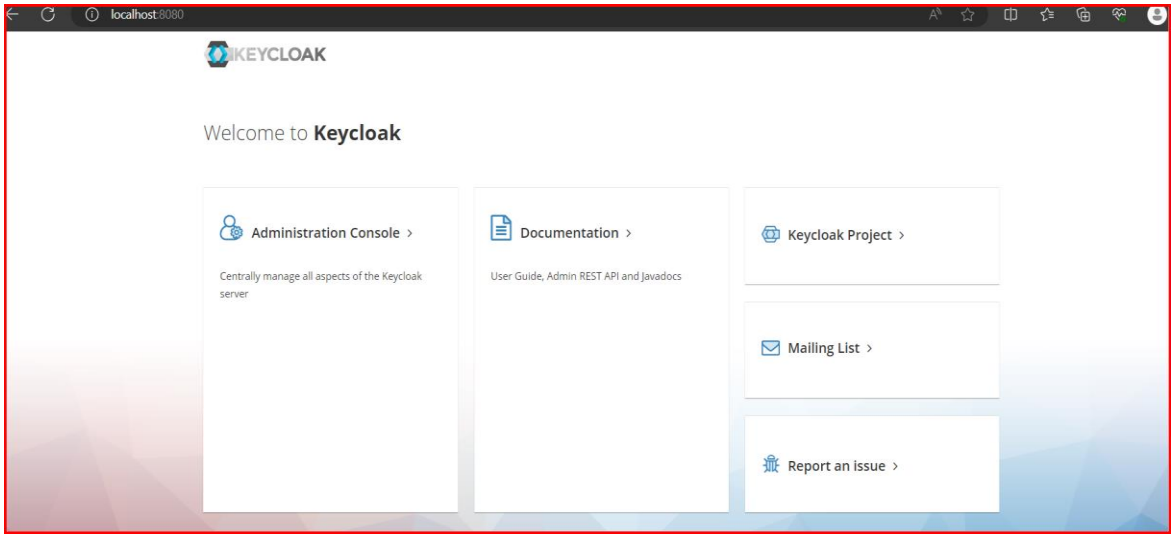
Sprawozdanie z zajęć nr 2Tomasz Bożek 83030

1. Keycloak

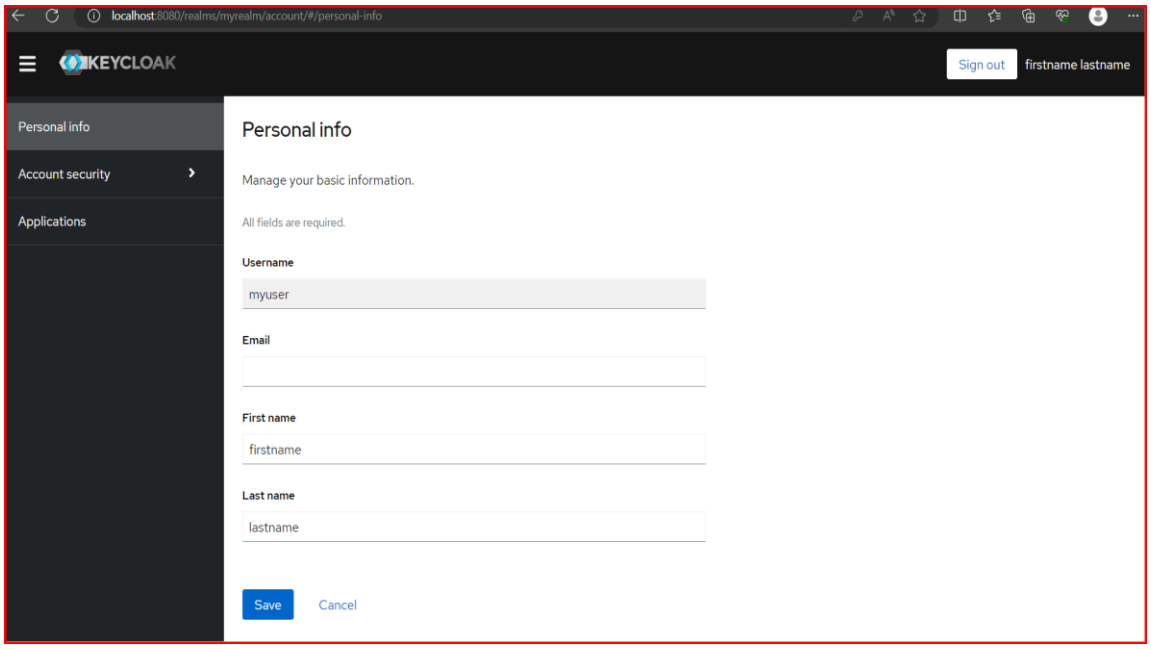
Rezultat wykonania polecania instalacji dockera:

```
[redacted]tid> 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: <uns
et>, 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 transac
tion 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 deprec
ated and planned for removal
2024-01-19 17:36:24,724 WARN  [org.infinispan.CONFIG] (keycloak-cache-init) ISPN000569: Unable to persist Infinispan int
ernal 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
```

