

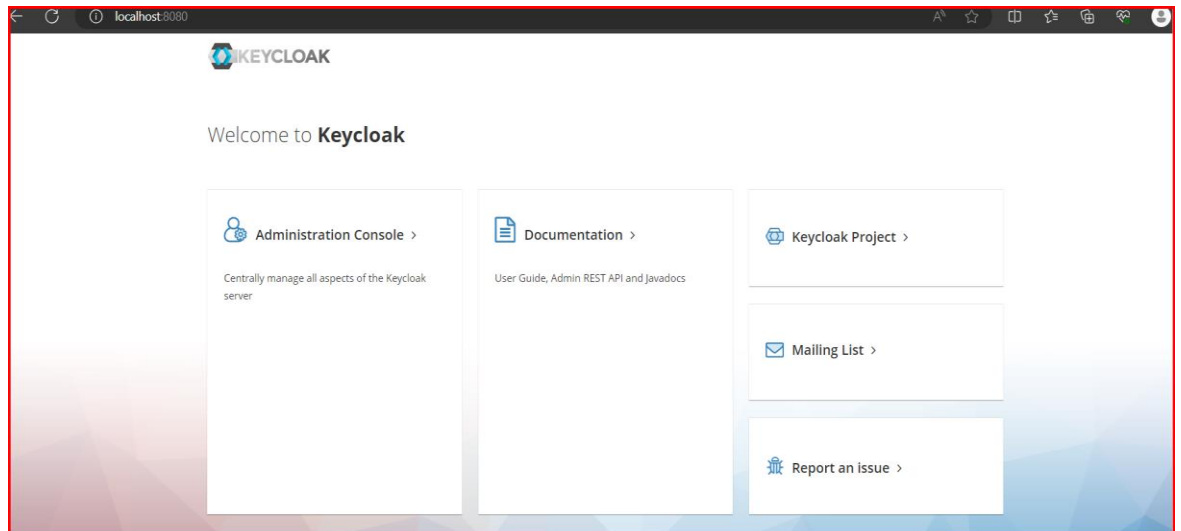
Sprawozdanie z zajęć nr 2Tomasz Bożek 83030

1. Keycloak

Rezultat wykonania polecenia instalacji dockera:

```
root@b572461870632:~# docker run -p 8080:8080 -e KEYCLOAK_ADMIN=admin -e KEYCLOAK_ADMIN_PASSWORD=admin quay.io/keycloak/keycloak:23.0.4 start-dev
Unable to find image 'quay.io/keycloak/keycloak:23.0.4' locally
23.0.4: Pulling from keycloak/keycloak
f72461870632: Pull complete
afd07d3d318a: Pull complete
b5ad681416a8: Pull complete
9b7befc5e951: Pull complete
Digest: sha256:cff31dc6fbb0ab0b66176b990e6b9e262fa74a501abb9a4bfa4a529cbc8a526a
Status: Downloaded newer image for quay.io/keycloak/keycloak:23.0.4
Updating the configuration and installing your custom providers, if any. Please wait.
2024-01-19 17:36:22,320 INFO [io.quarkus.deployment.QuarkusAugmentor] (main) Quarkus augmentation completed in 3792ms
2024-01-19 17:36:23,266 INFO [org.keycloak.quarkus.runtime.hostname.DefaultHostnameProvider] (main) Hostname settings:
Base URL: <unset>, Hostname: <request>, Strict HTTPS: false, Path: <request>, Strict BackChannel: false, Admin URL: <unset>, Admin: <request>, Port: -1, Proxied: false
2024-01-19 17:36:24,292 WARN [io.quarkus.agroal.runtime.DataSources] (main) Datasource <default> enables XA but transaction recovery is not enabled. Please enable transaction recovery by setting quarkus.transaction-manager.enable-recovery=true, otherwise data may be lost if the application is terminated abruptly
2024-01-19 17:36:24,636 WARN [org.infinispan.PERSISTENCE] (keycloak-cache-init) ISPN000554: jboss-marshalling is deprecated and planned for removal
2024-01-19 17:36:24,724 WARN [org.infinispan.CONFIG] (keycloak-cache-init) ISPN000569: Unable to persist Infinispan internal caches as no global state enabled
2024-01-19 17:36:24,811 INFO [org.infinispan.CONTAINER] (keycloak-cache-init) ISPN000556: Starting user marshaller 'org.infinispan.jboss.marshalling.core.JBossUserMarshaller'
2024-01-19 17:36:25,203 INFO [org.keycloak.connections.infinispan.DefaultInfinispanConnectionProviderFactory] (main) No
```

