: Jerzy Tester)
[dbo].[RejestrReklamacji]	[RowVersion]	4	5	29/01/2025 14:58:40	Anna Kajot-Kierownik (zastępstwo za: Jerzy Tester)

1 2 3 Pokaż: 10 21 - 27 z 27

Pokaż: 10 1 Idź do: 1 z 1 1 - 1 z 1

Audyt zmian dokumentu

Dodatkowo, zastępujący otrzymuje kopię powiadomień kierowanych do zastępowanego (z adnotacją, że powiadomienie dotyczy użytkownika zastępowanego), a przy próbie komunikacji przez czat z użytkownikiem, za którego wyznaczono zastępstwo, wyświetlana jest informacja o nieobecności i osobach zastępujących. Wiadomości wysyłane do nieobecnych użytkowników nie są przekierowywane do zastępujących.

Powiązane tematy:

- Uprawnienia do tworzenia zastępstw
- Dodaj zastępstwo
- Edytuj zastępstwo
- Usuń zastępstwo
- core.SubstitutionsPerformedByUser
- core.GetAllSubstitutions
- Ustawienia systemu, sekcja Zastępstwa

10.10. APLIKACJA NAXIOM NA URZĄDZENIACH MOBILNYCH

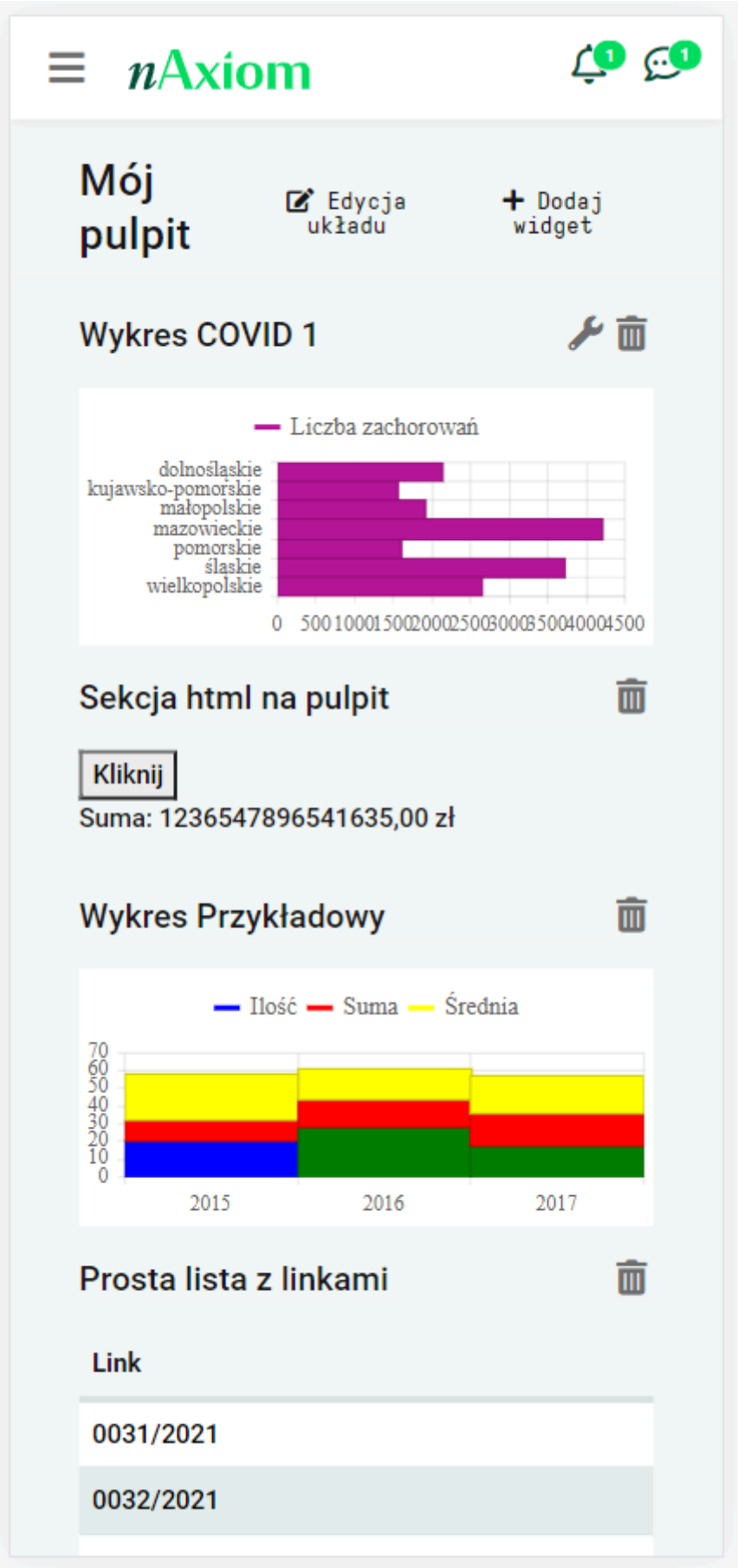
Podsumowanie:
Aplikacje nAxiom są aplikacjami *responsywnymi*, to jest dostosowanymi do wyświetlania na urządzeniach mobilnych. Dotyczy to pulpitu, widoków list i formularzy w aplikacji *FrontSPA*. Na urządzeniu mobilnym obsługa tych elementów interfejsu użytkownika wygląda nieco inaczej.

W TYM TEMACIE

- Mobilny pulpit
- Mobilny widok listy
- Mobilny formularz

MOBILNY PULPIT

W celu zalogowania do witryny nAxiom na urządzeniu mobilnym należy wpisać w przeglądarce adres aplikacji *FrontSPA* w witrynie. Po zalogowaniu jest wyświetlany pulpit, który wygląda nieco inaczej niż w przeglądarce na komputerze.



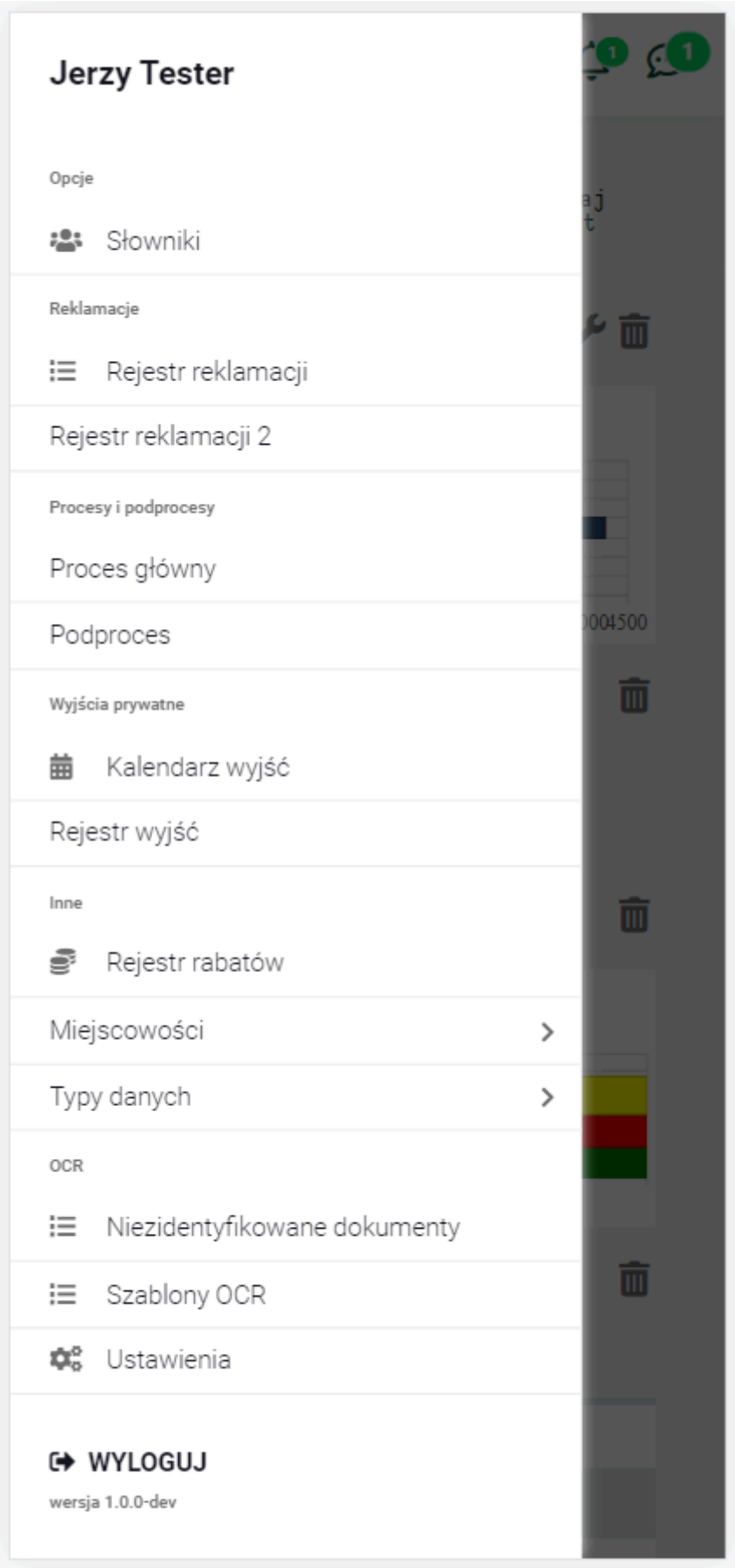
Pulpit nAxiom na urządzeniu mobilnym

Oprócz samego pulpitu z widżetami, który wypełnia prawie cały ekran, w górnej części znajduje się pasek narzędzi, który zawiera następujące elementy (od lewej):

