

# Trail Composer: instrukcja instalacji i wdrażania

## 1 ŚRODOWISKO ROZWOJOWE

---

### 1.1 INSTALOWANIE PAKIETÓW NUGET

Z poziomu root projektu wywołaj komendę dotnet restore

### 1.2 KONFIGURACJA BAZY DANYCH

Skrypt tworzący bazę danych znajduje się w

DBScripts/Trail\_composer\_entity\_diagram\_create Connection string do bazy danych należy wpisać w appsettings.json w przeznaczonym do tego miejscu.

### 1.3 KONFIGURACJA AUTORYZACJI UŻYTKOWNIKÓW

Aplikacja zakłada korzystanie z usługi Azure AD B2C. Aplikacja wymaga przepływów użytkownika (ang. user flows):

- rejestrowanie i logowanie ("SignUpSignInPolicyId")
- resetowanie hasła ("ResetPasswordPolicyId")
- edytowanie profilu (wykorzystywany atrybut to "Nazwa wyświetlana") ("EditProfilePolicyId")

Nazwy przepływów użytkownika używanych w Azure AD B2C oraz nazwa domeny ("Instance"), identyfikator subskrypcji ("ClientId") oraz nazwa domeny ("Domain") powinny zostać wprowadzone w odpowiednich miejscach w pliku appsettings.json Ponadto w pliku ClientApp/src/authConfig.js należy wprowadzić odpowiednie url do przepływów użytkowników, nazwę domeny, identyfikator subskrypcji oraz redirectUri dla opublikowanej aplikacji w przeznaczonych do tego miejscach. Jeżeli aplikacja nie jest opublikowana upewnij się, że w ClientApp/src/index.js jest wykorzystywana msalConfigLocal. Jeżeli zmienisz port dla serwera proxy, nie zapomnij zmienić msalConfigLocal.auth.redirectUri na nowe.

## 2 ŚRODOWISKO PRODUKCYJNE

---

Przed publikacją aplikacji upewnij się, że projekt jest prawidłowo skonfigurowany dla środowiska rozwojowego, tzn. aplikacja kliencka korzysta z poprawnie skonfigurowanego `msalConfigAzure` w pliku `index.js`. Publikacja odbywa się za pomocą Visual Studio 2022 (lub nowsze) na wybrany Azure AppService.