Keycloak jest widoczny i dostępny lokalnie na porcie 8080:



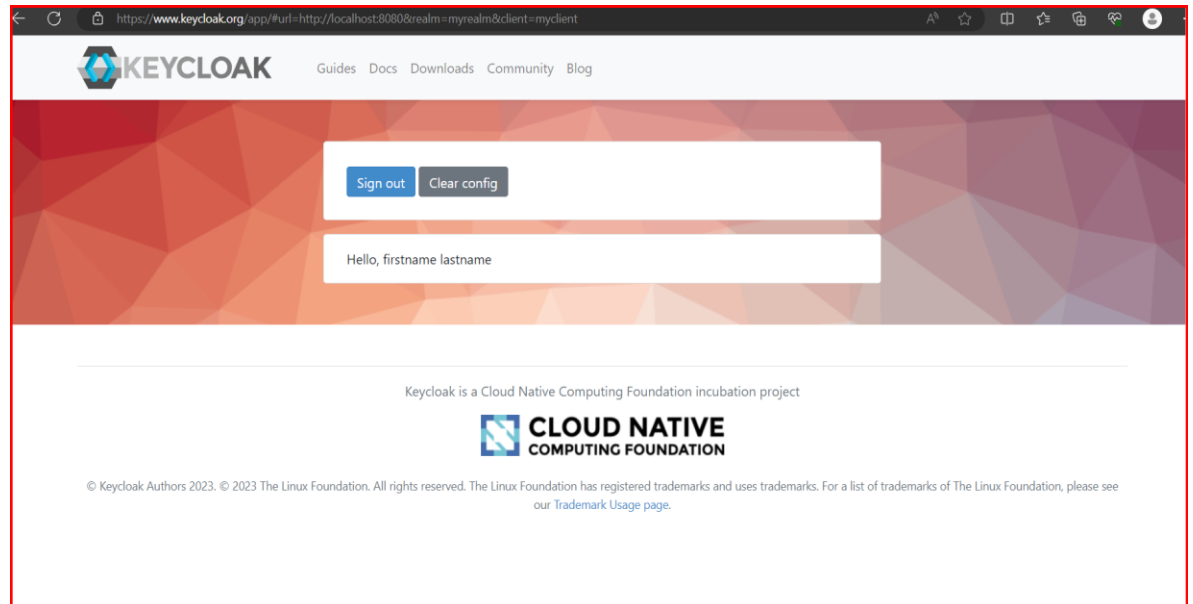
User „myuser” został utworzony poprawnie:

The screenshot shows the Keycloak user profile page for a user named 'myuser'. The page is titled 'Personal info' and contains a sidebar with 'Personal info', 'Account security', and 'Applications'. The main content area has a 'Personal info' section with a 'Manage your basic information.' header and a note that 'All fields are required.' The form includes input fields for 'Username' (filled with 'myuser'), 'Email', 'First name' (filled with 'firstname'), and 'Last name' (filled with 'lastname'). At the bottom are 'Save' and 'Cancel' buttons. The browser address bar shows 'localhost:8080/realms/myrealm/account/#/personal-info'.