- ikona menu ( ): rozwija menu witryny; aby je zwinąć, dotknij dowolny obszar ekranu poza menu
- ikona powiadomień ( ); pokazuje liczbę nieprzeczytanych powiadomień, po dotknięciu rozwija panel powiadomień
- ikona czatu ( ); pokazuje liczbę nieprzeczytanych konwersacji; rozwija panel czatu (komentarzy)

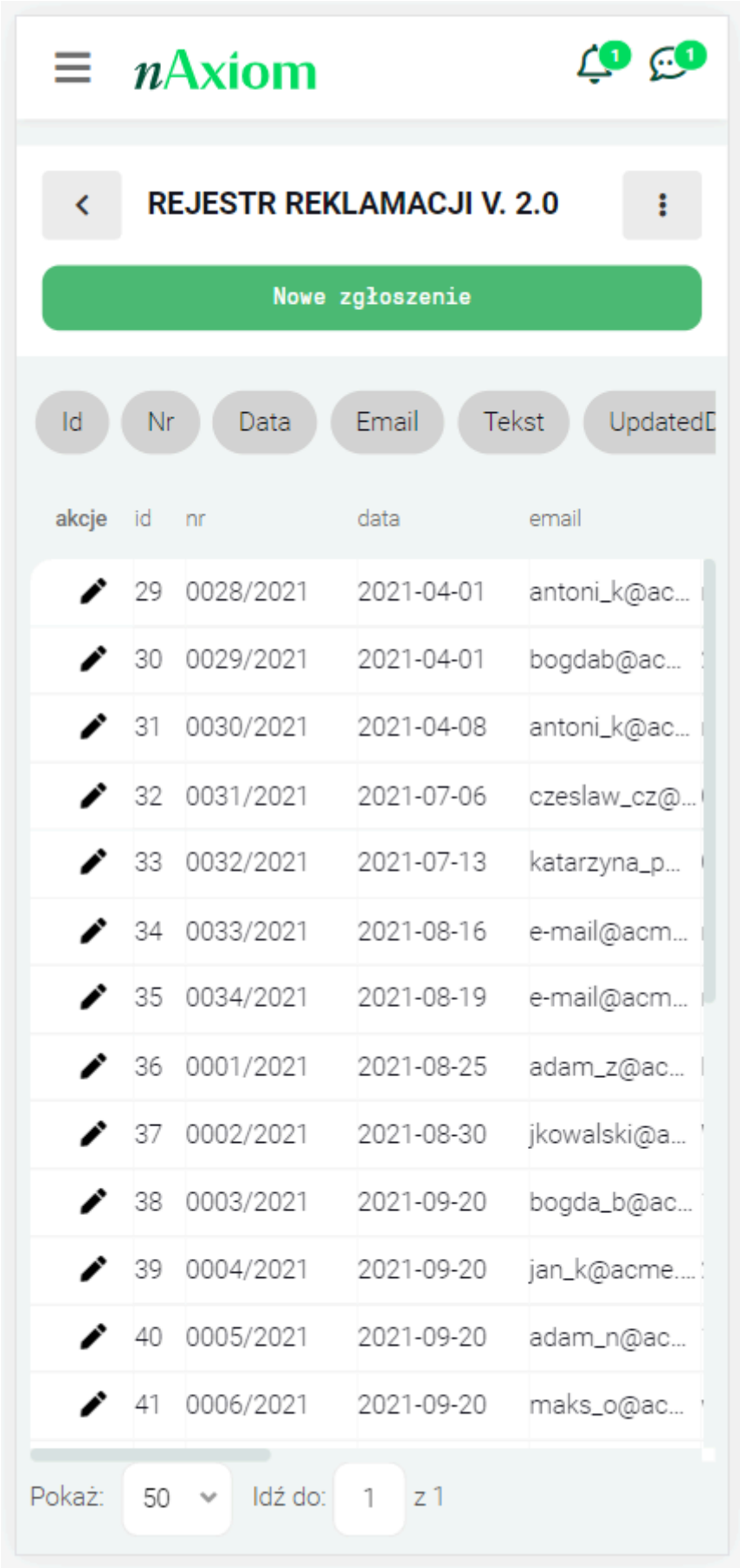
Rozwinięte menu, oprócz wszystkich sekcji i poleceń, do których użytkownik ma uprawnienia, zawiera u góry nazwę użytkownika, na końcu listy poleceń menu pozycję `Ustawienia`, która umożliwia edycję profilu użytkownika, a poniżej przycisk `Wyloguj` oraz numer bieżącej wersji nAxiom.

Pozycje menu, które mają pozycje podrzędne są po kliknięciu wyświetlane jako nowy panel, ze strzałką `Powrót` u góry.