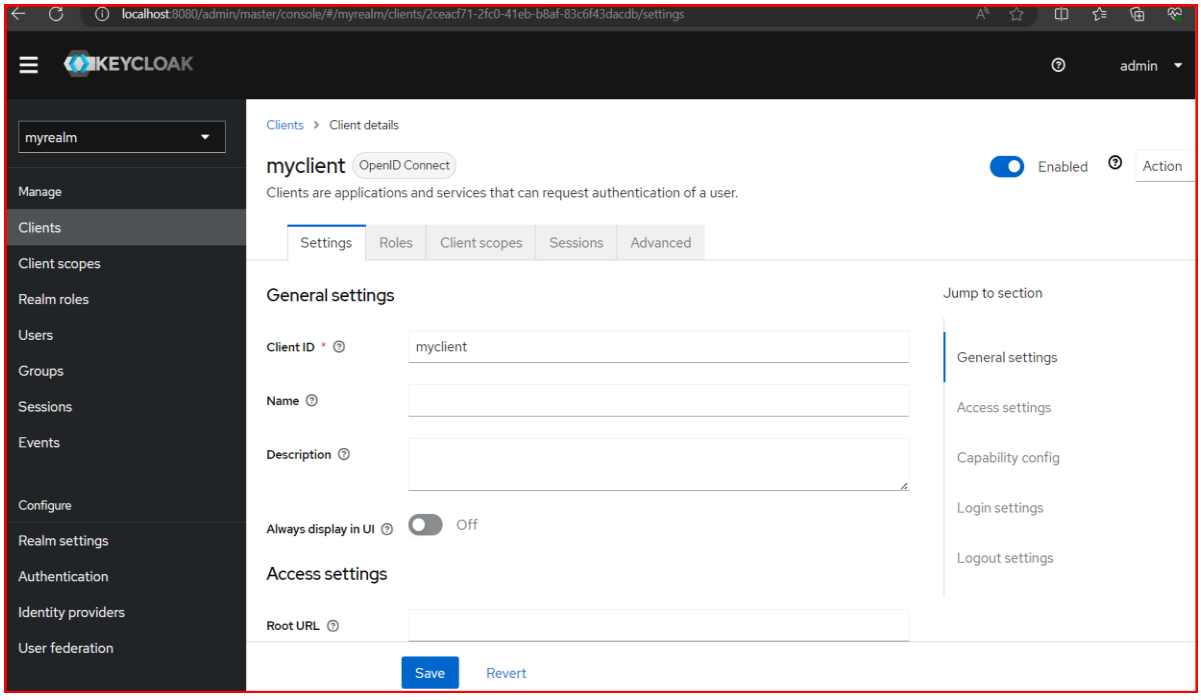
Keycloak jest widoczny i dostępny lokalnie na porcie 8080:



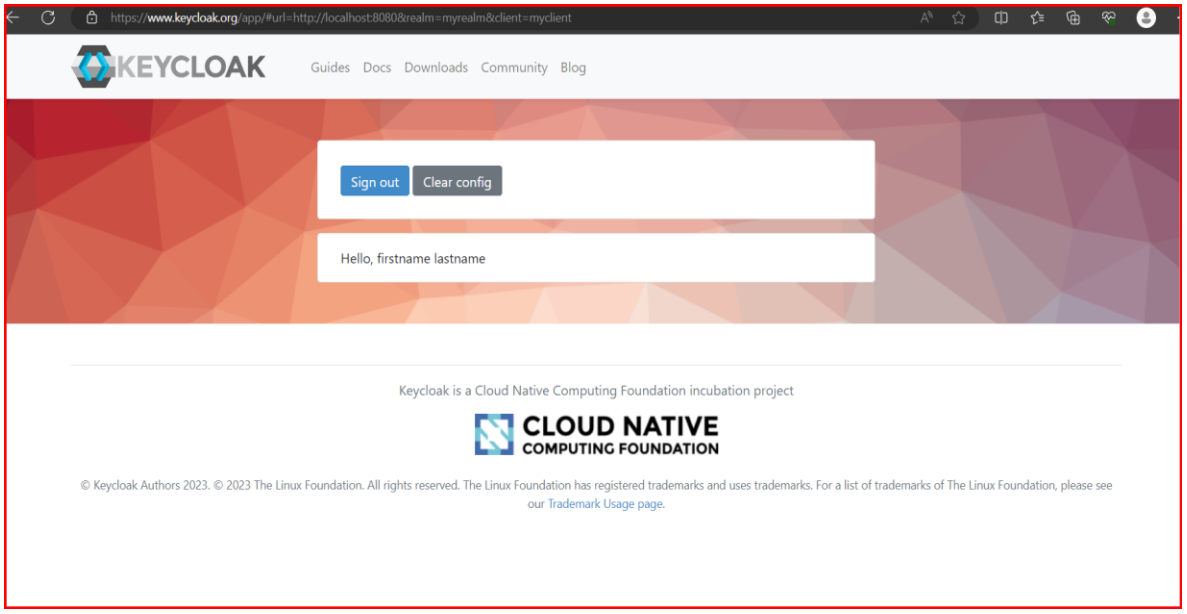
User „myuser” został utworzony poprawnie:



Client „myclient” także został utworzony poprawnie:



Poprawne zalogowanie do „keycloak/app”:



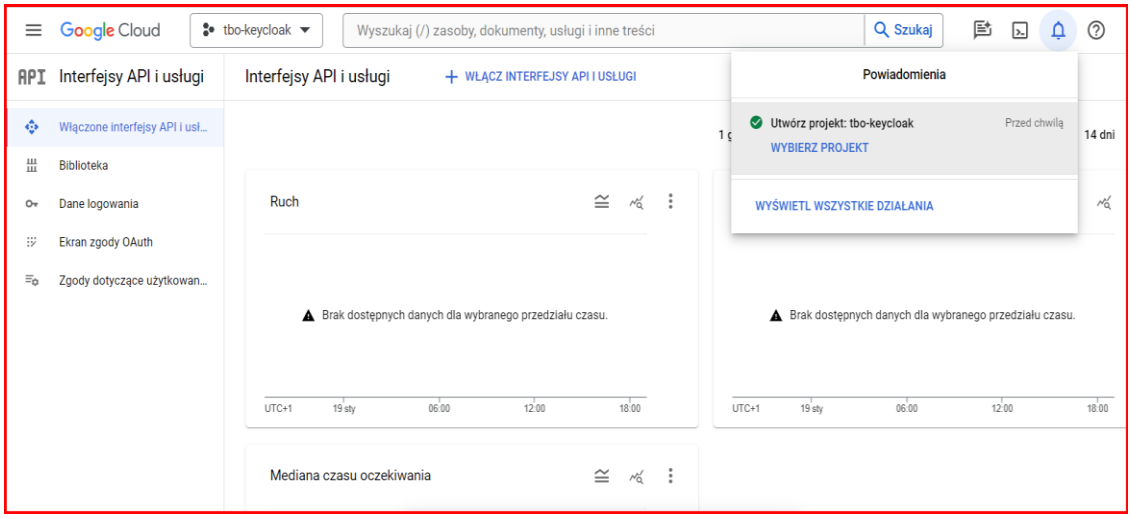
2. Opisz czym jest uwierzytelnianie oparte na haśle oraz uwierzytelnianie wieloskładnikowe. Porównaj obie metody oraz podaj przykłady zastosowań
    a. Uwierzytelnianie oparte na haśle oraz uwierzytelnianie wieloskładnikowe to dwie różne metody zabezpieczania dostępu do systemów komputerowych i danych. Poniżej znajdziesz opisy obu metod, porównanie ich oraz przykłady zastosowań:
        i. Uwierzytelnianie oparte na haśle:
        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 przełamania 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 haseł.
        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.