Client „myclient” także został utworzony poprawnie:

The screenshot shows the Keycloak client configuration page for a client named 'myclient'. The page is titled 'Clients > Client details' and includes a sidebar with 'myrealm', 'Manage', 'Clients', 'Client scopes', 'Realm roles', 'Users', 'Groups', 'Sessions', 'Events', 'Configure', 'Realm settings', 'Authentication', 'Identity providers', and 'User federation'. The main content area has a 'myclient' section with an 'OpenID Connect' tab and a status of 'Enabled'. Below this are tabs for 'Settings', 'Roles', 'Client scopes', 'Sessions', and 'Advanced'. The 'Settings' tab is active, showing 'General settings' and 'Access settings'. The 'General settings' section includes input fields for 'Client ID' (filled with 'myclient'), 'Name', and 'Description', and a toggle for 'Always display in UI' (set to 'Off'). The 'Access settings' section includes a 'Root URL' input field. At the bottom are 'Save' and 'Revert' buttons. The browser address bar shows 'localhost:8080/admin/master/console/#/myrealm/clients/2cead71-2fc0-41eb-b8af-83c6f43dacdb/settings'.

Poprawne zalogowanie do „keycloak/app”:



2. Opisz czym jest uwierzytelnianie oparte na hasle oraz uwierzytelnianie wieloskladnikowe. Porownaj obie metody oraz podaj przyklady zastosowan
 - a. Uwierzytelnianie oparte na hasle oraz uwierzytelnianie wieloskladnikowe to dwie rózne metody zabezpieczania dostępu do systemów komputerowych i danych. Poniżej znajdziesz opisy obu metod, porównanie ich oraz przykłady zastosowań:
 - i. Uwierzytelnianie oparte na hasle:

To tradycyjna metoda, w której użytkownik potwierdza swoją tożsamość, wpisując odpowiednie hasło. Hasło jest tajnym ciągiem znaków, którym tylko uprawniona osoba powinna dysponować.
 - ii. Zalety:
 1. Prostota i łatwość implementacji.
 2. Niskie koszty wdrożenia.
 - iii. Wady:
 1. Narażone na ataki typu "brute force" (próby przetamania hasła poprzez wielokrotne próby).
 2. Zagrożenie w przypadku wykorzystania słabych haseł.
 - iv. Przykłady zastosowań:
 1. Konta użytkowników w systemach internetowych (np. poczta elektroniczna, portale społecznościowe).
 2. Logowanie do systemów operacyjnych.

b. Uwierzytelnianie wieloskładnikowe:

- i. To bardziej zaawansowana metoda, która wymaga od użytkownika potwierdzenia swojej tożsamości za pomocą dwóch lub więcej niezależnych czynników. Czynniki te mogą obejmować coś, co użytkownik zna (np. hasło), coś, co użytkownik posiada (np. klucz fizyczny) i coś, co użytkownik jest (np. biometria).
- ii. Zalety:
 1. Wyższy poziom bezpieczeństwa dzięki używaniu kilku czynników.
 2. Trudniejsze do przełamania nawet w przypadku utraty jednego z czynników.
- iii. Wady:
 1. Wyższy koszt wdrożenia i utrzymania.
 2. Potencjalne trudności dla użytkowników w korzystaniu z systemu.
- iv. Przykłady zastosowań:
 1. Potwierdzanie tożsamości w transakcjach finansowych (np. bankowość internetowa) za pomocą kodów SMS i hasła.
 2. Uwierzytelnianie dwuskładnikowe w usługach internetowych, które wymagają dodatkowego kodu generowanego na urządzeniu mobilnym.
 3. Systemy dostępu do danych o wysokim stopniu poufności, gdzie wymagane jest połączenie hasła i biometrii.