Menu główne witryny nAxiom na urządzeniu mobilnym

MOBILNY WIDOK LISTY

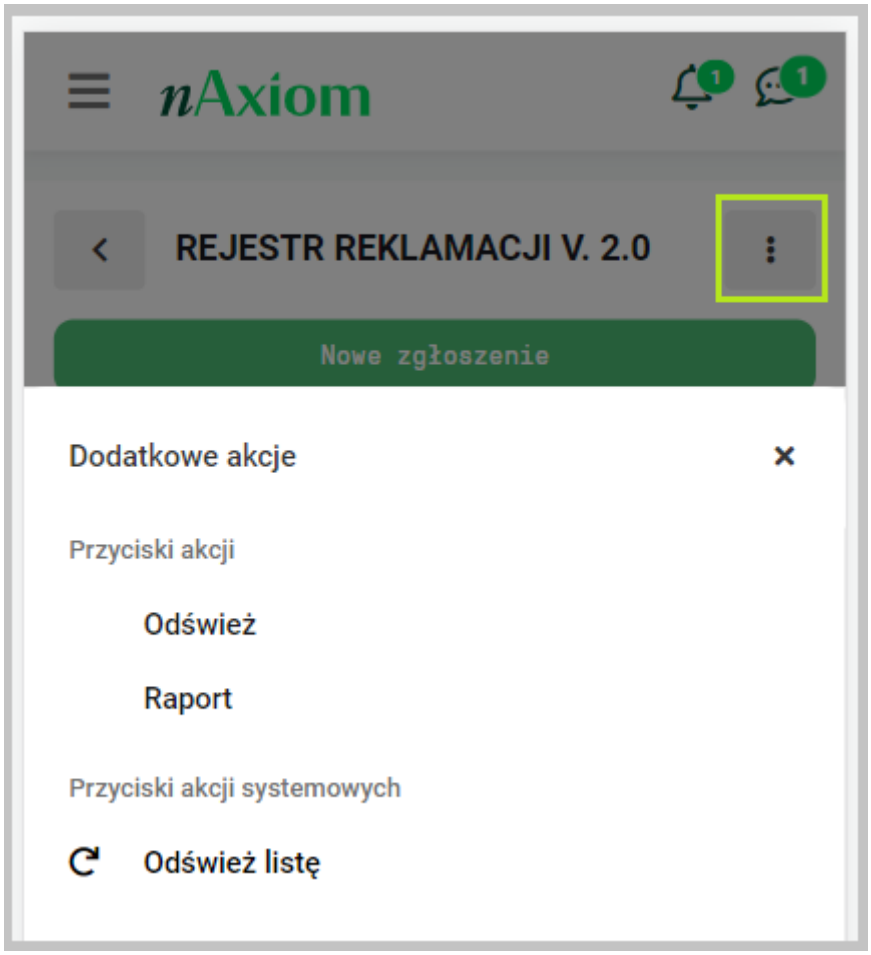


Widok listy na urządzeniu mobilnym

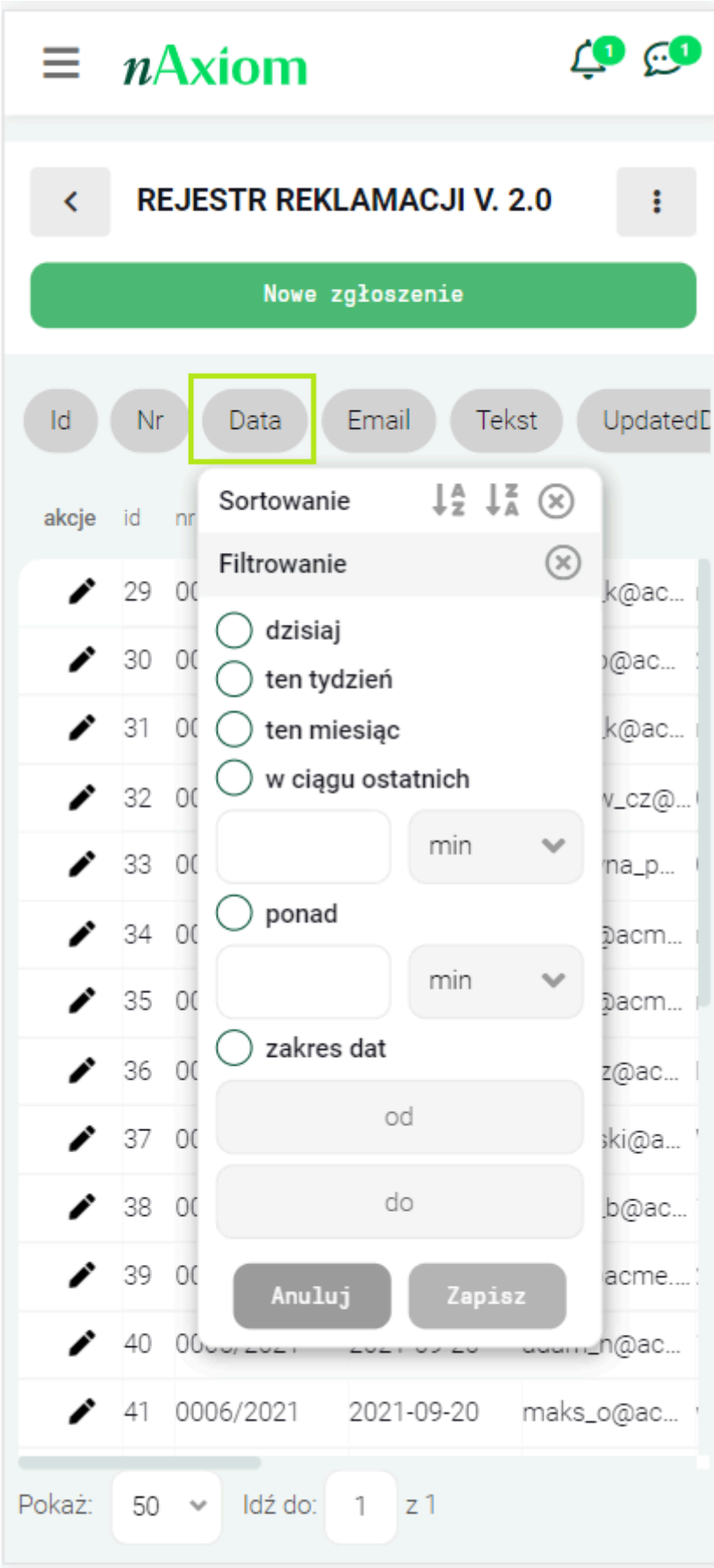
Widok listy na urządzeniu mobilnym również został dostosowany do małych rozmiarów ekranu. Na samej górze znajduje się pasek z nazwą listy. Z lewej strony tego paska znajduje się przycisk powrotu do poprzedniej strony. Z prawej strony przycisk z pionowym wielokropkiem wyświetla przycisk `Odśwież listę`.

Poniżej znajduje się, rozciągnięty na całą szerokość ekranu pierwszy przycisk zdefiniowany w sekcji `Górny panel` w widoku listy, zwykle jest to przycisk tworzenia nowego dokumentu. Pod nim znajdują się przyciski odpowiadające nagłówkom kolumn, dla których włączono filtrowanie i sortowanie w kreatorze listy (także dla kolumn nieaktywnych). Kliknięcie takiego przycisku powoduje wyświetlenie panelu z ustawieniami sortowania i filtrowania. Grupowanie danych można włączyć tylko w kreatorze listy, przeciągając odpowiednie kolumny do obszaru grupowania na podglądzie listy, albo po wyświetleniu listy na urządzeniu z większym ekranem.

Nagłówki kolumn listy zostały zmniejszone. Widoki list można swobodnie przewijać w pionie i poziomie, przeciągając zawartość palcem.

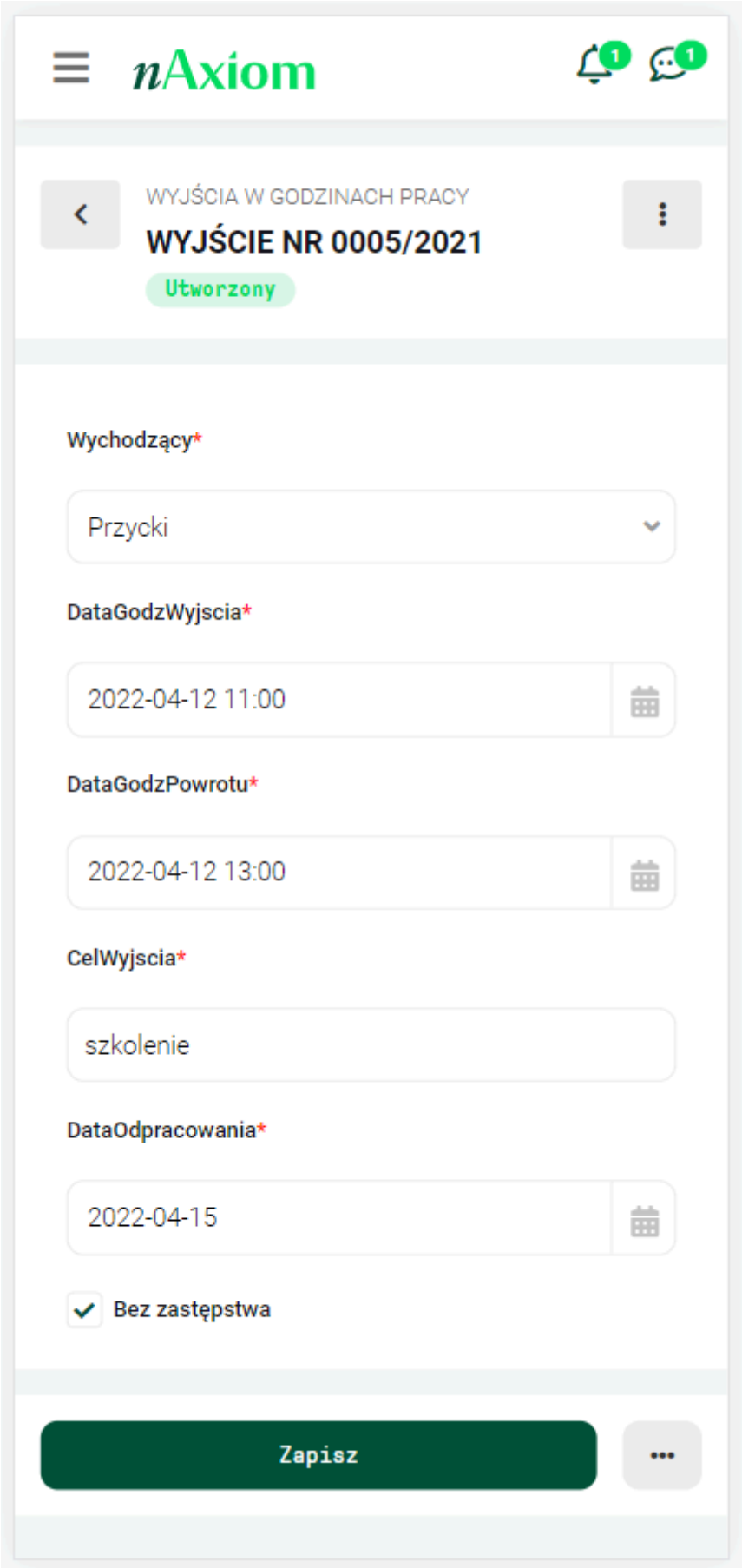


Panel przycisków



Panel sortowania/filtrowania

MOBILNY FORMULARZ

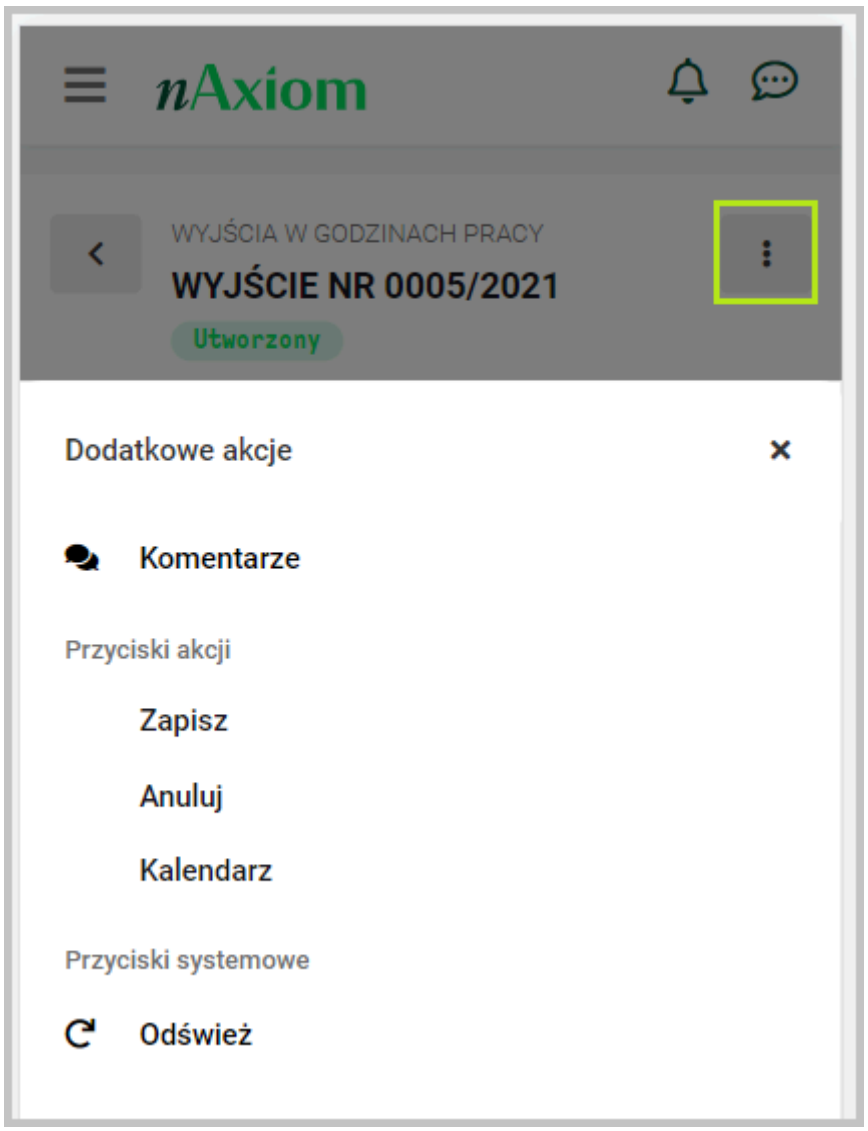


Formularz na urządzeniu mobilnym

Formularz w widoku mobilnym ma podobny pasek z nazwą jak lista, oprócz nazwy jest na nim wyświetlany także numer bieżącego dokumentu i jego status. Z lewej strony znajduje się przycisk `<` (wstecz), a z prawej przycisk , który wyświetla panel z następującymi elementami:

- `Komentarze`: wyświetla panel komentarzy
- `Przyciski akcji`: sekcja przycisków zamieszczonych w obu kontenerach przycisków akcji na pasku narzędzi formularza
- `Przejdź do kroku procesu`: sekcja przycisków diagramu procesu
- `Przyciski systemowe` (np. `Odśwież`)

U dołu formularza znajduje się przycisk przejścia do następnego kroku procesu (zmiany statusu) lub w przypadku nowo tworzonego dokumentu pierwszy przycisk zdefiniowany w sekcji *Toolbar*. Po kliknięciu ikony z wielokropkiem wyświetlane są pozostałe przyciski z nagłówka formularza.



Lista przycisków formularza

11. Obsługa wielu tenantów w nAxiom

Podsumowanie:
W środowisku nAxiom można definiować wielu *tenantów* („dzierżawców"). Każdy tenant korzysta z aplikacji FrontSPA i AdminSPA z indywidualnymi adresami URL (podstawowy adres środowiska z własnym przedrostkiem tenanta). Ponadto każdy tenant używa własnej instancji bazy danych, co zapewnia separację i bezpieczeństwo.

Tenantami zarządza administrator tenantów w aplikacji *TenantAdminSPA* dostępnej pod adresem *https://<bazowy-adres-naxiom>/tenantsadmin*. Podstawowe dane tenantów są zapisywane w osobnej instancji bazy danych, do której ma dostęp tylko administrator tenantów.

Podczas instalacji nAxiom w systemach z serwerem IIS tworzony jest tzw. *pierwszy tenant*. Jeśli klient nie ma potrzeby korzystania z wielu tenantów, może pracować w trybie *single tenant* z jednym tenantem. Domyślne przedrostek pierwszego tenanta w adresie URL witryny nAxiom jest pusty.

Każdy tenant korzysta z nAxiom w oparciu o własną licencję. Licencję musi zaimportować konsultant po zalogowaniu się do aplikacji *AdminSPA* (pod indywidualnym adresem tenanta). Do weryfikacji licencji służy ciąg znaków podawanych jako wartość parametru `Nazwa klienta`. Dla pierwszego tenanta określa się go podczas instalacji, a dla kolejnych w aplikacji *TenantAdminSPA* (`Customer name`). W konfiguracji tenanta definiuje się obowiązkowo `URL prefix` to jest przedrostek, pod którym użytkownicy danego tenanta uzyskują dostęp do nAxiom: *https://<url-prefix>/<base-naxiom-url>*.

Obsługa tenantów w architekturze platformy nAxiom jest realizowana przez następujące komponenty:

- *tenantsadmin*: serwis angularowy udostępniający GUI dla administratora tenantów
- *tenantsapi*: serwis dotnetowy obsługujący połączenie z bazą danych administratora tenantów.
- baza danych administratora tenantów: zawiera podstawowe dane poszczególnych tenantów, takie jak kod, nazwa, nazwa klienta (klucz licencji), przedrostek adresu URL do witryny nAxiom, parametry połączenia z bazą danych i inne; (domyślny format nazwy proponowany podczas instalacji to *<nazwa_witryny>_tenantsAdmin*)

BAZA DANYCH ADMINISTRATORA TENANTÓW

Baza danych administratora tenantów zawiera kilka tabel z ustawieniami tenantów oraz administratorów tenantów. Dane tenantów są zapisywane w tabeli *ta.Tenants*, która ma następujące kolumny:

- *Id* (*uniqueidentifier*): generowany automatycznie GUID tenanta.
- *Name* (*nvarchar(500)*): nazwa tenanta.
- *Code* (*nvarchar(50)*): kod tenanta.
- *UrlPrefix* (*nvarchar(50)*): przedrostek adresu URL do witryny nAxiom; domyślnie wartość taka jak *Code*.
- *IsActive* (*bit*): flaga aktywności tenanta.
- *ConnectionString* (*nvarchar(2000)*): parametry połączenia z bazą danych tenanta.
- *Settings* (*nvarchar(MAX)*): konfiguracja tenanta w formacie JSON.
- *Description* (*nvarchar(500)*): opis tenanta.
- *AdminEmail* (*nvarchar(254)*): adres e-mail administratora tenanta.

Ponadto baza danych zawiera tabelę z ustawieniami globalnymi oraz zestaw tabel dotyczących administratorów tenantów.

Z wyjątkiem zmiany ścieżek do repozytoriów , administrator tenantów nie ma potrzeby bezpośredniej ingerencji w dane zapisane w tabelach.

11.1. WDROŻENIE OBSŁUGI WIELU TENANTÓW

> **Podsumowanie:**
> Podczas instalacji nowej witryny nAxiom na serwerze IIS (Windows) tworzony jest login i baza danych administratora tenantów oraz login i baza danych pierwszego tenanta. W przypadku wdrażania platformy nAxiom jako zestawu kontenerów, konieczne jest powołanie pierwszego tenanta, co szczegółowo opisano w osobnym artykule.

Podczas instalacji nowej witryny nAxiom na serwerze IIS (Windows) tworzony jest login i baza danych administratora tenantów oraz login i baza danych pierwszego tenanta. W przypadku aktualizacji witryny z wersji nAxiom wcześniejszej niż 1.11, pierwszy tenant jest tworzony na podstawie dotychczasowej konfiguracji (pliki *appsettings.json* poszczególnych serwisów). Szczegóły instalacji opisano w Przewodniku instalacji .

Podczas aktualizacji parametry połączenia z bazą danych dla pierwszego tenanta są pobierane z pliku *appsettings.json* serwisu *api*.

Najważniejsze kwestie, o których należy pamiętać, to:

- Aby umożliwić połączenia *https* dla wielu tenantów wymagany jest certyfikat SSL typu *wildcard* **na poziomie bazowego adresu URL witryny nAxiom**. Ten adres jest określany w polu `Host` w kroku konfiguracji serwera IIS podczas instalacji i zapisywany w pliku *appsettings.json* serwisu *api*. Przykładowo:

Adres bazowy	Certyfikat dla
domena.com	*domena.com, *.domena.com*
sub.domena.com	*sub.domena.com, *.sub.domena.com*
test.sub.domena.com	*test.sub.domena.com, *.test.sub.domena.com*

- Instalacja serwisów *tenantsapi* i *tenantsadmin* musi być włączona (ustawienie domyślne).
- Jeśli baza danych administratora tenantów nie istnieje, wymagane jest podanie poświadczeń administratora bazy danych (*sysadmin* albo co najmniej *dbcreator* i *securityadmin*).
- Minimalny wymagany zestaw ról dla loginu do bazy administratora tenantów to

dbcreator, *securityadmin* i *public*.

W przypadku wdrażania platformy nAxiom jako zestawu kontenerów, konieczne jest powołanie pierwszego tenanta. Można to zrobić na dwa sposoby, które szczegółowo opisano w artykule Wdrożenie nAxiom z obrazów Docker . Poniżej wypunktowano najważniejsze informacje.

- Parametry połączenia z serwerem bazy danych dla bazy administratora tenantów podaje się w pliku *appsettings-custom.json* dla serwisu *tenant-api*.
- W celu utworzenia pierwszego tenanta w aplikacji *TenantAdminSPA* należy uruchomić tylko kontenery serwisów *tenant-admin* i *tenant-api*, następnie należy zalogować się jako administrator tenantów (login: *tenantsadmin*, hasło jest dostarczane z pakietem instalacyjnym) do aplikacji *TenantAdminSPA* pod adresem *https://<bazowy_URL_nAxiom>/tenantsadmin* i tam zdefiniować pierwszego tenanta zgodnie z opisem w dalszej części tego artykułu.
- Alternatywnie, można przygotować pliki konfiguracyjne *appsettings.json* serwisów dotnetowych, dodać do ich nazw nazwy serwisów jako przedrostek (np. *api_appsettings.json*) i skopiować do folderu o nazwie *toMigrate*. Jeśli podczas uruchamiania pełnego zestawu kontenerów nAxiom taki folder zostanie wykryty w katalogu głównym wdrożenia, na podstawie plików konfiguracyjnych zostanie utworzony pierwszy tenant.

11.2. ZARZĄDZANIE TENANTAMI

Podsumowanie:
Tenantami zarządza administrator tenantów w aplikacji *TenantAdminSPA*. Podstawowe dane tenantów są zapisywane w osobnej instancji bazy danych, do której ma dostęp tylko administrator tenantów.

W TYM TEMACIE

- Tenants list
- Global settings
- Library packages
- User list
- Environment secrets
- Tenant secrets
- Tenant variables
- Font families
- Obsługiwane formaty plików załączników

Do zarządzania tenantami służy odrębna aplikacja *TenantAdminSPA* dostępna pod adresem *https://<bazowy_URL_nAxiom>/tenantsadmin*. Łączy się ona z dedykowaną bazą danych, w której są przechowywane parametry konfiguracyjne poszczególnych tenantów. Po wpisaniu adresu w przeglądarce zostanie wyświetlone okno logowania, w którym należy podać standardowe dane logowania:

- login: *tenantsadmin*
- hasło jest dostarczane z pakietem instalacyjnym

Po zalogowaniu zostanie wyświetlona strona aplikacji z menu po lewej stronie i listą tenantów w części głównej. Poszczególne pozycje menu opisano w kolejnych sekcjach. Aplikacja jest dostępna w wersji angielskojęzycznej.

TENANTS LIST

Pozycja `Tenants list` wyświetla listę tenantów. Ustawienia każdego tenanta można edytować, klikając ikonę edycji w wierszu tenanta. Tenantów nie można usuwać, nieużywane definicje tenantów można dezaktywować. Kliknij przycisk `Add new tenant` i postępuj zgodnie z poleceniami w kreatorze.

- `Code`: kod tenanta; tylko znaki alfanumeryczne (*a-z*, *A-Z*, *0-9*) i łączniki (-); kodu tenanta określonego w tym kroku nie można zmienić.
- `Database`:
  - `New database`: dla tworzonego tenanta zostanie utworzona nowa baza danych o podanych parametrach.
  - `Existing database`: tenant zostanie powiązany z istniejącą bazą danych o podanych parametrach.
- `Database connection definition`:
  - `Parameters`: (domyślnie) należy podać parametry połączenia z bazą danych w polach:
    - `Server name`: adres serwera SQL Server.
    - `Login`: login do bazy danych; jeśli nie istnieje zostanie utworzony, zaleca się podanie loginu utworzonego z mocnym hasłem specjalnie dla administratora tenantów.
    - `Password`: Hasło do bazy danych (dostępny jest przycisk generowania mocnego hasła).
    - `Database`: Nazwa bazy danych pierwszego tenanta
  - `Connection string`: należy podać ciąg połączenia w polu tekstowym `Database connection string`.

Kliknij przycisk `Create`, aby utworzyć pierwszego tenanta w środowisku nAxiom. Wymagane jest podwójne potwierdzenie. Po utworzeniu wyświetlana jest strona edycji ustawień tenanta.

> **UWAGA**
> Po utworzeniu tenanta wymagany jest restart witryny nAxiom na serwerze IIS.

Strona `Edit tenant` ma trzy karty z polami ustawień tenanta. Poniżej opisano ustawienia na poszczególnych kartach.

Karta `Main`:

- `Identifier`: identyfikator GUID tenanta; tylko do odczytu.
- `Code`: kod definiowany podczas tworzenia tenanta; domyślnie używany jako przedrostek adresu URL dla tego tenanta; tylko do odczytu.
- `Name`: nazwa tenanta.
- `Customer Name`: ciąg znaków, na podstawie którego jest identyfikowana licencja danego tenanta (podczas pierwszego logowania konsultant tego tenanta musi zaimportować plik licencji zgodny z wartością w tym polu).
- `Description`: opis tenanta.
- `Is active`: przełącznik do aktywacji/dezaktywacji tenanta. Wyłączenie przełącznika powoduje, że wszyscy zalogowani użytkownicy tego tenanta zostaną wylogowani. Wylogowanie nastąpi w momencie wysyłania dowolnego żądania do serwisu *internal api*. Zmiana tego ustawienia zaczyna obowiązywać natychmiast, nie jest wymagany restart witryny.

> **WAŻNE**
> - Dla nowego tenanta przełącznik `Is active` jest zawsze wyłączony.
> - W przypadku dezaktywacji jedynego aktywnego tenanta (niezalecane) system zwróci błąd 503 z komunikatem „*No active tenants available*”.

Karta `Configuration`:

- `URL prefix`: przedrostek w adresie URL witryny nAxiom; tylko znaki alfanumeryczne (*a-z*, *A-Z*, *0-9*) i łączniki (-); użytkownicy końcowi z organizacji każdego tenanta logują się do witryny nAxiom, używając adresów różniących się przedrostkiem; przedrostek jest domyślnie taki sam, jak kod tenanta, ale można to zmienić; dla pierwszego tenanta powoływanego przez instalator przedrostek jest domyślnie pusty.

- `Tenant administrator e-mail`: adres e-mail do użytkownika z rolą administratora u danego tenanta.
- `Settings`: tekst w formacie JSON z ustawieniami konfiguracyjnymi, które można teraz ustawiać indywidualnie dla każdego tenanta, na przykład rozszerzenia nazw plików indeksowanych do wyszukiwania pełnotekstowego (o ile zostało skonfigurowane), ustawienia synchronizacji użytkowników i jednostek organizacyjnych z serwerem LDAP itp. Te ustawienia opisano w dodatku na końcu tego artykułu.
- `Logo`: umożliwia ustawienie niestandardowego logo używanego w oknach logowania i wylogowania dla bieżącego tenanta; w celu ustawienia logo należy wybrać obraz, odpowiednio powiększyć go lub pomniejszyć i ustawić prostokątną ramkę na wybranym fragmencie obrazu; dostępne kontrolki:
    - `Select image`: otwiera okno wyboru pliku;
    - `Clear logo`: przywraca standardowe logo (ustawione w sekcji `Global settings`);
    - `Zoom-in/out`: pomniejsza/powiększa obraz;
    - `Reset image`: przywraca oryginalny rozmiar obrazu.

Karta `Database`

- `Database connection definition`:
    - `Parameters`: (domyślnie) należy podać parametry połączenia z bazą danych w polach:
        - `Server name`: adres serwera SQL Server.
        - `Login`: Login do bazy danych.
        - `Password`: Hasło do bazy danych.
        - `Database`: Nazwa bazy danych pierwszego tenanta
    - `Connection string`: należy podać ciąg połączenia w polu tekstowym `Database connection string`.

Karta `Attachments`

- `Allowed attachment types`: lista dozwolonych formatów plików załączników (w formacie JSON). Aby dodać format do listy, należy podać jego rozszerzenie i typ MIME. Na końcu tego rozdziału znajduje się tabela formatów plików obsługiwanych przez nAxiom. Jeśli lista będzie pusta, typy plików załączników nie będą weryfikowane.

Karta `Guest login`

- `Guest login translations`: Tłumaczenia stringów wyświetlanych w oknie logowania. Należy podać następujące pozycje:
    - *LanguageCode*: kod języka, należy stosować kod dwuznakowy,
    - *LanguageName*: nazwa języka wyświetlana w polu wyboru języka,
    - *PageTitle*: tytuł witryny wyświetlany na zakładce karty przeglądarki,
    - *WelcomeTitle*: nadrzędny tytuł witryny wyświetlany w oknie logowania,
    - *WelcomeSubtitle*: podrzędny tytuł witryny wyświetlany w oknie logowania,
    - *InputPlaceholder*: placeholder w polu na hasło użytkownika,
    - *InputRequiredMessage*: komunikat błędu w przypadku braku podania hasła,
    - *ButtonText*: tekst wyświetlany w przycisku logowania,
    - *LoginErrorMessage*: ogólny komunikat błędu w przypadku logowania
- `Guest login background image`: Obraz wyświetlany w tle okna logowania w tym trybie

GLOBAL SETTINGS

W sekcji `Global settings` określa się globalne ustawienia środowiska nAxiom obowiązujące dla wszystkich tenantów.

- Sekcja `General`
    - `Directory location...`: katalog główny repozytoriów plikowych wszystkich tenantów; domyślnie *<katalog_instalacyjny_nAxiom>\SystemRepositories*; w tym katalogu każdy tenant ma własny podkatalog o nazwie takiej jak GUID tenanta, a w nim podkatalogi repozytoriów różnych zasobów (wiadomości e-mail, załączniki itp.).
    - `Error logging level`: poziom logowania komunikatów dla serwisów *api* i *publicapi*; taki sam dla wszystkich tenantów; dostępne wartości (poziomy są wymienione w kolejności od najniższej istotności, każdy poziom obejmuje także informacje z poziomów o wyższej istotności):
        - `Trace`: szczegółowe informacje przydatne przy śledzeniu miejsca wystąpienia błędu na stronie;
        - `Debug`: szczegóły na temat akcji podjętych przez system, odzwierciedlające jego aktualny stan;
        - `Info`: wiadomości informacyjne odnoszące się do sposobu działania systemu;
        - `Warn`: ostrzeżenia przed potencjalnym zagrożeniem w działaniu systemu przy wykonywaniu dalszej części kodu;

        - `Error`: (wartość domyślna) błędy występujące podczas żądania wysyłanego przez użytkownika, umożliwiające dalszą pracę systemu;
        - `Fatal`: błędy, które uniemożliwiają dalsze działanie systemu i powodują jego zawieszenie.
    - `Logo`: patrz opis powyżej .
- Sekcja `API anonymous endpoints rate limiting`:
    - `Maximum number of requests from a single IP address per 1 minute`: wartość domyślna 100; wymaga restartowania serwisów API i PublicAPI; przekroczenie limitu powoduje wyświetlenie komunikatu o błędzie i pustej strony.
- Sekcja `Google reCAPTCHA v3`:
    - `Secure public forms`: włącza lub wyłacza mechanizm weryfikacji reCAPTCHA dla formularzy publicznych ; po włączeniu wymaga podania kolejnych parametrów.
    - `SiteKey`: klucz publiczny witryny uzyskiwany w procesie rejestracji w Google Cloud; służy do uzyskania tokena używanego w żądaniu weryfikacji.
    - `SecretKey`: klucz tajny witryny uzyskiwany w procesie rejestracji w Google Cloud; wraz z tokenem jest wysyłany w żądaniu weryfikacji.
    - `Acceptable scoring`: minimalna wartość oceny zwracanej przez reCAPTCHA, która zostanie uznana za potwierdzenie interakcji z człowiekiem; domyślnie 0,5.
- Sekcja `Secrets Management System`: pozwala określić parametry do zewnętrznego oprogramowania pełniącego rolę *sejfu* służącego do przechowywania haseł i certyfikatów używanych w aplikacjach biznesowych; patrz kolejne sekcje w tym temacie oraz artykuł Integracja nAxiom z OpenBao :
    - `Enable Secrets`: włącza korzystanie z zewnętrznego sejfu,
    - `URL`: adres URL sejfu,
    - `Username`: nazwa użytkownika sejfu,
    - `Password`: hasło użytkownika sejfu,
    - `Namespace`: przestrzeń nazw, w której będą przechowywane hasła i certyfikaty.

LIBRARY PACKAGES

Wyświetla listę *paczek bibliotek*, to jest pakietów zawierających biblioteki DLL, które mogą być używane w akcjach C#. Aby zdefiniować paczkę bibliotek należy podać jej nazwę i załadować co najmniej jeden plik biblioteki DLL. Zdefiniowane paczki bibliotek można eksportować do pliku *naxlib*. Taki plik można następnie zaimportować do innego

środowiska nAxiom. Paczki są przechowywane w bazie tenantów w tabelach *ta.LibraryPackages* (definicje) i *ta.LibraryPackageFiles* (pliki bibliotek).

USER LIST

Wyświetla listę użytkowników z uprawnieniami do administrowania tenantami. Aby dodać użytkownika, kliknij przycisk `Add new user` i na wyświetlonej stronie podaj nazwę użytkownika, adres e-mail i hasło.

Każdy administrator tenantów może dodawać nowych administratorów oraz edytować istniejących. W celu zmiany hasła kliknij ikonę edycji dla danego administratora i podaj nowe hasło.

ENVIRONMENT SECRETS

Na tej stronie można definiować *sekrety* na poziomie całego środowiska nAxiom, to jest dostępne dla wszystkich tenantów. W celu zdefiniowania sekretu należy określić:

- `Code`: maksymalnie 100 znaków, małe i wielkie litery alfabetu łacińskiego, cyfry oraz łącznik (-); kod sekretu musi być unikalny.
- `Name`: maksymalnie 255 znaków.
- `Type`: dostępne wartości:
  - `Secret`: sekret typu tekstowego, np. hasło, klucz API, token dostępu; do sekretów tego typu można się odwoływać przy użyciu zmiennych aplikacji .
  - `File`: sekret typu plikowego, np. certyfikat.
- `Value`: pole tekstowe do wprowadzenia sekretu tekstowego lub przycisk `Upload file` do wczytania sekretu plikowego.

TENANT SECRETS

Zakłada się, że w aplikacjach używanych przez różnych tenantów używane będą takie same sekrety, ale o różnych wartościach. Dlatego osobno definiuje się sekrety, a osobno wartości tych sekretów dla poszczególnych tenantów. W związku z tym pozycja menu `Tenant secrets` zawiera dwie pozycje podrzędne:

- `Definitions`: na tej stronie definiuje się sekrety, określając podobnie jak w przypadku sekretów środowiska atrybuty `Code`, `Name` i `Type`.
- `Values`: na tej stronie określa się wartości sekretów zdefiniowanych na stronie `Definitions` indywidualnie dla każdego tenanta.

TENANT VARIABLES

Służy do definiowania zmiennych konfiguracyjnych tenanta, które będą obsługiwane przez program OpenBao. Zawiera dwie pozycje podrzędne.

`Definitions`: służy do definiowania zmiennych; wymagane jest podanie kodu zmiennej zgodnie z przyjętą konwencją, to jest jako ścieżki JSON z pliku appsettings.json, w której kolejne segmenty są rozdzielone dwoma znakami podkreślenia. Jeśli dowolny segment ścieżki waskazuje na element tablisy, należy podać jego indeks (pierwszy element ma indeks 0). Na przykład dla klucza *password* z poniższego pliku konfiguracyjnego

```
"OpenLDAPConf": {
  "UserId": "3F53097E-BFE6-4690-E075-08D6CEBA7D84",
  "RepeatInterval": 240,
  "UserSettings": [
    {
      "SynchronizationType": "Full",
      "DefaultEmailAddress": "brakmaila@domain.com",
      "DefaultLanguageCode": "pl",
      "SkipUsersWithoutEmail": false,
      "DefaultOrganizationSymbols": [ "System" ],
      "SystemRoleCode": "CLIENT",
      "UserMatchingType": "Login",
      "DefaultBusinessRoleCodes": [ "LOKALNA" ],
      "LDAPConfiguration": {
        "Path": "192.168.40.51",
        "User": "cn=Manager,dc=maxcrc,dc=com",
        "Password": "veEzPBSsXgrmcRhztdbWUA=="
      }
    }
  ]
}
```

należy zdefiniować następujący kod zmiennej:

OpenLDAPConf__UserSettings__0__LDAPConfiguration__Password

`Values`: służy do przypisywania wartości zdefiniowanej zmiennej indywidualnie dla każdego tenanta.

> **WAŻNE**
> Hasła i tym podobne dane wrażliwe należy podać w postaci tekstowej, bez szyfrowania.

FONT FAMILIES

Służy do dodawania rodzin czcionek, z których można będzie korzystać w nAxiom po wybraniu odpowiednich ustawień w Kreatorze stylów systemu w Admin SPA. Obsługiwane są następujące formaty plików czcionek: TTF, OTF, WOFF i WOFF2.

Rozwijane listy wartości w kolumnach `Weight` i `Style` pozwalają określić, dla jakich ustawień *font-weight* i *font-style* w arkuszach CSS zostanie wybrana bieżąca czcionka z danej rodziny.

OBSŁUGIWANE FORMATY PLIKÓW ZAŁĄCZNIKÓW

Poniższa tabela zawiera listę formatów plików, które mogą być weryfikowane przez nAxiom po ich dodaniu do listy dozwolonych typów załączników (`Edit tenant` > `Attachments` > `Allowed attachment types`).

Lp.	Rozszerzenie	MIME Type	Źródło detekcji
Obrazy			
1	jpg	image/jpeg	Sygnatura
2	png	image/png	Sygnatura
3	gif	image/gif	Sygnatura
4	bmp	image/bmp	Sygnatura
5	tif	image/tiff	Sygnatura
6	ico	image/x-icon	Sygnatura
7	webp	image/webp	Sygnatura
8	psd	image/vnd.adobe.photoshop	Sygnatura
9	heic	image/heic	Sygnatura
10	heif	image/heic	Sygnatura
11	avif	image/avif	Sygnatura
12	jxl	image/jxl	Sygnatura
13	wmf	image/x-wmf	Sygnatura
14	svg	image/svg+xml	Analiza tekstu
Dokumenty			
15	pdf	application/pdf	Sygnatura
16	rtf	application/rtf	Sygnatura
17	ps	application/postscript	Sygnatura
18	eps	application/postscript	Sygnatura
19	docx	application/vnd.openxmlformats-officedocument.wordprocessingml.document	Analiza ZIP
20	xlsx	application/vnd.openxmlformats-officedocument.spreadsheetml.sheet	Analiza ZIP

Lp.	Rozszerzenie	MIME Type	Źródło detekcji
21	pptx	application/vnd.openxmlformats-officedocument.presentationml.presentation	Analiza ZIP
22	doc	application/msword	Analiza OLE
23	xls	application/vnd.ms-excel	Analiza OLE
24	ppt	application/vnd.ms-powerpoint	Analiza OLE
25	odt	application/vnd.oasis.opendocument.text	Analiza ZIP
26	ods	application/ vnd.oasis.opendocument.spreadsheet	Analiza ZIP
27	odp	application/ vnd.oasis.opendocument.presentation	Analiza ZIP
28	epub	application/epub+zip	Analiza ZIP
29	chm	application/vnd.ms-htmlhelp	Sygnatura
Web / Tekst / XML			
30	html/htm	text/html	Analiza tekstu
31	xhtml	application/xhtml+xml	Analiza tekstu
32	xml	application/xml	Sygnatura
33	drawio	application/vnd.jgraph.mxfile	Analiza tekstu
34	txt	text/plain	Sygnatura / heurystyka UTF-8
35	sh	application/x-sh	Sygnatura
E-mail			

Lp.	Rozszerzenie	MIME Type	Źródło detekcji
36	eml	message/rfc822	Sygnatura + analiza nagłówków
37	pst	application/vnd.ms-outlook-pst	Sygnatura
38	msg	application/vnd.ms-outlook	Analiza OLE
Archiwa			
39	zip	application/zip	Analiza ZIP
40	rar	application/vnd.rar	Sygnatura
41	7z	application/x-7z-compressed	Sygnatura
42	gz	application/gzip	Sygnatura
43	bz2	application/x-bzip2	Sygnatura
44	xz	application/x-xz	Sygnatura
45	tar	application/x-tar	Sygnatura
46	cab	application/vnd.ms-cab-compressed	Sygnatura
47	z	application/x-compress	Sygnatura
48	lz4	application/x-lz4	Sygnatura
49	zst	application/zstd	Sygnatura
Pliki wykonywalne / binarne			
50	exe	application/vnd.microsoft.portable-executable	Sygnatura
51	elf	application/x-elf	Sygnatura
52	class	application/java-vm	Sygnatura
53	(brak)	application/octet-stream	Sygnatura (Mach-O

Lp.	Rozszerzenie	MIME Type	Źródło detekcji
			32-bit)
54	(brak)	application/octet-stream	Sygnatura (Mach-O 64-bit)
55	(brak)	application/octet-stream	Sygnatura (Mach-O 32-bit reverse)
56	(brak)	application/octet-stream	Sygnatura (Mach-O 64-bit reverse)
57	(brak)	application/octet-stream	Sygnatura (Java Serialized Data)
58	jks	application/x-java-keystore	Sygnatura
59	dex	application/x-dex	Sygnatura
60	wasm	application/wasm	Sygnatura
61	jar	application/java-archive	Analiza ZIP
62	apk	application/vnd.android.package-archive	Analiza ZIP
Czcionki			
63	ttf	font/ttf	Sygnatura
64	otf	font/otf	Sygnatura
65	woff	font/woff	Sygnatura
66	woff2	font/woff2	Sygnatura
Bazy danych			
67	sqlite	application/vnd.sqlite3	Sygnatura
68	accdb	application/x-msaccess	Sygnatura

Lp.	Rozszerzenie	MIME Type	Źródło detekcji
69	mdb	application/x-msaccess	Sygnatura
Bezpieczeństwo / System			
70	lnk	application/x-ms-shortcut	Sygnatura
71	iso	application/x-iso9660-image	Sygnatura
72	crx	application/x-chrome-extension	Sygnatura
73	bak	application/x-mssql-backup	Sygnatura
OLE (fallback)			
74	vsd	application/vnd.visio	Analiza OLE
75	ole	application/x-ole-storage	Analiza OLE (fallback)
Wszystkie inne pliki			
		application/octet-stream	Domyślny (default)

11.3. REPOZYTORIA TENANTÓW

Podsumowanie:
W trybie obsługi wielu tenantów obowiązują specjalne zasady dotyczące repozytoriów plikowych.

W nowych środowiskach repozytoria dyskowe wszystkich tenantów znajdują się w ścieżce określonej przez wartość parametru `Directory location for relative paths of file repositories` (*TenantAdminSPA* > `Global settings`).

Podkatalog dla każdego tenanta jest określony przez wartość klucza *SubDirName* w sekcji *FileStorageConfiguration* w ustawieniach konfiguracyjnych tenanta (*TenantAdminSPA* > `Edit tenant` > `Configuration` > `Settings`). Domyślnie ta wartość jest równa identyfikatorowi GUID tenanta.

Dodatkowo, drugi parametr w sekcji *FileStorageConfiguration*, *AllowUseRepoFullDir*, określa, czy konsultanci danego tenanta mogą definiować repozytoria, używając ścieżek względnych, czy bezwzględnych.

Dla nowych tenantów wartość domyślna to *false* — konsultanci mogą definiować repozytoria używając wyłącznie ścieżek względnych odnoszących się do podkatalogu tenanta w katalogu głównym repozytoriów.

W przypadku aktualizacji nAxiom z wersji sprzed obsługi wielu tenantów parametr ten ma wartość *true*. W takim przypadku konsultanci tego tenanta będą mogli określać repozytoria dyskowe, używając tylko ścieżek bezwzględnych.

Tabela *core.AttachmentRepositories* zawiera trzy kolumny do obsługi repozytoriów tenantów:

- *TableName*: nazwa tabeli w przypadku repozytorium bazodanowego.
- *AbsoluteDirectory*: ścieżka bezwzględna do repozytorium dyskowego.
- *RelativeDirectory*: ścieżka względna do repozytorium dyskowego.

Instalator ustawia wartość parametru *AllowUseRepoFullDir* na podstawie danych z kolumny *AbsoluteDirectory* w bazie danego tenanta. Jeśli ta kolumna jest pusta, parametr ma wartość *false*.

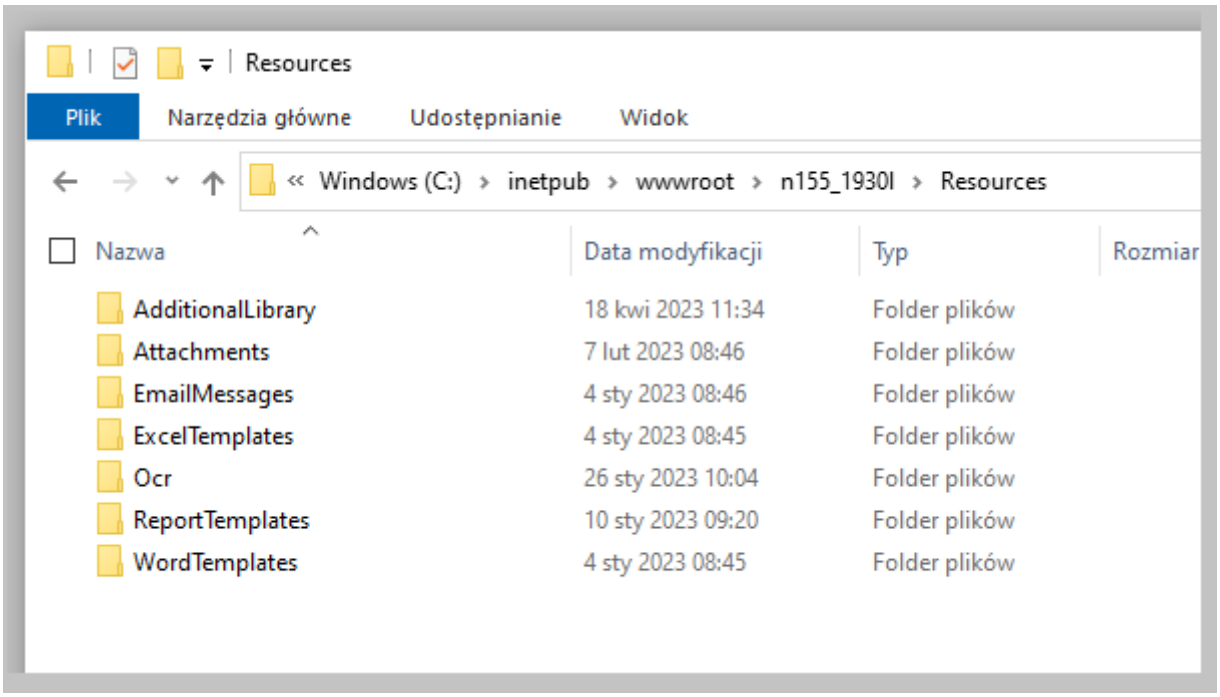
Aby dla konkretnego tenanta przejść ze ścieżek bezwzględnych na ścieżki względne, należy wykonać następujące kroki:

1. Utwórz katalog główny repozytoriów dyskowych (jeśli nie istnieje) zgodnie z wartością parametru `Directory location for relative paths...` na stronie `Global settings` w aplikacji *TenantAdminSPA*.
2. W katalogu głównym utwórz podkatalog o nazwie określonej przez klucz *SubDirName* w sekcji *FileStorageConfiguration* w ustawieniach konfiguracyjnych tenanta.
3. Skopiuj do tego podkatalogu katalogi repozytoriów wraz z zawartością.
4. Usuń dane z kolumny *AbsoluteDirectory* w tabeli *core.AttachmentRepositories* w bazie danych tenanta.
5. Zmień wartość klucza *AllowUseRepoFullDir* z *true* na *false*.
6. Uruchom ponownie witrynę.
7. Poleć konsultantom tenanta utworzenie repozytoriów o nazwach odpowiadających przenoszonym repozytoriom.

Przykład:

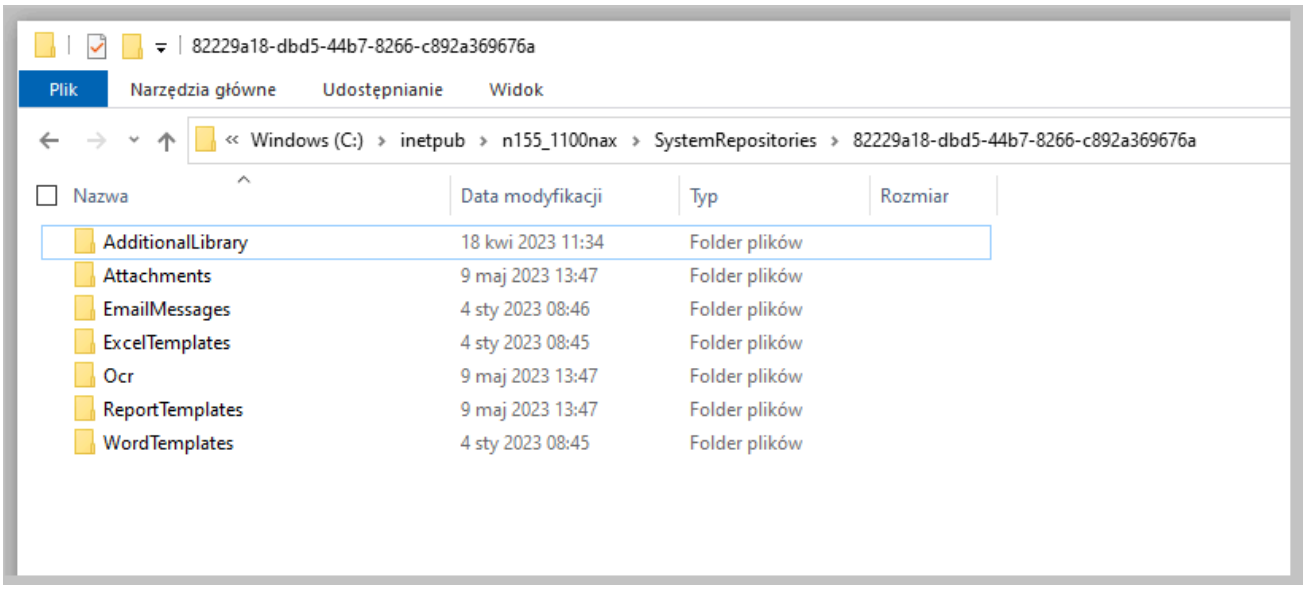
W wyniku aktualizacji witryny nAxiom z repozytoriami dyskowymi tworzony jest pierwszy tenant o identyfikatorze GUID *82229a18-dbd5-44b7-8266-c892a369676a*, dla którego klucz

AllowUseRepoFullDir ma wartość *true*. Strukturę katalogów repozytoriów przedstawiono poniżej.



Struktura repozytoriów, ścieżki bezwzględne

Aby przełączyć tego tenanta na korzystanie ze ścieżek względnych, należy skopiować katalogi repozytoriów z zawartością do katalogu jak na ilustracji poniżej oraz wykonać pozostałe kroki powyższej procedury.



Struktura repozytoriów, ścieżki względne

Ścieżka *C:\inetpub\n155_1100nax\SystemRepositories* to wartość parametru `Directory location for relative paths...`.

11.4. USTAWIENIA KONFIGURACYJNE W TENANTSADMINSPA

Podsumowanie:
W aplikacji *TenantsAdminSPA*, w ustawieniach każdego tenanta na karcie `Configuration` są dostępne ustawienia konfiguracyjne z pliku *appsettings.json* serwisu *api*. Ustawienia te umożliwiają zaawansowaną konfigurację tenantów.

- ***IndexableFileExtensions***: określa rozszerzenia nazw plików indeksowanych przez usługę wyszukiwania pełnotekstowego.
- ***CoreTablesModificationEnabled***: pozwala włączyć dostęp do tabel systemowych *core.SyncOUInstances* i *core.SyncUserProfiles* w kreatorze modeli danych, których edycja jest wymagana w przypadku synchronizacji struktury organizacyjnej i listy użytkowników z systemem zewnętrznym. Ponadto w sekcji *SchemaSettings* można określić nazwy schematów tabel systemowych (domyślnie *core*) i tabel serwisu *auth* (domyślnie *auth*) oraz schematów tabel użytkownika (ustawienie domyślne * odpowiada dowolnym schematom).
- ***OpenLDAPConf*** i ***StructureSynchronizationConfiguration***: sekcje odpowiadające konfiguracji synchronizacji profili użytkowników i struktury organizacyjnej z serwerem usług katalogowych LDAP. Szczegółowy opis zawiera osobny artykuł Synchronizacja z usługami katalogowymi
- ***PageTitles: LoginPage***: pozwala określić niestandardowy tytuł strony logowania do nAxiom (wartość domyślna *Logowanie*).
- ***EmailSenderConf: AddressFrom***: adres e-mail administratora danego tenanta, obecnie nie używany.
- ***Security: HideResetPasswordLogic***: decyduje, czy w oknie logowania ma być widoczny link do resetowania hasła.
- ***ExternalLogin***: ta sekcja zawiera ustawienia konfiguracyjne logowania z wykorzystaniem zewnętrznego dostawcy usług uwierzytelniania. Ustawienia te zostały opisane w artykule Konfiguracja logowania zewnętrznego do nAxiom.
- ***WindowsLogin***: w tej sekcji można określić konfigurację logowania z wykorzystaniem kontrolera domeny Windows; zawiera ona następujące parametry:
    - ***UseWindowsLogin***: włącza uwierzytelnianie Windows,
    - ***DisplayName***: tytuł okna logowania (domyślnie *Logowanie Windows*),
    - ***AutomaticAuthentication***: określa, czy logowanie ma się odbywać automatycznie; jeśli nie będzie wyświetlane okno logowania,
    - ***AutomaticRedirect***: określa, czy podczas logowania do nAxiom ma następować przekierowanie do kontrolera domeny Windows w celu zalogowania,
- ***UseADAuthentication***: pozwala włączyć uwierzytelnianie na serwerze usług LDAP,
- ***DefaultCulture***: określa domyślne ustawienia narodowe dla indywidualnych

tenantów (wartość domyślna *pl-PL*),

- ***GuestLogin***: włącza tryb logowania za pomocą hasła; patrz opis
- ***SimpleLogin***: pozwala włączyć wyświetlanie strony uproszczonego logowania, przez którą użytkownik loguje się do nAxiom używając kodu numerycznego ustawianego w profilu użytkownika,
- ***FileStorageConfiguration***: w tej sekcji określa się konfigurację przechowywania plików statycznych (załączniki, wiadomości e-mail, rozpoznane pliki OCR i inne) dla danego tenanta; zawiera ustawienia:
    - ***SubDirName***: nazwa folderu w systemie plików hosta aplikacji, w którym będą znajdować się repozytoria plikowe tenanta; domyślnie jest to identyfikator GUID tenanta. Dla wdrożeń w środowisku Windows jest to podfolder folderu *SystemRepositories* w katalogu instalacyjnym nAxiom,
    - ***AllowUseRepoFullDir***: określa, czy użytkownicy tego tenanta mogą określać lokalizację swoich repozytoriów plikowych za pomocą ścieżek bezwzględnych; domyślnie parametr jest włączony dla pierwszego tenanta w przypadku aktualizacji witryny nAxiom, a wyłączony w innych przypadkach. Jeśli parametr jest wyłączony, użytkownicy tenanta mogą tworzyć repozytoria plikowe tylko w folderze określonym parametrem *SubDirName*; aby zmienić ustawienie z *false* na *true*, należy wykonać procedurę opisaną w rozdziale Repozytoria plików .
- ***PasswordPolicy***: w tej sekcji określa się ustawienia polityki wymuszania okresowej zmiany haseł
    - ***EnforcePasswordExpiration***: *true/false*, włącza politykę wymuszania okresowej zmiany haseł; użytkownik z odpowiednimi uprawnieniami PBA może wyłączać indywidualne profile użytkowników z tej polityki.
    - ***PasswordExpirationRules***: sekcja ustawień konfiguracyjnych polityki
        - ***ForceExpirationAfterRestart***: *true/false*, zmiana wartości tego ustawienia z *false* na *true* spowoduje automatyczne zresetowanie okresu ważności haseł dla wszystkich użytkowników, co spowoduje, że podczas pierwszego logowania będą musieli zmienić hasło
        - ***ValidityPeriodDays***: *int*, okres ważności haseł w dniach (domyślnie 90), zmiana hasła zostanie wymuszona podczas pierwszego logowania w kolejnym dniu kalendarzowym po upływie tego okresu.
        - ***ReminderDaysBeforeExpiration***: *int*, okres w dniach przed upływem ważności hasła (domyślnie 5) w którym codziennie będzie wysyłane do użytkownika przypomnienie o zbliżającym się terminie zmiany hasła.