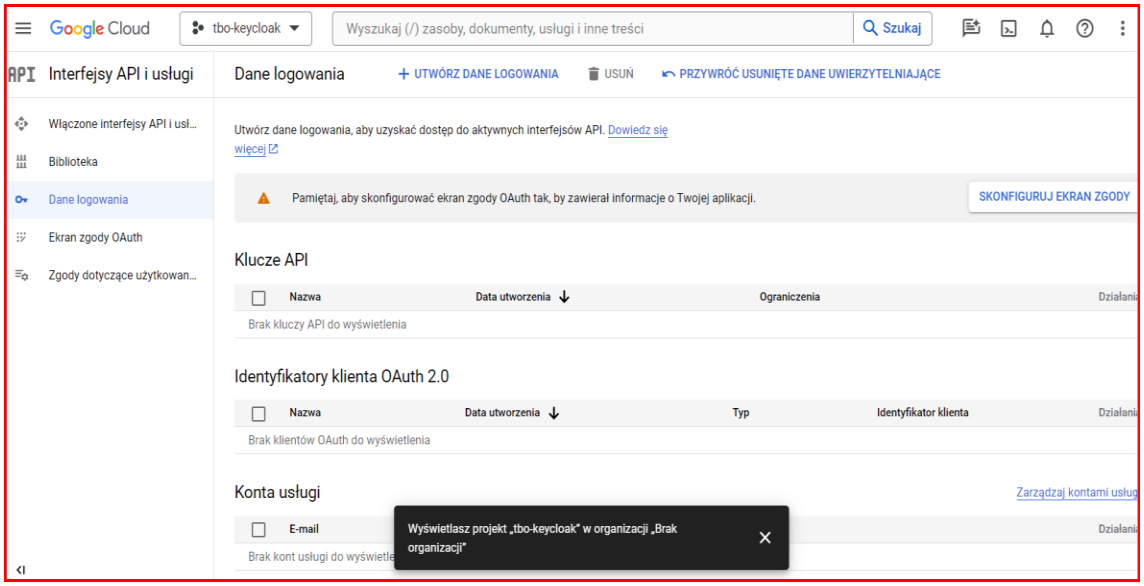
c. Podsumowanie:
    Porównując obie metody, uwierzytelnianie wieloskładnikowe oferuje wyższy poziom bezpieczeństwa, ale także jest bardziej zaawansowane i kosztowne do wdrożenia. Wybór między tymi dwoma metodami zależy od poziomu bezpieczeństwa wymaganego przez konkretną aplikację czy system, a także 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.

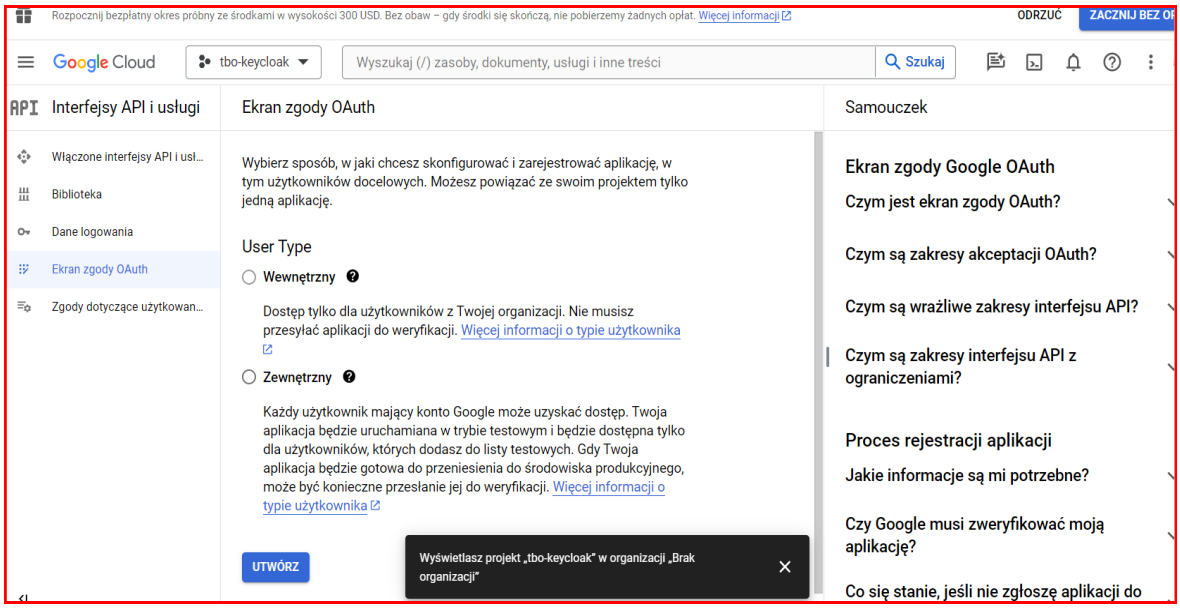
a. Wybrałem google providera. Pierwszy krok to utworzenie projektu w google apis:



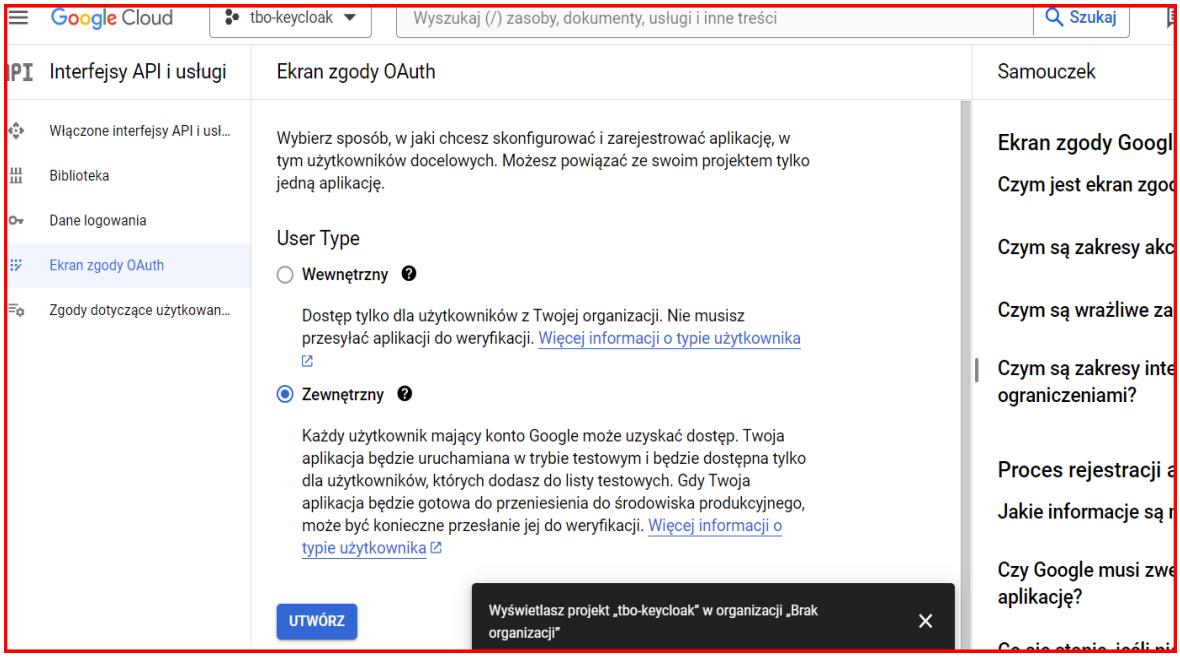
b. Kolejny krok to przejście do ekranu konfigurowania zgód:



c. Kolejny krok to konfiguracja ekranu zgody:

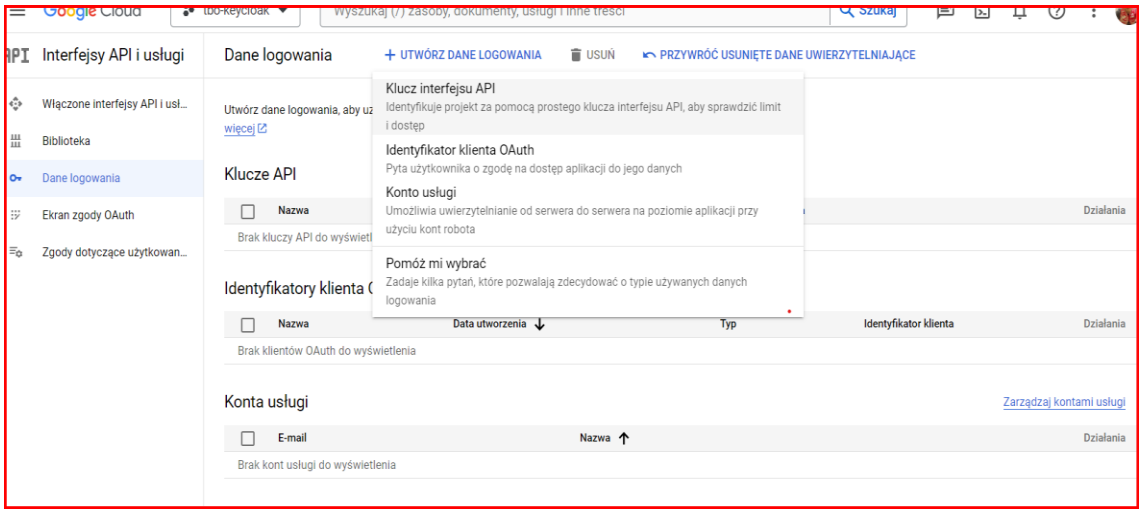


d. Wybieramy zewnętrzny:

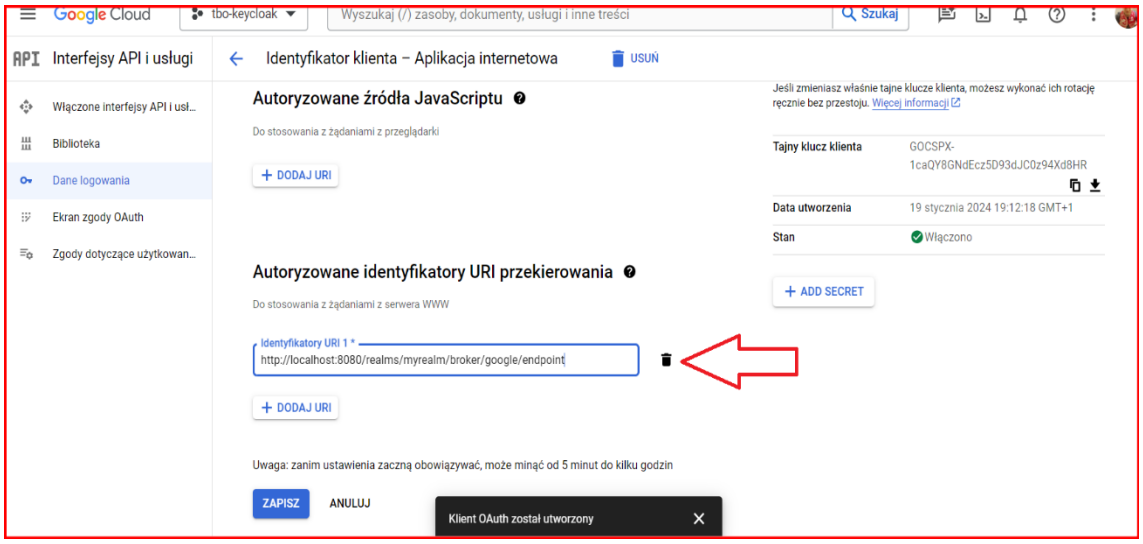


e. Konfigurujemy podstawowe ustawienia:
   i. Application type: Public
   ii. Application name: Your application name (for this example I will use Keycloak Test App)
   iii. Authorized domains: Your application's top-level domain name
   iv. Application Homepage link: Your application homepage
   v. Application Privacy Policy link: Your application privacy policy link

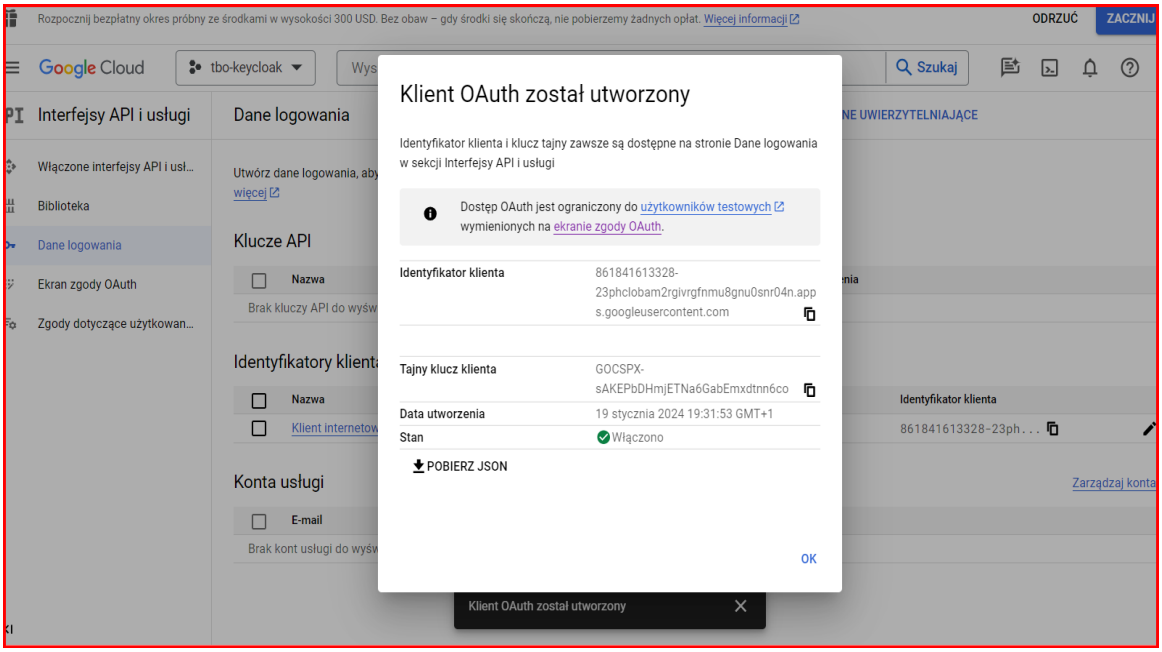
f. Następnie tworzy dane dostępowe clienta oauth:



g. Dodajemy uri przekierowania które dostępne jest w keycloak:



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



Należy je potem wpisać na ekranie konfiguracji providera google: