



Korzystanie z interfejsu nAxiom PublicAPI

Wersja nAxiom: 1.14.0.3

Spis treści

1. Wprowadzenie	2
2. Dokumentacja PublicAPI	2
3. Konfiguracja klienta <i>auth</i>	4
4. Konfiguracja profilu użytkownika	5
5. Uzyskiwanie tokenu	8
6. Przykłady żądań do PublicAPI	11
6.1. [GET]/PermissionDelegation/Get	11
6.2. [GET]/Document/Get	13
6.3. [POST]/Document/Save	14
6.4. [POST]/Attachment/UploadAttachment	17
7. Błędy braku uprawnień	19
8. Znane problemy	22

1. Wprowadzenie

Na potrzeby integracji z systemami zewnętrznymi platforma nAxiom udostępnia interfejs PublicAPI z kilkudziesięcioma endpointami obsługującymi szereg operacji w zakresie pobierania, dodawania i aktualizowania danych przetwarzanych w witrynie nAxiom. Opis tych endpointów zawiera dokumentacja generowana automatycznie przez program Swagger, dostępna pod adresem witryny naxiom w katalogu *publicapi* (*{adres_URL_witryny_naxiom}/publicapi/*).

W tym artykule podano podstawowe informacje dotyczące korzystania z interfejsu PublicAPI za pomocą programu *Postman* i polecenia *curl*.

2. Dokumentacja PublicAPI

Dokumentacja interfejsu PublicAPI witryny nAxiom jest dostępna pod adresem witryny w katalogu *publicapi*. Na przykład:

https://naxiom.mysite.com/publicapi/

UWAGA

W adresie należy wpisać końcowy ukośnik, w przeciwnym razie zostanie wyświetlona strona błędu 404.

Swagger UI

naxiom.local:3131/publicapi/index.html

Swagger
Supported by SMARTBEAR

Select a definition API V1

Public API ^{v1} OAS3

</publicapi/swagger/v1/swagger.json>

Public system services. Intended use: integration with external systems.

Servers

/publicapi

Authorize

AttachmentAnonymous

Methods for attachments and not requiring login. Used by the Word plug-in.

POST /AttachmentAnonymous/Update Updates a given attachment by overwriting. Method does not require authorization by auth module, it is used by the Word plug-in.

POST /AttachmentAnonymous/UpdateAsNewVersion Update a given attachment by adding a new version. Method does not require authorization by auth module, it is used by the Word plug-in.

Attachment

Controller contains methods for attachments

GET /Attachment/DownloadAttachment Download the attachment by id

GET /Attachment/GetAttachment Retrieves information about attachment by its id

Dokumentacja PublicApi wygenerowana przez narzędzie Swagger

W wersji nAxiom 1.14.1.0 zostanie opublikowana zmiana umożliwiająca autoryzację użytkownika z poziomu strony dokumentacji Swagger (przyciskiem `Authorize`) i testowanie żądań.

3. Konfiguracja klienta *auth*

WSKAZÓWKA

Do uzyskania tokenu z serwisu *auth* można także użyć predefiniowanego klienta *openjs*, jednak nie można wprowadzać żadnych zmian w jego konfiguracji, ponieważ jest on używany m.in. przez wtyczki nAxiom do programów Outlook i Word.

Aby użytkownik mógł uzyskać token potrzebny do uwierzytelniania żądań do PublicAPI, wymagane jest skonfigurowanie klienta w pliku *appsettings.json* serwisu *auth*. W tym celu, w sekcji *Clients* tego pliku należy dodać wpis:

```
{
  "ClientId": "publicApiClient",
  "ClientName": "PublicAPI Client",
  "AllowedGrantTypes": "ResourceOwnerPassword",
  "SecretKeySha256": "secret",
  "AllowAccessTokensViaBrowser": true,
  "RequireConsent": false,
  "AlwaysIncludeUserClaimsInIdToken": true,
  "AllowedScopes": [
    "public-api"
  ]
}
```

Jeśli żądanie */connect/token* będzie wysyłane bez pośrednictwa przeglądarki (Postman, *curl*), parametr *AllowAccessTokensViaBrowser* może mieć wartość *false*.

Jeśli zachodzi konieczność dodatkowego zabezpieczenia klienta używanego do uzyskania tokenu, można użyć parametru *RequireClientSecret* z wartością *true*. W takim przypadku w pliku *appsettings-protected.json* należy dodać wpis z zaszyfrowanym hasłem klienta:

```
"AppConfiguration": {  
  "Secrets": {  
    ...,  
    "Client-publicApiClient": "DVqipgOwCBz9D3JSi9nd9qJGiWzuhI1VZ6+wPH5Hn6sT+1CGONMismjywy/sHWxI"  
  }  
}
```

Informacje dotyczące szyfrowania hasła można uzyskać u producenta nAxiom.

Taka konfiguracja spowoduje, że w żądaniu tokenu konieczne będzie podanie hasła klienta jako wartości klucza *client_secret*.

WAŻNE

Po zmianie pliku *appsettings.json* konieczne jest ponowne uruchomienie witryny w programie Menedżer IIS lub w oprogramowaniu do zarządzania implementacją kontenerową (Kubernetes, OpenShift, Docker Desktop).

4. Konfiguracja profilu użytkownika

Użytkownik, który będzie zdalnie logował się do witryny nAxiom i następnie wysyłał żądania API musi mieć w swoim profilu w nAxiom włączony przełącznik Dostęp do Public API.

Dane podstawowe

Edycja profilu użytkownika

Nazwa użytkownika:*	jtester
Email:*	dboratyn@naxiom.com
Imię:*	Jerzy
Nazwisko:*	Tester
Aktywny:	☒
Dostęp do Internal Api:	☒
Dostęp do Public Api:	☒
Dostęp do Mobile Api:	☒

Profil użytkownika w nAxiom

Dodatkowo, korzystanie z endpointów */User/Save*, */User/AddNewUser* oraz endpointów w węzłach */PermissionDelegation* i */Permission* wymaga, aby użytkownik miał przypisaną predefiniowaną rolę biznesową *Administrator użytkowników PublicAPI*.

Data ważności:	26-02-2025	
Przełożony:	akajot (akajot@acme.com)	
Rola systemowa:	Klient uproszczony	
Globalne role biznesowe:	Administrator użytkowników PublicAPI ✕ Kontrola jakości ✕ Finanse ✕ ✕ Akceptant ✕	
Role biznesowe w jednostkach organizacyjnych:	Jednostka organizacyjna	Rola biznesowa
	Wydział I Organizacyjny > Biblioteka Wydział III Wywiadowczy > Referat I Techniki wywiadu	Pracownik ✕ ✕ Wybierz rolę

Rola administratora użytkowników PublicAPI

nAxiom zezwoli na administrowanie profilami użytkowników pod warunkiem, że dany użytkownik będzie miał przypisaną rolę biznesową o kodzie `USER_ADMIN_PUBLICAPI`. To oznacza, że nazwę predefiniowanej roli i jej przypisanie do aplikacji można zmienić.

Edycja roli ?

Aplikacja:	BaseApp
Moduł:	BaseModule
Kod:	USER_ADMIN_PUBLICAPI
Nazwa:	Administrator użytkowników PublicAPI

Edycja roli administratora użytkowników PublicAPI

Podczas wykonywania żądań do PublicAPI sprawdzane są także odpowiednie uprawnienia użytkowników w nAxiom: uprawnienia ACL w przypadku operacji na dokumentach oraz uprawnienia PBA, np. do edycji profili użytkowników. Brak odpowiednich uprawnień będzie skutkował odmową wykonania żądania z błędem HTTP 403.

5. Uzyskiwanie tokenu

Do autoryzowania żądań wysyłanych do PublicAPI konieczny jest token przypisywany podczas logowania użytkownika. Aby go uzyskać, należy wysłać do serwisu *auth* danej witryny nAxiom żądanie `{url}/connect/token`. Konfiguracja żądania jest następująca:

- Typ żądania: *POST*
- Sekcja *Body*, parametry typu *x-www-form-urlencoded*:
 - `grant_type`: ***password***
 - `username`: nazwa użytkownika
 - `password`: hasło użytkownika
 - `scope`: ***public-api***
 - `client_secret`: hasło klienta używanego do wysłania żądania; hasło jest wymagane tylko wtedy, gdy w konfiguracji klienta włączono atrybut *RequireClientSecret*.
 - `client_id`: identyfikator klienta zgodny z deklaracją w pliku *appsettings.json* serwisu *auth*; w tym przypadku ***publicApiClient***

W programie Postman żądanie i odpowiedź mogą wyglądać jak na ilustracji poniżej.

UWAGA

W przypadku witryny nAxiom na hoście z certyfikatem samopodpisanym polecenie `curl` wymaga użycia parametru `-k` w celu wyłączenia sprawdzania certyfikatu.

```
curl -k --location 'https://localhost:1414/connect/token' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--data-urlencode 'grant_type=password' \
--data-urlencode 'username=admin' \
--data-urlencode 'password=!Q2w3e4r' \
--data-urlencode 'scope=public-api' \
--data-urlencode 'client_secret=secret' \
--data-urlencode 'client_id=publicApiClient'
```

Uzyskaną w odpowiedzi wartość parametru `access_token` należy używać w żądaniach wysyłanych do PublicAPI.

Postman:

- W definicji żądania na karcie `Authorization` rozwiń listę `Type` i wybierz wartość `Bearer Token`.
- Wklej wartość tokena w polu `Token`

curl:

- Użyj wartości tokenu w parametrze `--header` jak w przykładzie poniżej:

```
curl -k --location 'https://localhost:1414/publicapi/...' \
--header 'Authorization: Bearer eyJhbGciOiJSUzI1NiIsImtpZCI6IjUyMkMzE2REU5...'
```

Czas ważności tokenu wynosi 3600 s.

6. Przykłady żądań do PublicAPI

6.1. [GET]/PermissionDelegation/Get

Ten endpoint zwraca dane zastępstwa o podanym identyfikatorze. Odpowiedź zawiera pełne dane z profili użytkowników zastępowanego i zastępującego. Żądanie i fragment odpowiedzi przedstawiono na ilustracji.

PUBLIC / **/PermissionDelegation/Get**

Save

Share

GET

https://localhost:1414/publicapi/PermissionDelegation/Get?id=10

Send

Params
 Auth
 Headers (8)
 Body
 Scripts
 Tests
 Settings

Cookies

Query Params

☒	Key	Value	Description	...	Bulk Edit
☒	id	10			
	Key	Value	Description		

Body

Cookies

Headers (11)

Test Results

200 OK 729 ms 3.17 KB

Save as example
 ...

Pretty

Raw

Preview

Visualize

JSON

```

101     "indefinitePassword": false
102   },
103   "creationDate": "2024-06-07T12:13:25.6008698",
104   "dateFrom": "2024-06-07T00:00:00",
105   "dateTo": "2024-06-07T23:59:59",
106   "allowACLAccess": false,
107   "permittedRoles": [
108     {
109       "permissionDelegationId": 10,
110       "roleId": 18,
111       "role": {
112         "id": 18,
113         "externalSystemId": null,
114         "externalSystemSource": null,
115         "type": 1,
116         "code": "ADMIN2",
117         "name": "Administrator 2",
118         "source": 0,
119         "moduleDefinition": null,
120         "moduleDefinitionId": 5,
121         "userProfileRoles": [],
122         "roleMenus": [],
123         "widgetRoles": null,
124         "dashboardRoles": null

```

Endpoint /PermissionDelegation/Get

curl:

```
curl -k --location 'https://localhost:1414/publicapi/PermissionDelegation/Get?id=10' \
--header 'Authorization: Bearer eyJhbGciOiJSUzI1NiIsImtpZCI6...'
```

6.2. [GET]/Document/Get

Endpoint zwraca dane instancji dokumentu dla podanej definicji dokumentu. Wymaga podania identyfikatorów definicji dokumentu i instancji dokumentu.

The screenshot shows a REST client interface with the following details:

- Method:** GET
- URL:** https://localhost:1414/publicapi/Document/Get?documentDefinitionId=26&recordId=36
- Params:** documentDefinitionId=26
- Status:** 200 OK
- Time:** 52 ms
- Size:** 1003 B
- Response Body (JSON):**

```
[{"Transition":120,"Status":58,"Type1":null,"Type2":null,"Type3":null,"Id":36,"Code":"0035\\2024",
  "Organization":null,"RowVersion":2,"UserId":"18EF4E5E-11EE-41CD-5A9D-08D8DA52C48B",
  "CreateDate":"2024-04-15T14:17:49.450","UpdatedBy":"18EF4E5E-11EE-41CD-5A9D-08D8DA52C48B",
  "UpdatedDate":"2024-04-15T14:35:24.943","ACLId":"35A71E0C-F45A-4498-8C3D-65963BFCC17F",
  "Blocked":false,"BlockUser":null,"LastEntryDate":null,"ArchiveStatus":1,"IsAnonimized":false,
  "Marka":"wołga","Model":"wołga","RokProdukcji":1956,"Przebieg":123456789,"Cena":123456789.00,
  "Link":null}]
```

Endpoint /Document/Get

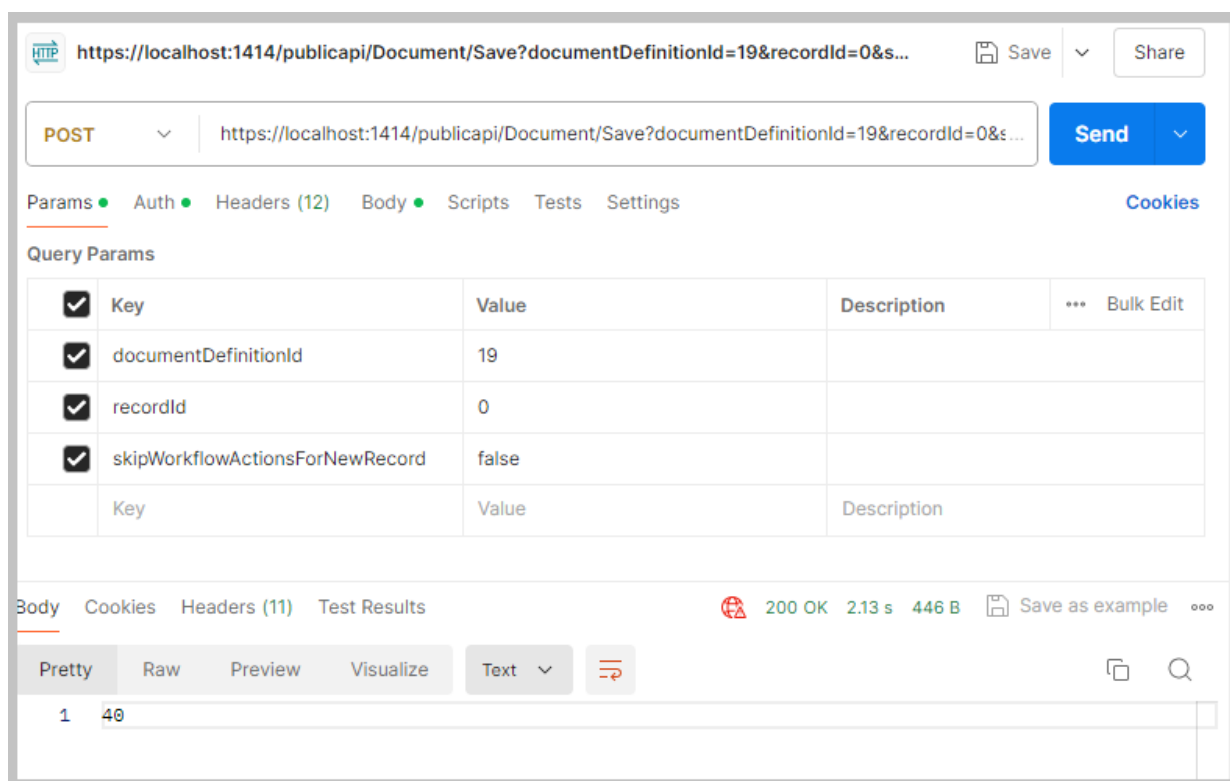
curl:

```
curl -k --location \
--request GET 'https://localhost:1414/publicapi/Document/Get?documentDefinitionId=26&recordId=37' \
--header 'Content-Type: application/json' \
--header 'Authorization: Bearer eyJhbGciOiJSUzI1NiIsImtpZCI6...' \
--data '{"string"}'
```

6.3. [POST]/Document/Save

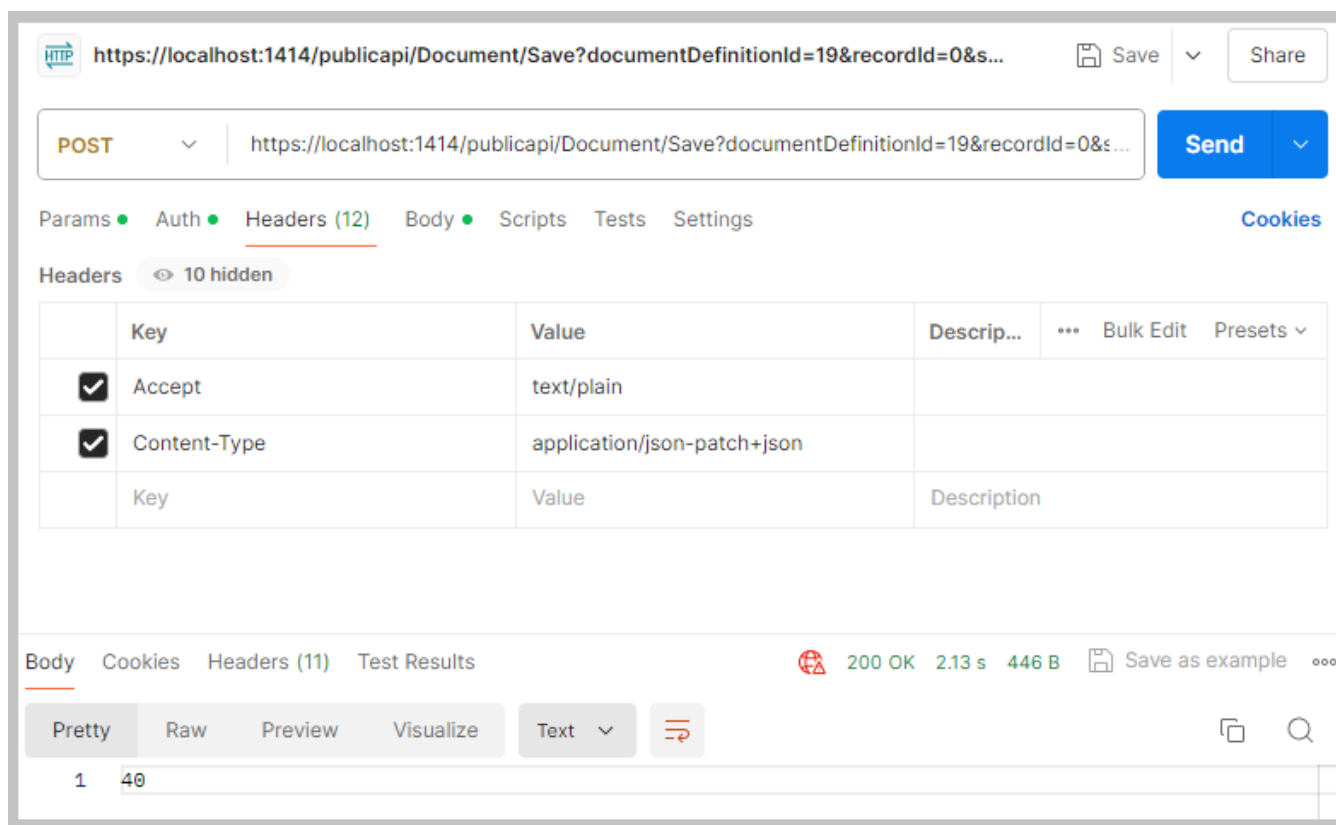
Endpoint dodaje nowy dokument lub aktualizuje istniejący. Wymaga podania następujących parametrów:

- *documentDefinitionId*: identyfikator definicji dokumentu.
- *recordId*: identyfikator zmienianej instancji dokumentu lub *0* w przypadku dodawania nowego dokumentu.
- *skipWorkflowActionsForNewRecord*: flaga pomijania akcji wykonywanych podczas tworzenia nowego dokumentu; domyślnie *false*.



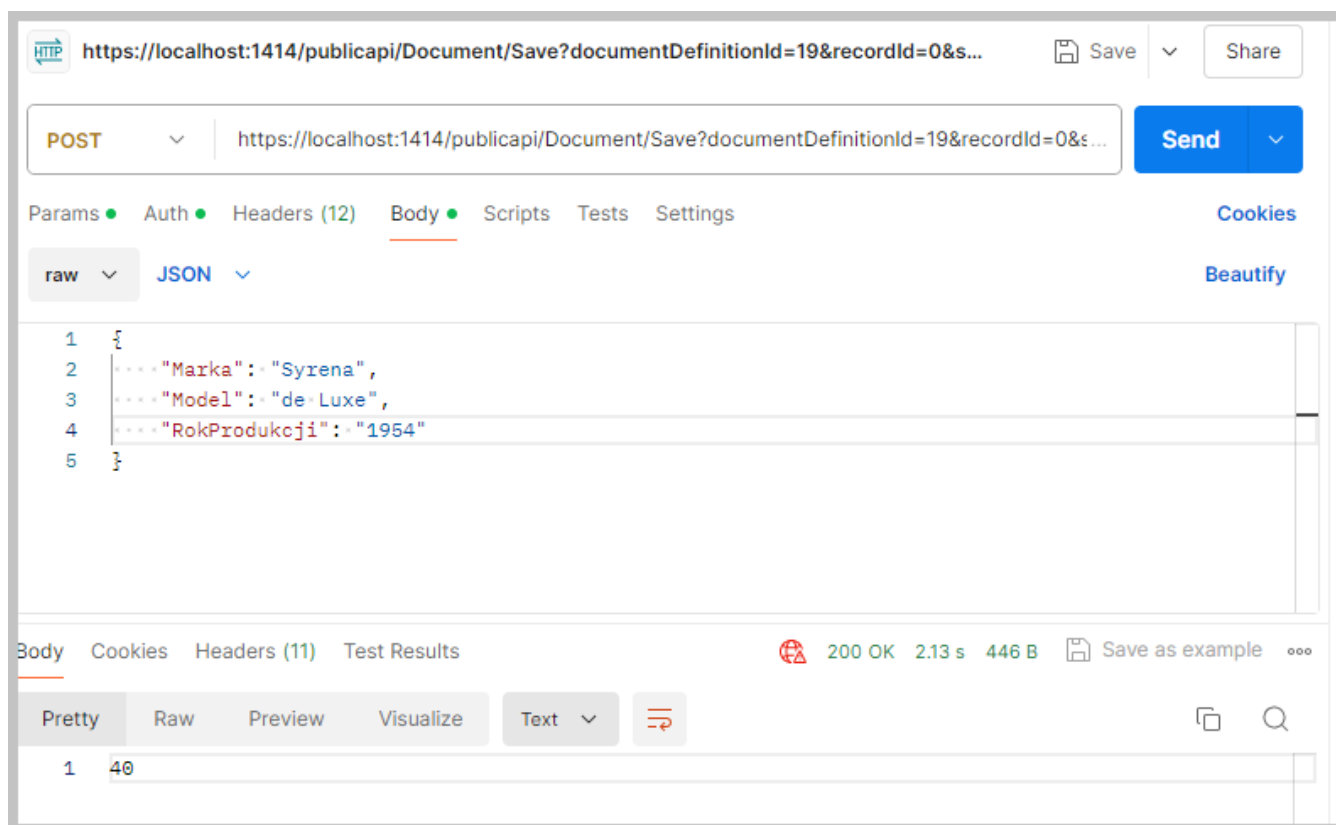
Endpoint /Document/Save - parametry

W żądaniu należy określić nagłówki jak na ilustracji poniżej.



Endpoint /Document/Save - nagłówki

W sekcji *body* żądania można określić wartości pól dokumentu w formacie JSON jak na ilustracji poniżej.



Endpoint /Document/Save - body

W przypadku pomyślnego dodania nowego dokumentu w odpowiedzi zwracany jest jego identyfikator.

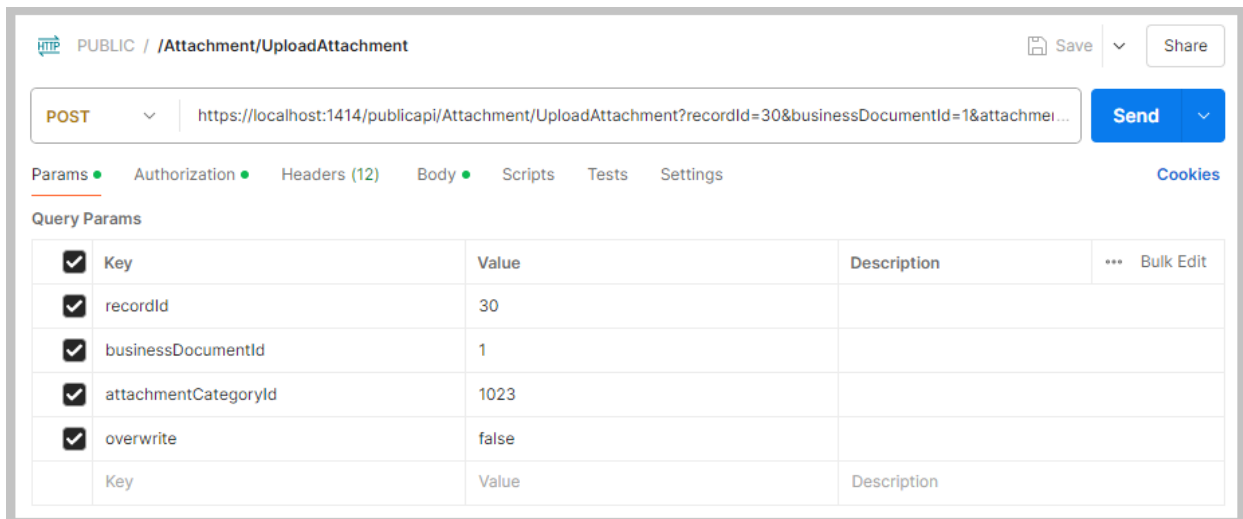
curl:

```
curl -k --location 'https://localhost:1414/publicapi/Document/Save?documentDefinitionId=19&recordId=0&skipWorkflowActionsForNewRecord=false' \
--header 'Accept: text/plain' \
--header 'Content-Type: application/json-patch+json' \
--header 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsImt...' \
--data '{
  "Marka": "Syrena",
  "Model": "105 de Luxe",
  "RokProdukcji": "1964"
}'
```

6.4. [POST]/Attachment/UploadAttachment

Dodaje załącznik do wskazanej kategorii załączników określonego dokumentu. W zależności od wartości parametru *overwrite* nadpisuje istniejący plik lub dodaje nową wersję. Wymaga podania następujących parametrów:

- *recordId*: identyfikator zmienianej instancji dokumentu lub 0 w przypadku dodawania nowego dokumentu.
- *businessDocumentId*: identyfikator definicji dokumentu.
- *attachmentCategoryId*: identyfikator kategorii załączników; jeśli zostanie pominięty, załącznik zostanie dodany do kategorii domyślnej.
- *overwrite*: decyduje, czy istniejący plik załącznika o takiej samej nazwie ma zostać nadpisany (*true*), czy załącznik ma zostać dodany jako nowa wersja (*false*).



HTTP PUBLIC / Attachment/UploadAttachment

POST https://localhost:1414/publicapi/Attachment/UploadAttachment?recordId=30&businessDocumentId=1&attachmentCategoryId=1023&overwrite=false

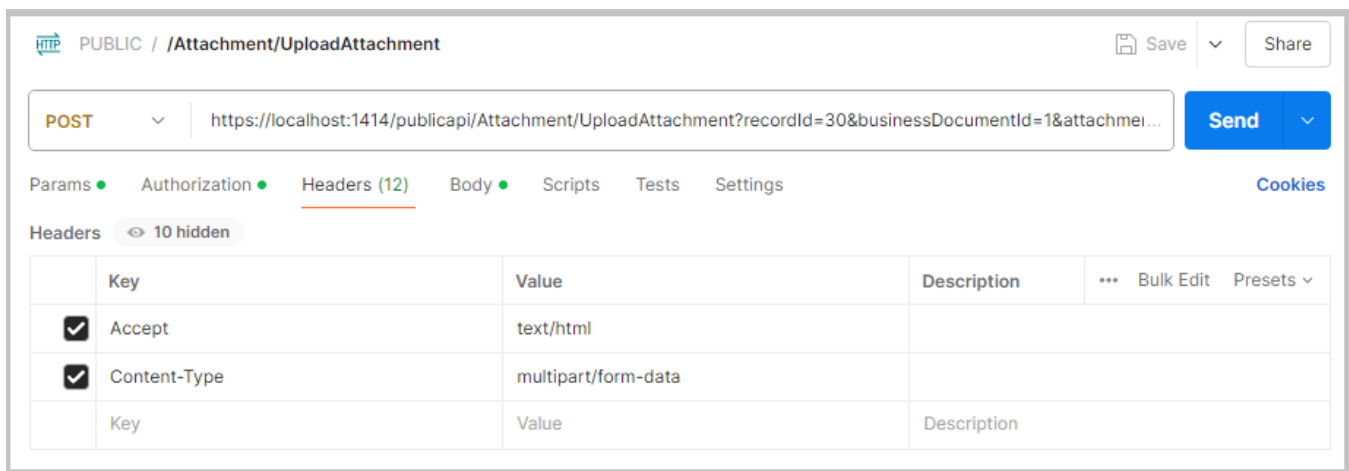
Params Authorization Headers (12) Body Scripts Tests Settings Cookies

Query Params

☒	Key	Value	Description	...	Bulk Edit
☒	recordId	30			
☒	businessDocumentId	1			
☒	attachmentCategoryId	1023			
☒	overwrite	false			
	Key	Value	Description		

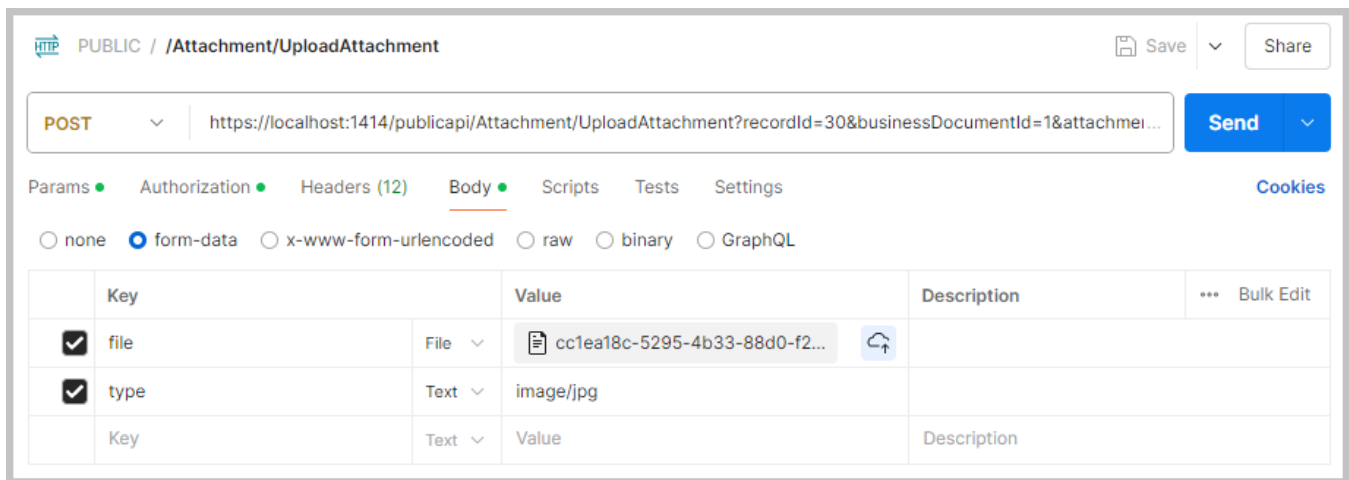
Endpoint /Attachment/UploadAttachment - parametry

W sekcji nagłówek należy ustawić parametry jak na ilustracji poniżej.



Endpoint /Attachment/UploadAttachment - nagłówki

Plik załącznika należy załączyć w sekcji *body*, wybierając typ *form-data* i dodając parametry *file* i *type*. Dla pierwszego należy wybrać z listy rozwijanej rodzaj *File*, co umożliwi wybranie pliku załącznika z dysku w kolumnie wartości. W programie Postman plik musi znajdować się w katalogu roboczym definiowanym w ustawieniach. W przypadku polecenia *curl* plik musi znajdować się w bieżącym katalogu, w którym jest uruchamiane polecenie.



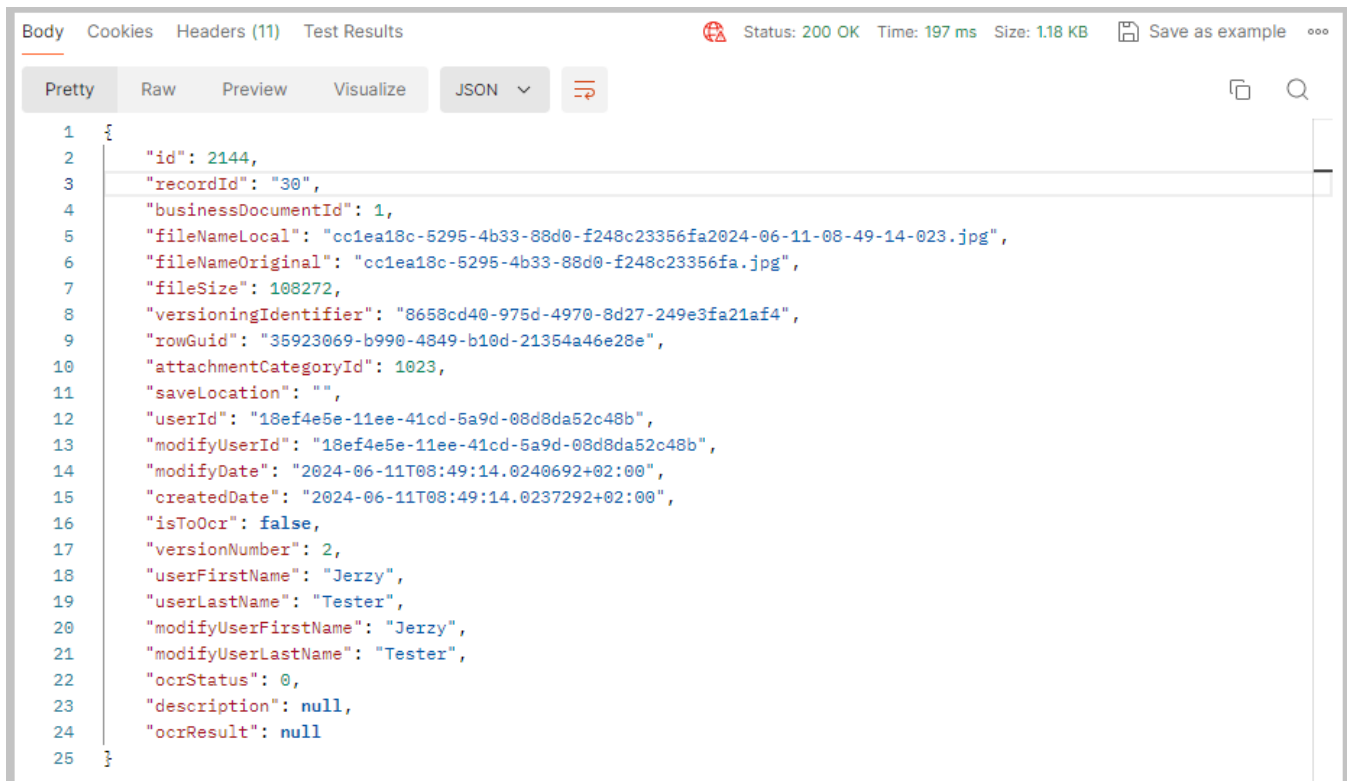
Endpoint /Attachment/UploadAttachment - body

curl:

```
curl -k --location 'https://localhost:1414/publicapi/Attachment/UploadAttachment?recordId=30&businessDocumentId=1&attachmentCategoryId=1023&overwrite=false' \
--header 'Accept: text/html' \
```

```
--header 'Content-Type: multipart/form-data' \
--header 'Authorization: Bearer eyJhbGciOiJSUzI1NiIsImtpZCI6...' \
--form 'file=@"cc1ea18c-5295-4b33-88d0-f248c23356fa.jpg"' \
--form 'type="image/jpg"'
```

Odpowiedź zwracana przez endpoint po pomyślnym wykonaniu przedstawiono na ilustracji poniżej.

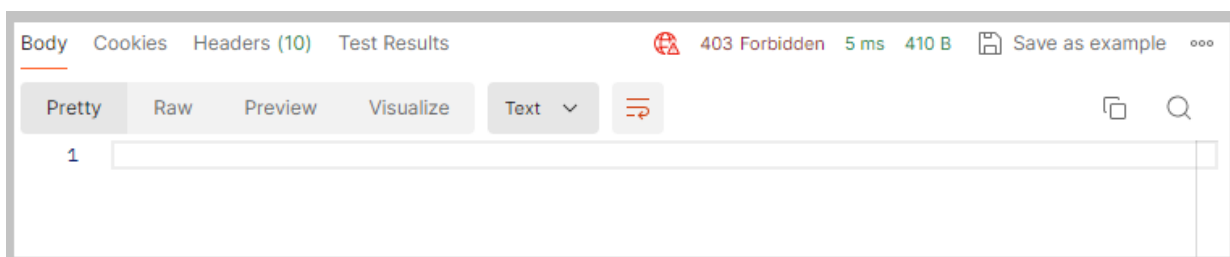


Endpoint /Attachment/UploadAttachment - response

7. Błędy braku uprawnień

Kiedy użytkownik wysyłający żądanie nie ma odpowiednich uprawnień, w odpowiedzi zostanie zwrócony błąd z komunikatem lub bez dodatkowego komunikatu. Poniżej przedstawiono kilka podstawowych przypadków.

1. Użytkownik nie ma włączonego dostępu do PublicAPI w profilu użytkownika.



Brak dostępu do PublicAPI

2. Użytkownik nie ma roli *Administrator użytkowników PublicAPI*.



Brak roli administratora PublicAPI

3. Użytkownik nie ma uprawnień ACL do modyfikowania dokumentu.

```
1 [
2   {
3     "propertyName": "",
4     "errorMessage": "Api.BusinessModule.ErrorOnSavingDocument",
5     "attemptedValue": null,
6     "customState": null,
7     "severity": 0,
8     "errorCode": null,
9     "formattedMessageArguments": null,
10    "formattedMessagePlaceholderValues": {}
11  },
12  {
13    "propertyName": "",
14    "errorMessage": "Api.PermissionHelper.NoAccess",
15    "attemptedValue": null,
16    "customState": null,
17    "severity": 0,
18    "errorCode": null,
19    "formattedMessageArguments": null,
20    "formattedMessagePlaceholderValues": {
21      "businessDocumentCode": "USZ",
22      "id": "40",
23      "permission": "Update"
24    }
25  }
26 ]
```

Brak uprawnień ACL do dokumentu

4. Użytkownik nie ma uprawnień PBA do wykonania danej czynności.

```
1 [
2   {
3     "propertyName": "",
4     "errorMessage": "Api.Shared.NoPermissions",
5     "attemptedValue": null,
6     "customState": null,
7     "severity": 0,
8     "errorCode": null,
9     "formattedMessageArguments": null,
10    "formattedMessagePlaceholderValues": {}
11  }
12 ]
```

Brak uprawnień PBA

8. Znane problemy

Obecnie (wersja nAxiom 1.14.0.3) znane są następujące problemy dotyczące endpointów PublicAPI.

1. */Attachment/RemoveAttachment*: usunięcie wersji załącznika powoduje całkowite usunięcie załącznika z dokumentu.
2. */Attachment/UploadAttachment*: nie jest walidowany identyfikator dokumentu, do którego ma zostać dodany załącznik; w przypadku podania id dokumentu, który nie istnieje, wykonanie żądania powiedzie się i informacje o załączniku trafią do tabeli *core.Attachments*, a sam załącznik do repozytorium danej kategorii. Jednak wobec braku dokumentu, zarządzanie załącznikiem i wyświetlanie jego podglądu na formularzu nie będzie możliwe.
3. */Document/CreateNewSchema*: endpoint działa nieprawidłowo, jest przeznaczony do wycofania.
4. */Document/GetFormTransitions*: endpoint za każdym razem zwraca status 400.
5. */Document/GetTransitions*: endpoint za każdym razem zwraca status 400.
6. */Permission/GetAll*: w zwracanej odpowiedzi brakuje niektórych atrybutów (nazwy modułu, id aplikacji, nazwy aplikacji, ról z dostępem do danej pozycji menu).
7. */OrganizationStructure/GetOrganizationById*: nie zwraca ról użytkownika w jednostce organizacyjnej.
8. */OrganizationStructure/SaveOrganization*: umożliwia zapisanie kodu jednostki organizacyjnej, który zawiera nieprawidłowe znaki; taki kod powoduje błędy w działaniu aplikacji nAxiom.
9. */User/GetById*: zwraca status 200 OK także wtedy, gdy użytkownik o podanym id nie istnieje.
10. */User/Save*: jeśli nie zostanie podana rola systemowa, żądanie zwraca status 400, ale użytkownik jest dodawany do tabeli *auth.AspNetUSers*.
11. */User/Save*: jeśli parametr *recordId* zostanie podany także w sekcji *body* żądania, wartość z *body* przesłania wartość podaną w parametrze.

Ponadto, ustawienia systemu są wczytywane do PublicAPI w momencie uruchamiania, dlatego ich zmiana wymaga restartowania puli *publicapi*