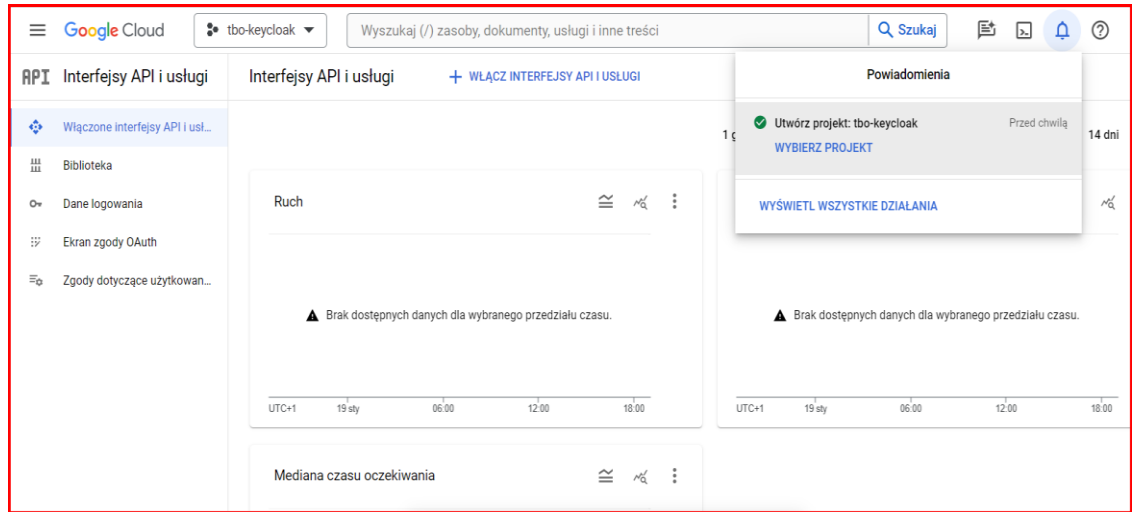
c. Podsumowanie:

Porównując obie metody, uwierzytelnianie wieloskładnikowe oferuje wyższy poziom bezpieczeństwa, ale także jest bardziej zaawansowane i kosztowne do wdrożenia. Wybór między tymi dwoma metodami zależy od poziomu bezpieczeństwa wymaganego przez konkretną aplikację czy system, a także od akceptowalnego poziomu wygody dla użytkowników. W wielu przypadkach, zwłaszcza tam, gdzie istnieje ryzyko utraty ważnych danych, korzystanie z uwierzytelniania wieloskładnikowego jest zalecane.

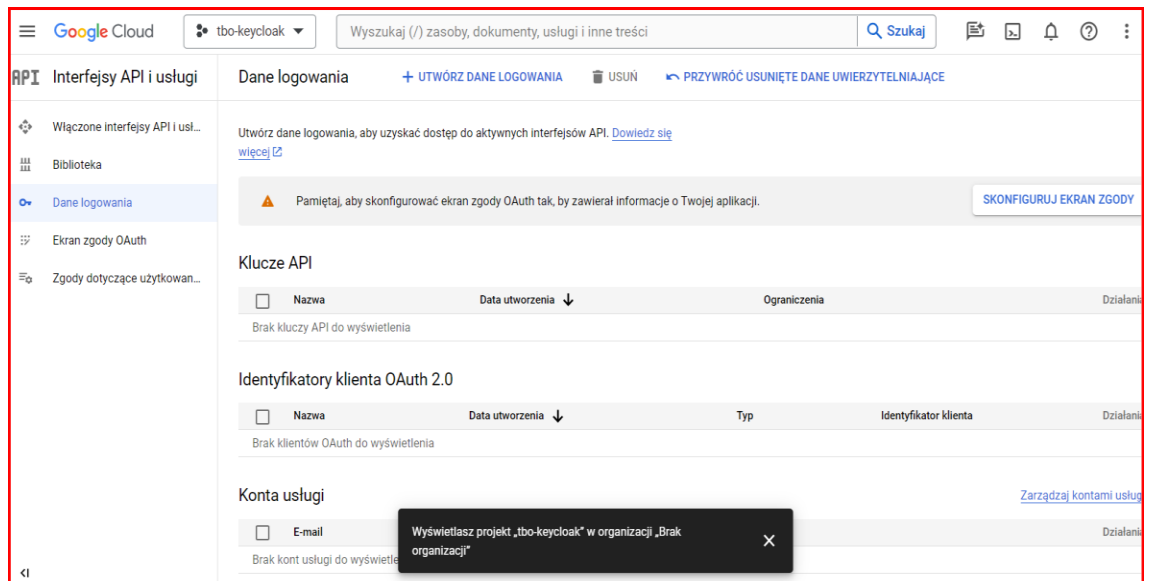
3. *Zadanie dodatkowe:

Spróbuj skonfigurować w Keycloaku identity providera np.. Google Identity Provider.

- a. Wybrałem google providera. Pierwszy krok to utworzenie projektu w google apis:



- b. Kolejny krok to przejście do ekranu konfigurowania zgód:



c. Kolejny krok to konfiguracja ekranu zgody:

Rozpocznij bezpłatny okres próbny ze środkami w wysokości 300 USD. Bez obaw – gdy środki się skończą, nie pobierzemy żadnych opłat. [Więcej informacji](#)

Google Cloud tbo-keycloak Wyszukaj (/) zasoby, dokumenty, usługi i inne treści Szukaj

API Interfejsy API i usługi Ekran zgody OAuth Samouczek

Włączone interfejsy API i us...
Biblioteka
Dane logowania
Ekran zgody OAuth
Zgody dotyczące użytkowan...

Wybierz sposób, w jaki chcesz skonfigurować i zarejestrować aplikację, w tym użytkowników docelowych. Możesz powiązać ze swoim projektem tylko jedną aplikację.

User Type

☐ Wewnętrzny ?
Dostęp tylko dla użytkowników z Twojej organizacji. Nie musisz przysyłać aplikacji do weryfikacji. [Więcej informacji o typie użytkownika](#)

☐ Zewnętrzny ?
Każdy użytkownik mający konto Google może uzyskać dostęp. Twoja aplikacja będzie uruchamiana w trybie testowym i będzie dostępna tylko dla użytkowników, których dodasz do listy testowych. Gdy Twoja aplikacja będzie gotowa do przeniesienia do środowiska produkcyjnego, może być konieczne przesłanie jej do weryfikacji. [Więcej informacji o typie użytkownika](#)

UTWÓRZ

Wyświetlasz projekt „tbo-keycloak” w organizacji „Brak organizacji”

Samouczek

Ekran zgody Google OAuth

Czym jest ekran zgody OAuth?

Czym są zakresy akceptacji OAuth?

Czym są wrażliwe zakresy interfejsu API?

Czym są zakresy interfejsu API z ograniczeniami?

Proces rejestracji aplikacji

Jakie informacje są mi potrzebne?

Czy Google musi zweryfikować moją aplikację?

Co się stanie, jeśli nie zgłoszę aplikacji do

d. Wybieramy zewnętrzny:

Google Cloud tbo-keycloak Wyszukaj (/) zasoby, dokumenty, usługi i inne treści Szukaj

API Interfejsy API i usługi Ekran zgody OAuth Samouczek

Włączone interfejsy API i us...
Biblioteka
Dane logowania
Ekran zgody OAuth
Zgody dotyczące użytkowan...

Wybierz sposób, w jaki chcesz skonfigurować i zarejestrować aplikację, w tym użytkowników docelowych. Możesz powiązać ze swoim projektem tylko jedną aplikację.

User Type

☐ Wewnętrzny ?
Dostęp tylko dla użytkowników z Twojej organizacji. Nie musisz przysyłać aplikacji do weryfikacji. [Więcej informacji o typie użytkownika](#)

☒ Zewnętrzny ?
Każdy użytkownik mający konto Google może uzyskać dostęp. Twoja aplikacja będzie uruchamiana w trybie testowym i będzie dostępna tylko dla użytkowników, których dodasz do listy testowych. Gdy Twoja aplikacja będzie gotowa do przeniesienia do środowiska produkcyjnego, może być konieczne przesłanie jej do weryfikacji. [Więcej informacji o typie użytkownika](#)

UTWÓRZ

Wyświetlasz projekt „tbo-keycloak” w organizacji „Brak organizacji”

Samouczek

Ekran zgody Google OAuth

Czym jest ekran zgod

Czym są zakresy ako

Czym są wrażliwe za

Czym są zakresy inte

Czym są zakresy inte

Proces rejestracji a

Jakie informacje są r

Czy Google musi zve

Czy Google musi zve

Co się stanie, jeśli ni

e. Konfigurujemy podstawowe ustawienia:

- Application type: Public
- Application name: Your application name (for this example I will use Keycloak Test App)
- Authorized domains: Your application's top-level domain name
- Application Homepage link: Your application homepage
- Application Privacy Policy link: Your application privacy policy link

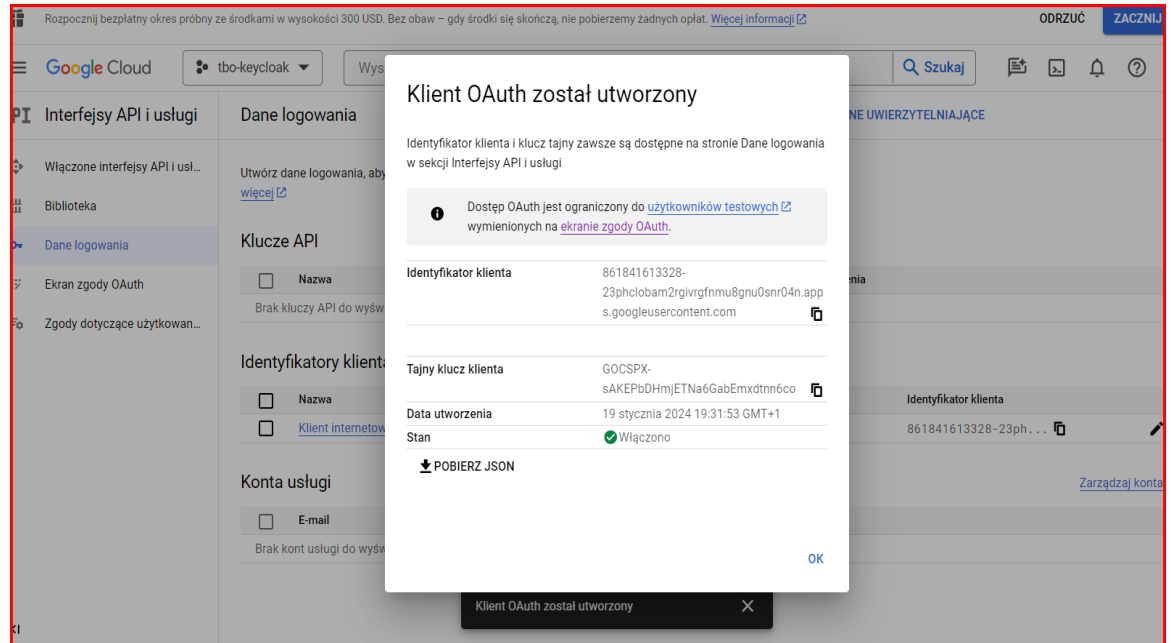
f. Następnie tworzy dane dostępowe klienta oauth:

The screenshot shows the Google Cloud IAM & Admin console. The left sidebar has 'Dane logowania' selected. The main content area shows the 'Dane logowania' page for a service account. A dropdown menu is open, showing options: 'Klucz interfejsu API', 'Identyfikator klienta OAuth', 'Konto usługi', and 'Pomóż mi wybrać'. The 'Identyfikator klienta OAuth' option is selected. The page shows sections for 'Klucze API', 'Identyfikatory klienta', and 'Konta usługi', all with 'Brak' (None) items displayed.

g. Dodajemy uri przekierowania które dostępne jest w keycloak:

The screenshot shows the Keycloak 'Identyfikator klienta - Aplikacja internetowa' (Web Application) page. The 'Autoryzowane źródła JavaScriptu' (Authorized JavaScript sources) section is visible. The 'Autoryzowane identyfikatory URI przekierowania' (Authorized redirect URIs) section has a text input field containing 'http://localhost:8080/realms/myrealm/broker/google/endpoint'. A red arrow points to the trash icon next to the input field. The 'Tajny klucz klienta' (Client secret) section shows a secret key and its expiration date. The 'ZAPISZ' (Save) button is visible at the bottom.

- h. Po poprawnym zapisaniu zarówno id klienta jak i sekretny klucz powinny zostać wygenerowane:



Należy je potem wpisać na ekranie konfiguracji providera google:

