



*n*Axiom

Generowanie aplikacji mobilnej dla witryny nAxiom

Wersja nAxiom: 1.8



Spis treści

1. Streszczenie.....	3
2. Aplikacja mobilna dla systemu Android.....	3
2.1. Wymagania.....	3
2.2. Uruchomienie generatora.....	3
2.3. Opis dostępnych parametrów.....	4
2.3.1. <code>-unique_app_id</code>	4
2.3.2. <code>-app_name</code>	4
2.3.3. <code>-app_version_number</code>	5
2.3.4. <code>-icons_directory_path</code>	5
2.3.5. <code>-logo_path</code>	5
2.3.6. <code>-config_path</code>	5
2.3.7. <code>-key_store_path:</code>	6
2.3.8. <code>-key_password</code>	6
2.4. Podsumowanie.....	6
3. Aplikacja mobilna dla systemu iOS.....	7
3.1. Wymagania.....	7
3.2. Uruchomienie generatora.....	7
3.3. Opis dostępnych parametrów.....	8
3.3.1. <code>-unique_app_id</code>	8
3.3.2. <code>-app_name</code>	8
3.3.3. <code>-app_version_number</code>	8
3.3.4. <code>-icons_directory_path</code>	8
3.3.5. <code>-logo_path</code>	10
3.3.6. <code>-config_path</code>	11
3.3.7. <code>-distribution_certificate_name</code>	11
3.3.8. <code>-provisioning_profile_path</code>	11
3.4. Znane problemy.....	12
3.5. Podsumowanie.....	13
4. Plik konfiguracyjny aplikacji.....	13

1. Streszczenie

Tematem niniejszej instrukcji jest procedura tworzenia pakietu aplikacji mobilnej dla witryny nAxiom. Taki pakiet można dystrybuować za pośrednictwem sklepów Google Play i Apple Store. Z czasem planowane jest dodanie do aplikacji mobilnej obsługi natywnych funkcjonalności urządzeń komórkowych, np. aparatu oraz rozszerzenie zestawu gestów do obsługi aplikacji. W ogólności pakiet aplikacji mobilnej jest tworzony za pomocą skryptu uruchamianego w systemie MacOS.

2. Aplikacja mobilna dla systemu Android

2.1. Wymagania

- System macOS w wersji min. 12.0 Monterey.
- Zainstalowany pakiet Java JDK.
- Nadanie uprawnień do odczytu, zapisu oraz wykonywania (rwx) dla pliku generatora `mobile_Android_generator.run`:

```
chmod 700 mobile_Android_generator.run
```

- Konto deweloperskie *Google Play Console*, o ile aplikacja ma być dystrybuowana przez Google Play.

2.2. Uruchomienie generatora

1. Umieścić plik generatora `mobile_Android_generator.run` w katalogu użytkownika (folder `/Users/{userName}`). Plik musi mieć uprawnienia `rwx`; patrz poprzednia sekcja.
2. Z poziomu terminala uruchomić generator, za pomocą komendy z podanymi parametrami jak w przykładzie:

UWAGA: Przekazywane do generatora jako parametry ścieżki do plików statycznych lub folderów muszą być ścieżkami bezwzględnymi. Wszystkie parametry generatora są wymagane.

```
./mobile_Android_generator.run \  
-unique_app_id 'com.example.exampleapp' \  
-app_name 'EXAMPLE NAME' \  
-app_version_number 1 \  
-icons_directory_path '/Users/{userName}/input_parameters/app_icons' \  
-logo_path '/Users/{userName}/input_parameters/start_page_logo.svg' \  
-config_path '/Users/{userName}/input_parameters/config.json' \  
-key_store_path '/Users/{userName}/input_parameters/keystore.jks' \  
-key_password 'passwordToKeyStore'
```

Generator tworzy dwa pliki wynikowe:

- pakiet Android App Bundle (AAB) gotowy do wdrożenia w Google Play,
- pakiet Android Package Kit (APK), który można zainstalować bezpośrednio na urządzeniu Android, z pominięciem sklepu Google Play.

2.3. Opis dostępnych parametrów

2.3.1. `-unique_app_id`

- Unikatowy identyfikator aplikacji, potrzebny do jej wdrożenia w Google Play. Musi zostać nadany w momencie dystrybucji pierwszej wersji aplikacji. Po opublikowaniu aplikacji, nie można go zmienić. Identyfikator musi zaczynać się od członu *com.* i zawierać co najmniej dwa człony rozdzielone kropkami, np. *com.example.exampleapp*. Identyfikator musi być unikalny w Google Play. Próba użycia już wykorzystywanego identyfikatora jest niedozwolona.
- Dodatkowe informacje:
<https://developer.android.com/studio/build/configure-app-module#set-application-id>.
- Dopuszczalna wartość: *text*, format *com.**, np. *com.example.exampleapp*.

2.3.2. `-app_name`

- Nazwa aplikacji, widoczna dla użytkownika po jej zainstalowaniu.
- Dopuszczalna wartość: *text*, np. *Example App*

2.3.3. `-app_version_number`

- Numer wersji aplikacji reprezentujący wartość liczbową, wymagany do wdrożenia i wersjonowania aplikacji w Google Play. Każda kolejna wersja aplikacji, która ma zostać opublikowana, powinna mieć numer większy od poprzedniego. Przy pierwszej publikacji zalecane jest ustalenie numeru początkowego wynoszącego np. 1. Każda kolejna wersja powinna być większa od aktualnie dostępnej w Konsoli Google Play. W podanym przykładzie powinno to być co najmniej 2 lub więcej.
- Dodatkowe informacje:
<https://developer.android.com/studio/publish/versioning>.
- Dopuszczalna wartość: *number*, np. 1.

2.3.4. `-icons_directory_path`

- Ścieżka bezwzględna do folderu zawierającego ikony aplikacji. Muszą one być w formacie *PNG* oraz mieć następujące nazwy i rozdzielczości:
 - mipmap-hdpi - 72x72
 - mipmap-mdpi - 48x48
 - mipmap-xhdpi - 96x96
 - mipmap-xxhdpi - 144x144
 - mipmap-xxxhdpi - 192x192
- Dodatkowe informacje:
<https://developer.android.com/training/multiscreen/screendensities#TaskProvideAltBmp>;
<https://developer.android.com/distribute/google-play/resources/icon-design-specifications>
- Dopuszczalna wartość: *absolute directory path*, np.
`/Users/{userName}/input_parameters/app_icons`

2.3.5. `-logo_path`

- Ścieżka bezwzględna do pliku logo aplikacji w formacie **.svg*; logo jest wyświetlane na ekranie startowym.
- Dopuszczalna wartość: *absolute file path*, np.
`/Users/{userName}/input_parameters/start_page_logo.svg`

2.3.6. `-config_path`

- Ścieżka bezwzględna do pliku konfiguracyjnego w formacie *JSON*, którego struktura powinna być zgodna z opisem w sekcji [Plik konfiguracyjny aplikacji](#).
- Dopuszczalna wartość: *absolute file path*, np. `/Users/{userName}/input_parameters/config.json`

2.3.7. -key_store_path:

- Ścieżka bezwzględna do wygenerowanego klucza, wymaganego do podpisania oraz dystrybucji aplikacji w Google Play.
- Dodatkowe informacje:
<https://developer.android.com/studio/publish/app-signing#generate-key>.
- Dopuszczalna wartość: *absolute file path*, np. */Users/{userName}/input_parameters/keystore.jks*

2.3.8. -key_password

- Hasło do podanego wyżej klucza. W przypadku, gdy podane hasło jest niepoprawne zostanie zalogowany komunikat o błędzie następującej treści:

```
Failed to load signer "signer #1"  
java.io.IOException: keystore password was incorrect
```

Pomimo tego, plik wynikowy aplikacji mobilnej zostanie wygenerowany, ale nie będzie możliwe przesłanie go poprzez konsolę Google Play. Zostanie wyświetlony komunikat o błędzie:

```
The Android App Bundle was not signed.
```

- Dodatkowe informacje:
<https://developer.android.com/studio/publish/app-signing#generate-key>.
- Dopuszczalna wartość: *text*

2.4. Podsumowanie

Po wykonaniu skryptu z podaniem odpowiednich parametrów zostaną utworzone dwa pliki wynikowe aplikacji mobilnej (w tej samej lokalizacji, w której znajduje się plik generatora *mobile_Android_generator.run*) o nazwach:

- *output.aab*: gotowy do wdrożenia w Google Play,
- *output.apk*: gotowy do wdrożenia bezpośrednio na urządzeniu mobilnym.

3. Aplikacja mobilna dla systemu iOS

3.1. Wymagania

- System macOS w wersji min. 12.0 Monterey.
- xCode w wersji min. 13.2.0.
- Nadanie uprawnień do odczytu, zapisu oraz wykonywania (rwx) dla pliku generatora `mobile_iOS_generator.run`:

```
chmod 700 mobile_iOS_generator.run
```

- Konto deweloperskie *Apple Developer Program* (<https://developer.apple.com/programs>).
- Certyfikat dystrybucyjny (*Distribution Certificate*) wraz z prywatnym kluczem użytym do jego wygenerowania, który będzie użyty do podpisania aplikacji, zaimportowane w pęku kluczy (*Keychain*).
- Profil *provisioning profile* powiązany z wspomnianym wyżej certyfikatem (**.mobileprovision*).

3.2. Uruchomienie generatora

1. Umieścić plik generatora `mobile_iOS_generator.run` w katalogu użytkownika (folder `/Users/{userName}`). Plik musi mieć uprawnienia *rwx*, patrz poprzednia sekcja.
2. Z poziomu terminala uruchomić generator, za pomocą komendy z podanymi parametrami jak w przykładzie:

UWAGA: Przekazywane do generatora jako parametry ścieżki do plików statycznych lub folderów muszą być ścieżkami bezwzględnymi. Wszystkie parametry generatora są wymagane.

```
./mobile_iOS_generator.run \
-unique_app_id `com.example.exampleapp` \
-app_name 'EXAMPLE NAME' \
-app_version_number 1 \
-icons_directory_path '/Users/{userName}/input_parameters/app_icons' \
-logo_path '/Users/{userName}/input_parameters/start_page_logo.svg' \
```

```
-config_path '/Users/{userName}/input_parameters/config.json' \  
-distribution_certificate_name 'Apple Distribution: Example Ltd. (XC5D4E1234)' \  
-provisioning_profile_path  
  '/Users/{userName}/input_parameters/ExampleProvisioningProfile.mobileprovision' \  

```

3.3. Opis dostępnych parametrów

3.3.1. `-unique_app_id`

- Unikatowy identyfikator aplikacji, potrzebny do jej wdrożenia w AppStore. Musi zostać nadany w momencie dystrybucji pierwszej wersji aplikacji. Po opublikowaniu aplikacji, nie można go zmienić. Identyfikator musi zaczynać się od członu *com.* i zawierać co najmniej dwa człony rozdzielone kropkami, np. *com.example.exampleapp*. Identyfikator musi być unikalny w AppStore. Próba użycia już wykorzystywanego identyfikatora jest niedozwolona.
- Dodatkowe informacje:
https://developer.apple.com/documentation/bundleresources/information_property_list/cfbundleidentifier.
- Dopuszczalna wartość: *text*, format *com.**, np. *com.example.exampleapp*.

3.3.2. `-app_name`

- Nazwa aplikacji, widoczna dla użytkownika po jej zainstalowaniu.
- Dopuszczalna wartość: *text*, np. *Example App*

3.3.3. `-app_version_number`

- Numer wersji aplikacji reprezentujący wartość liczbową, wymagany do wdrożenia i wersjonowania aplikacji w AppStore. Każda kolejna wersja aplikacji, która ma zostać opublikowana, powinna mieć numer większy od poprzedniego. Przy pierwszej publikacji zalecane jest ustalenie numeru początkowego wynoszącego np. *1*. Każda kolejna wersja powinna być większa od aktualnie dostępnej w konsoli Apple Developer Program. W podanym przykładzie powinno to być co najmniej 2 lub więcej.
- Dodatkowe informacje:
<https://help.apple.com/xcode/mac/current/#/devba7f53ad4>.
- Dopuszczalna wartość: *number*, np. *1*.

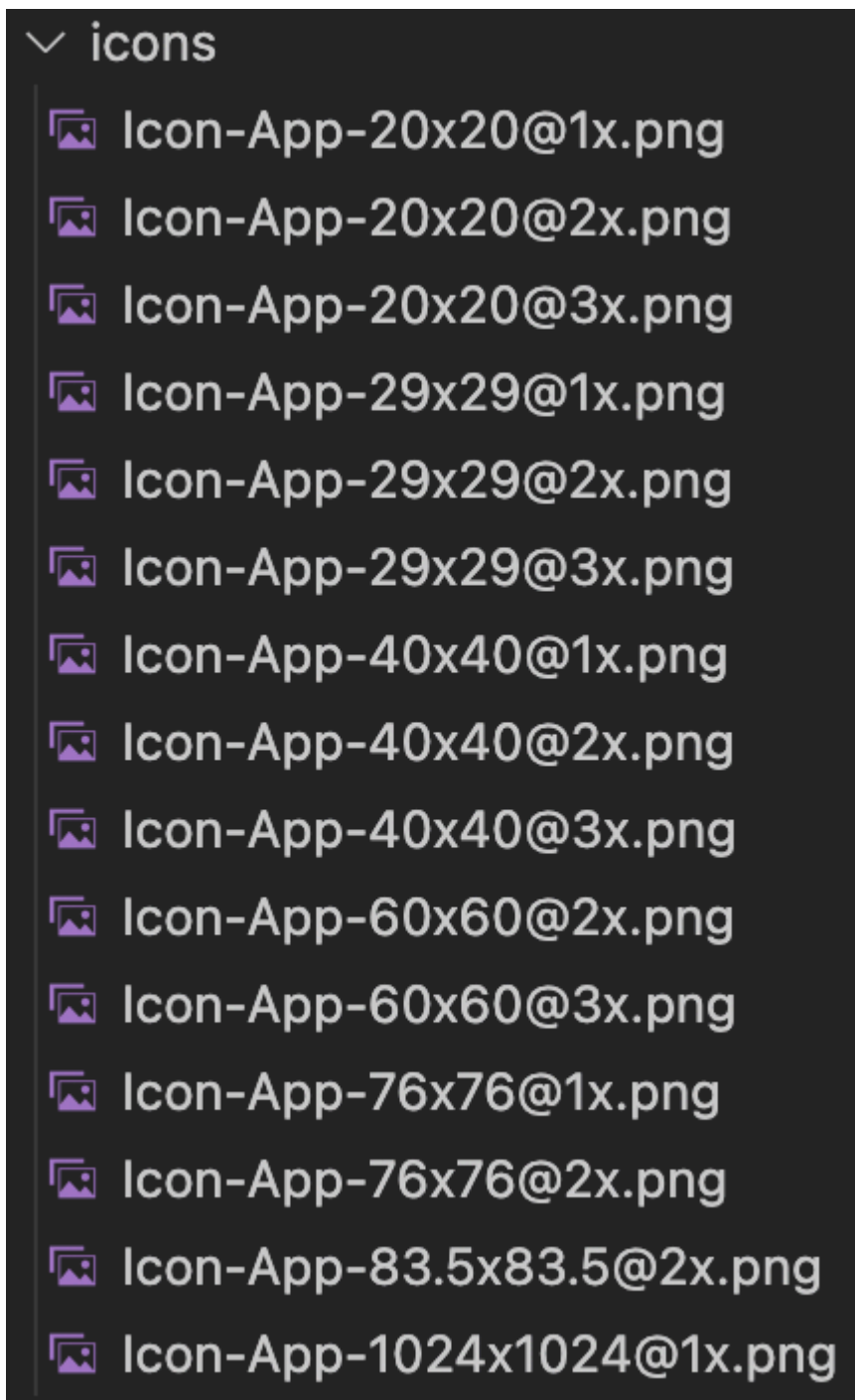
3.3.4. `-icons_directory_path`

- Ścieżka bezwzględna do folderu zawierającego ikony aplikacji. Muszą one być w formacie *PNG*

oraz mieć nazwy i rozdzielczości (podane w nazwie) jak na ilustracji poniżej.

Na przykład:

- Icon-App-20x20@1x.png; rozmiar 20 x 20 px
- Icon-App-20x20@2x.png; rozmiar 40 x 40 px
- Icon-App-20x20@3x.png; rozmiar 60 x 60 px
- Dodatkowe informacje:
<https://developer.apple.com/design/human-interface-guidelines/ios/icons-and-images/app-icon/>
- Dopuszczalna wartość: *absolute directory path*, np.
/Users/{userName}/input_parameters/app_icons



3.3.5. `-logo_path`

- Ścieżka bezwzględna do pliku logo aplikacji w formacie *.svg; logo jest wyświetlane na ekranie startowym.
- Dopuszczalna wartość: *absolute file path*, np.

/Users/{userName}/input_parameters/start_page_logo.svg

3.3.6. `-config_path`

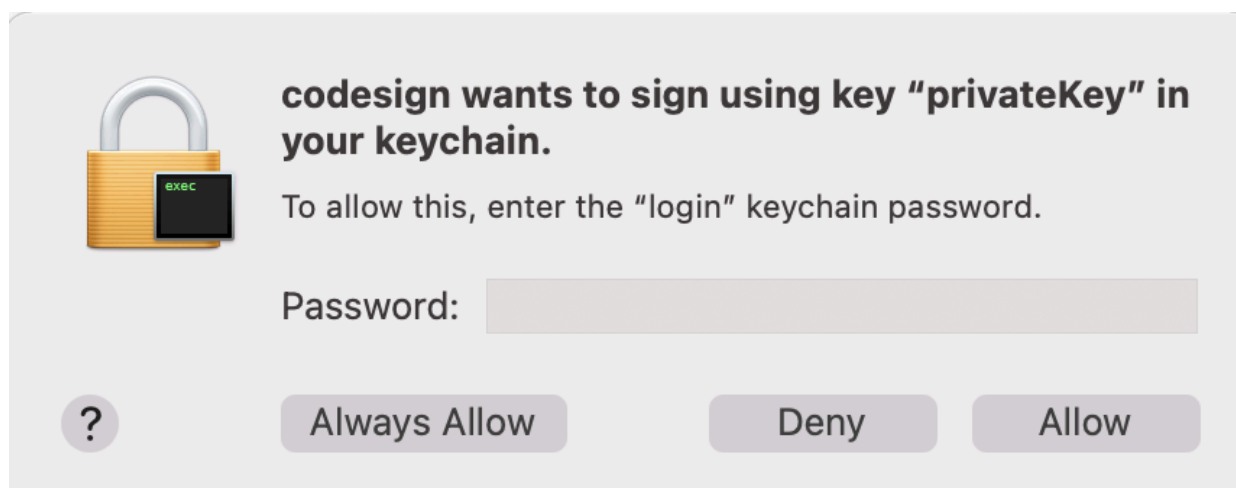
- Ścieżka bezwzględna do pliku konfiguracyjnego w formacie *JSON*, którego struktura powinna być zgodna z opisem w sekcji [Plik konfiguracyjny aplikacji](#).
- Dopuszczalna wartość: *absolute file path*, np. `/Users/{userName}/input_parameters/config.json`

3.3.7. `-distribution_certificate_name`

- Nazwa certyfikatu dystrybucyjnego (*Distribution Certificate*) wraz z powiązaniem z nim kluczem prywatnym, który powinien być zaimportowany w pęku kluczy (*Keychain*); certyfikat można wygenerować na koncie deweloperskim Apple:

<https://developer.apple.com/account/resources/certificates/list>

W momencie działania generatora będzie wymagane zezwolenie na dostęp do pęku kluczy (*Keychain*), jak w przykładzie poniżej.



- Dodatkowe informacje:
<https://developer.apple.com/support/certificates>
- Dopuszczalna wartość: *text*, np. *Apple Distribution: Example Ltd. (XC5D4E1234)*.

3.3.8. `-provisioning_profile_path`

- Ścieżka bezwzględna do pliku profilu *Provisioning Profile* aplikacji w AppStore, w formacie **.mobileprovision*. Można go wygenerować na koncie deweloperskim Apple (<https://developer.apple.com/account/resources/profiles/list>).

Powinien być powiązany z certyfikatem Distribution Certificate.

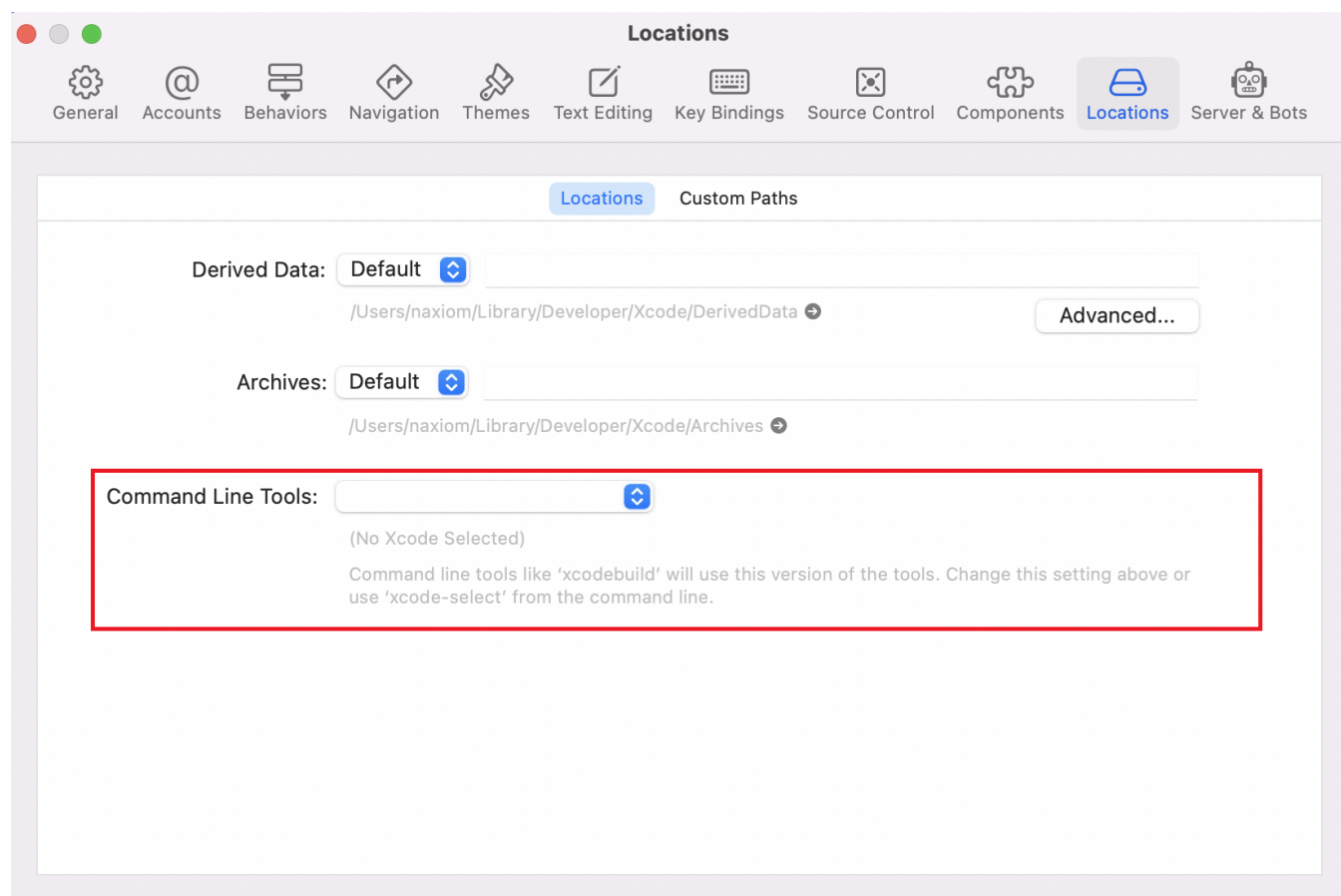
- Dopuszczalna wartość: *absolute file path*, np.
/Users/{userName}/input_parameters/ExampleProvisioningProfile.mobileprovision.

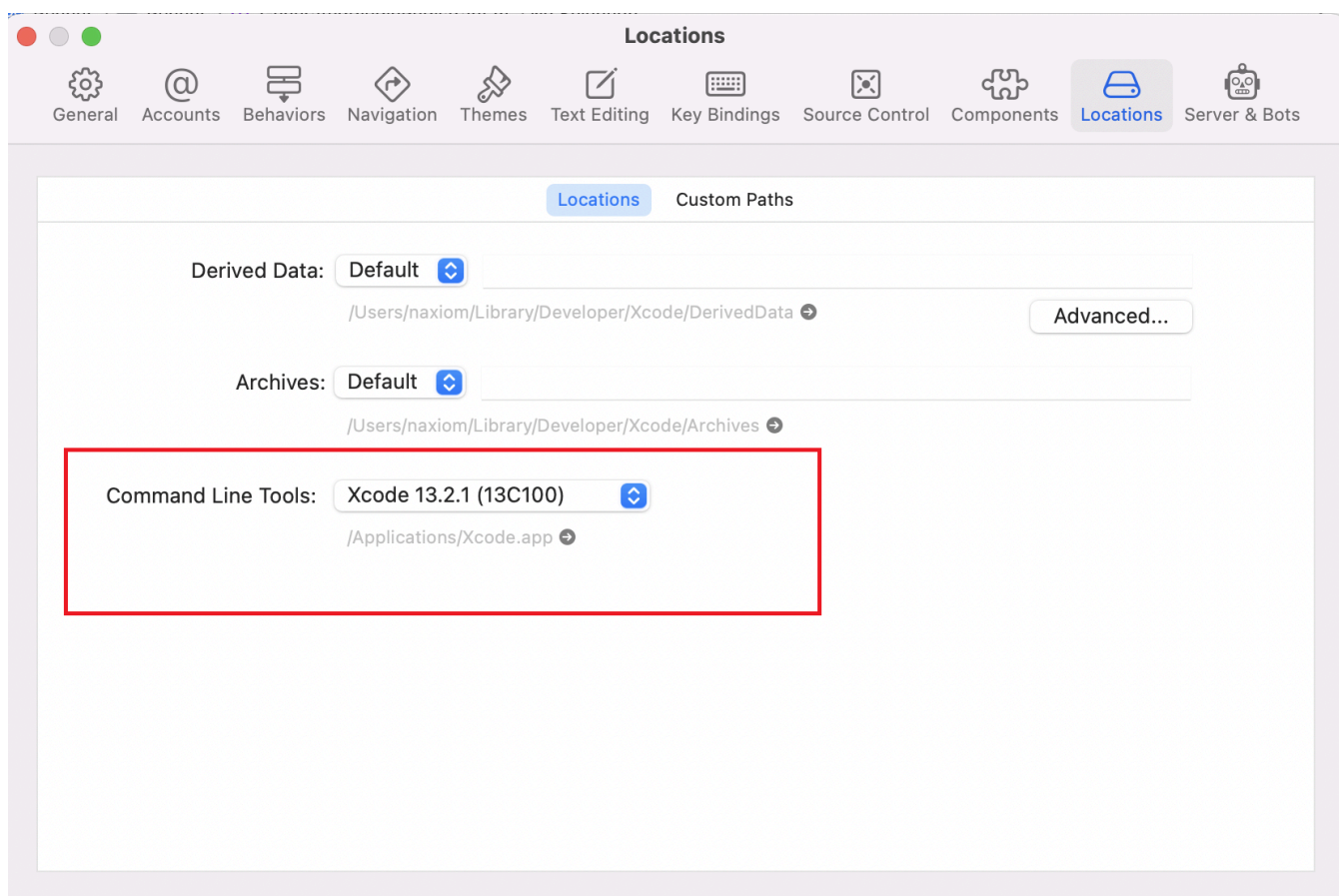
3.4. Znane problemy

W przypadku, gdy pomimo zainstalowania środowiska programistycznego Xcode widoczny jest następujący błąd:

```
Xcode should be installed before run.
```

należy upewnić się czy narzędzie *Xcode Command Line Tools* zostało poprawnie skonfigurowane. W tym celu należy z poziomu *Xcode/Preferences/Locations/Command Line Tools* zaznaczyć najnowszy dostępny pakiet.





3.5. Podsumowanie

Po wykonaniu skryptu z podaniem odpowiednich parametrów zostanie stworzony plik wynikowy aplikacji mobilnej (w tej samej lokalizacji, w której znajduje się plik generatora *mobile_iOS_generator.run*) o nazwie *output.ipa*, który jest gotowy do wdrożenia w AppStore. Można użyć do tego celu narzędzia *Transporter* dostępnego w AppStore dla macOS.

4. Plik konfiguracyjny aplikacji

```
{  
  "platformBaseUrl": "https://example.example.com/",  
  "connectionTimeout": 31000,  
  "applicationName": "przykładowa nazwa",  
  "applicationDescription": "przykładowy opis",  
  "primaryColor": "#bd0606",  
}
```

```
"onPrimaryColor": "#19fce2",  
"textColor": "#ffffff",  
"backgroundColor": "#000000",  
"errorColor": "#00ff15",  
"localization": "pl"  
}
```

Opis parametrów:

- `platformBaseUrl`: adres do witryny nAxiom, którą ma obsługiwać aplikacja mobilna.
- `connectionTimeout`: maksymalny czas oczekiwania na odpowiedź z API.
- `applicationName`: nazwa aplikacji widoczna na ekranie startowym.
- `applicationDescription`: opis aplikacji widoczny na ekranie startowym.
- `primaryColor`: przewodni kolor aplikacji w zapisie szesnastkowym (np. kolor przycisków).
- `onPrimaryColor`: kolor kontrastujący z przewodnim kolorem aplikacji w zapisie szesnastkowym.
- `textColor`: kolor tekstu w zapisie szesnastkowym.
- `backgroundColor`: kolor tła aplikacji w zapisie szesnastkowym.
- `errorColor`: kolor komunikatów o błędzie w zapisie szesnastkowym.
- `localization`: parametr ustawień narodowych dla natywnych elementów aplikacji mobilnej (ustawiany niezależnie od języka aplikacji nAxiom); obsługiwane wartości: *pl* oraz *en*