Załącznik 5

Instrukcja Korzystania z API Modułu Tożsamości

Wersja 1.8

Spis treści

1. Wstęp 3
2. Opis usługi uwierzytelnienia użytkownika 3
2.1. Zalogowanie użytkownika 3
3. Dostęp do usługi uwierzytelnienia na platformie eZamówienia 5
3.1. Konfiguracja SSL/TLS 5
4. Wersjonowanie usługi do uwierzytelnienia na platformie eZamówienia 6
5. Opis parametrów przekazywanych w procesie uwierzytelnienia 6
5.1. Usługa uwierzytelnienia 6
5.2. Usługa pobrania tokenu dostępowego 6
5.3. Usługa odczytu danych zalogowanego użytkownika 7
5.4. Usługa wylogowania 8
5.5. Usługa odczytu klucza publicznego do weryfikacji tokenu JWT 8
6. Przykładowa sekwencja żądań dla procesu uwierzytelnienia/autoryzacji: 9
7. Zawartość tokenu dostępowego 12

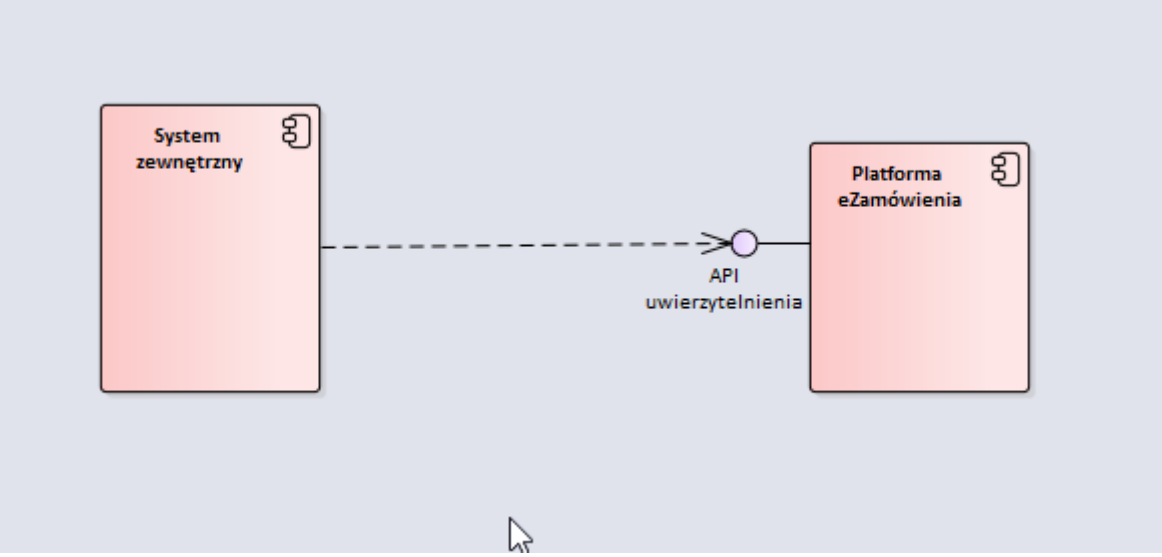
1. Wstęp

Niniejszy dokument opisuje integrację z usługą uwierzytelnienia użytkownika udostępnianą przez platformę eZamówienia dla systemów zewnętrznych.

2. Opis usługi uwierzytelnienia użytkownika

Usługa uwierzytelnienia użytkownika udostępniana przez platformę eZamówienia jest dostępna dla zarejestrowanych systemów zewnętrznych.

Usługa uwierzytelnia użytkowników zarejestrowanych w platformie eZamówienia.



Usługa uwierzytelnienia użytkownika jest zgodna ze standardem OAuth 2.0, scenariusz Authorization Code Grant. W odpowiedzi na uwierzytelnienie do systemu zewnętrznego przekazywany jest kod, który pozwala na pobranie tokenu dostępowego zgodnego z JWT.

2.1. Zalogowanie użytkownika

Niniejszy rozdział prezentuje diagram sekwencji wymagany do uwierzytelnienia użytkownika w platformie eZamówienia.

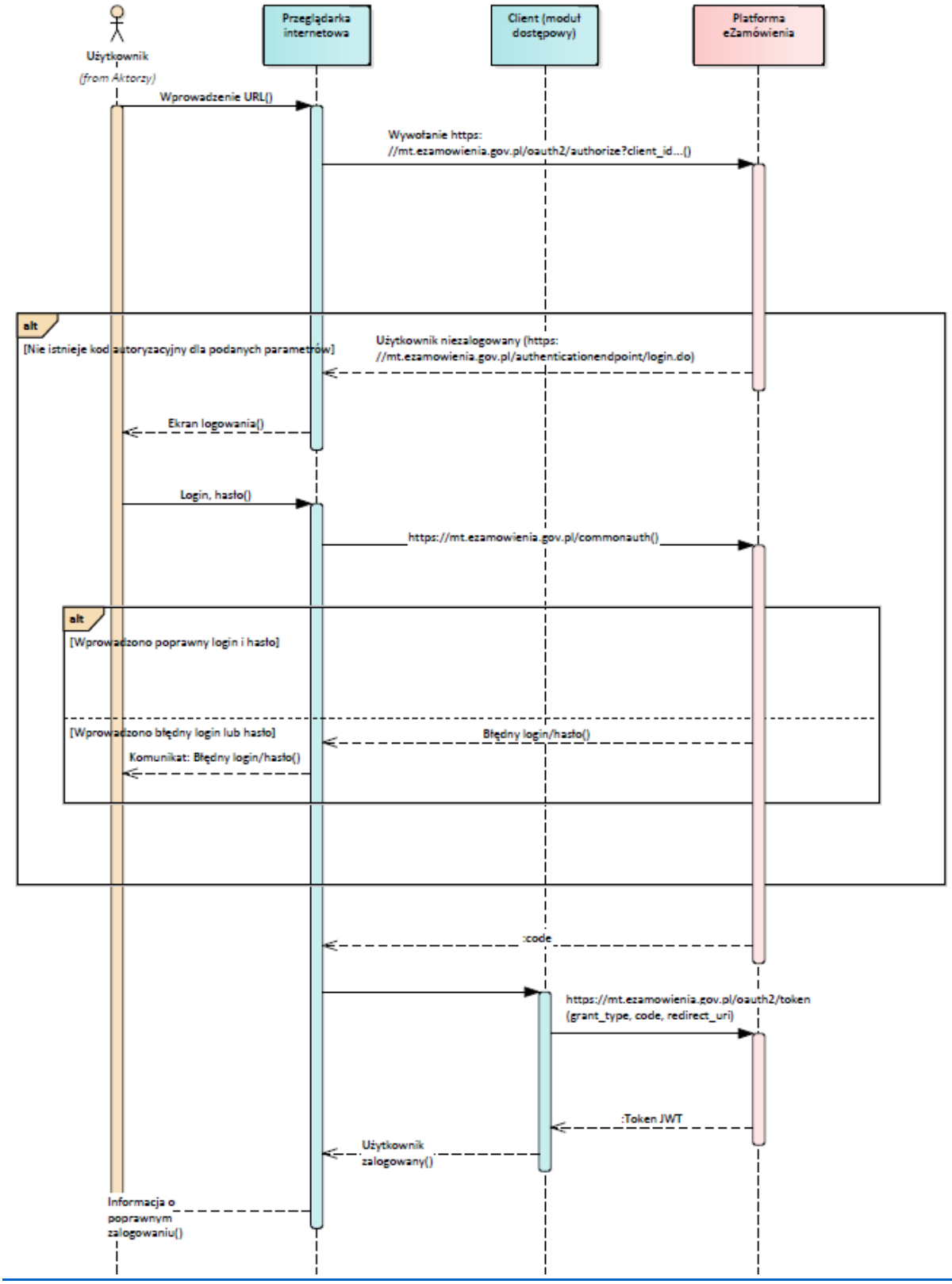
Opis procesu:

Użytkownik wysyła żądanie autoryzacji.

Jeżeli żądanie nie zawiera kodu autoryzacji, następuje przekierowanie na ekran logowania. Użytkownik wprowadza login i hasło, następuje uwierzytelnienie. Jeżeli użytkownik wprowadził nieprawidłowy login/hasło zostaje wyświetlony odpowiedni komunikat.

Po poprawnym uwierzytelnieniu do systemu zewnętrznego przekazywany jest kod autoryzacji. Na podstawie kodu autoryzacji zawartego w żądaniu serwer uwierzytelniający zwraca token dostępowy JWT.

Użytkownik jest poprawnie zalogowany do platformy eZamówienia, następuje przekierowanie do aplikacji klienckiej. Poniżej diagram procesu:



3. Dostęp do usługi uwierzytelnienia na platformie eZamówienia

Dostęp do API w celu uwierzytelnienia wymaga podania parametrów dostępowych. Aby je uzyskać należy uprzednio zarejestrować system zewnętrzny, w odpowiedzi Administrator przekaże potrzebne parametry konfiguracyjne. Szczegóły rejestracji systemu zewnętrznego zostały opisane w dokumencie „Procedura uzyskania dostępu do platformy eZamówienia" dostępnym na Portalu Deweloperskim.

Dane do przekazania

- Nazwa systemu zewnętrznego
- Krótki opis systemu zewnętrznego
- Adres lub adresy systemu zewnętrznego, na który zostanie przekierowany użytkownik po zalogowaniu na platformie eZamówienia

W odpowiedzi administrator systemu eZamówienia przekaże następujące parametry

- client-id – identyfikator aplikacji klienckiej przypisanej do systemu zewnętrznego
- client-secret – hasło aplikacji klienckiej – używane jeśli nie korzystamy z PKCE
- access-token-uri - adres do pobrania tokenu z systemu eZamówienia
- user-authorization-uri - adres do uwierzytelnienia użytkownika w systemie eZamówienia
- user-info-uri - adres do pobrania informacji o zalogowanym użytkowniku
- checksession-uri - adres do sprawdzenia zmian w sesji w systemie eZamówienia
- logout-uri - adres do usługi wylogowania w systemie eZamówienia
- scope – zakres dostepu aplikacji klienckiej, scope wykorzystywane do budowania tokenu to: openid,profile

Przykładowa konfiguracja aplikacji zbudowanej na spring-boot:

```
epzp.base.address=https://ezamowienia.gov.pl
security.oauth2.client.client-id=client-id
security.oauth2.client.client-secret=client-secret
security.oauth2.client.access-token-uri=${epzp.base.address}/oauth2/token
security.oauth2.client.user-authorization-uri=${epzp.base.address}/oauth2/authorize
security.oauth2.resource.user-info-uri=${epzp.base.address}/oauth2/userinfo
security.oauth2.client.pre-established-redirect-uri=http://client.ezamowienia.gov.pl/login
security.oauth2.client.use-current-uri=false
security.oauth2.client.checksession-uri=${epzp.base.address}/oidc/checksession
security.oauth2.client.logout-uri=${epzp.base.address}/oidc/logout
security.oauth2.client.scope=openid,profile
```

3.1. Konfiguracja SSL/TLS

Usługi wykorzystywane w procesie uwierzytelnienia użytkownika w platformie eZamówienia używają połączenia szyfrowanego. W związku z tym systemy zewnętrzne powinny

akceptować certyfikaty i klucze platformy eZamówienia wykorzystywane do połączenia szyfrowanego. Klucz i certyfikat publiczny eZamówienia zostanie przekazany przez administratora platformy eZamówienia.

4. Wersjonowanie usługi do uwierzytelnienia na platformie eZamówienia

Usługi realizujące proces uwierzytelnienia nie są wersjonowane.

5. Opis parametrów przekazywanych w procesie uwierzytelnienia

5.1. Usługa uwierzytelnienia

Adres usługi: https://ezamowienia.gov.pl/oauth2/authorize
Parametry żądania GET:

- client_id – identyfikator modułu, z którego przychodzi żądanie o token dostępowy
- redirect_uri – adres aplikacji, na który zostanie przekazana odpowiedź z uwierzytelnienia
- response_type=code – metoda uwierzytelnienia wskazująca na wykorzystanie „OAuth 2.0 Authorization Code Grant"
- scope=openid,profile - zakres informacji o jaki moduł prosi mt. profile - dane użytkownika, openid - jest to rozszerzony zakres informacji zgodny ze specyfikacją oauth2/oidc zawierający m.in. role użytkownika. Dane o które wystąpi moduł będą dołączone do ID tokenu (id_token) i zwracane przez usługę odczytu danych zalogowanego użytkownika.
- state - Wartość do dodatkowego zabezpieczenia komunikacji pomiędzy żądaniem klienta a wywołaniem zwrotnym serwera autoryzacji (Aby zapobiec atakom).
- code_challenge - zabezpieczanie uprawnień kodu autoryzacji za pośrednictwem klucza weryfikacji Exchange (w przypadku używania PKCE).
- code_challenge_method - metoda kodowania code_verifier code_challenge parametru np. S256 (w przypadku używania PKCE).
- nonce - ciąg znaków pozwalający na identyfikowanie pochodzenia żądania (w przypadku używania PKCE).

Odpowiedź zawiera HTTP Status = 302 i nagłówek Location wskazujący na stronę uwierzytelnienia np.
Location: https://ezamowienia.gov.pl/authenticationendpoint/login.do?(...)

5.2. Usługa pobrania tokenu dostępowego

Adres usługi: https://ezamowienia.gov.pl/oauth2/token
Dane przekazywane w żądaniu POST:

- code – kod autoryzacji otrzymany w odpowiedzi na uwierzytelnienie, otrzymany w odpowiedzi na żądanie z punktu 5.1.
- redirect_uri – adres aplikacji, na który zostanie przekazana odpowiedź z uwierzytelnienia
- grant_type - Wartość "authorization_code". Typ poświadczenia, który zwraca serwer autoryzacji.
- code_verifier - Wymagane, jeśli w żądaniu udzielenia kodu autoryzacji był używany kod PKCE.
- client_id – identyfikator systemu, z którego przychodzi żądanie o token dostępowy

W żądaniu należy dodać nagłówek „Content-Type" z wartością „application/x-www-form-urlencoded".

Jeśli system zewnętrzny nie używa PKCE to w nagłówkach: Authorization o wartości "Basic SECRET" (zgodnie z OAuth2), gdzie:

- SECRET to zakodowane w Base64 CLIENT_ID oraz CLIENT_SECRET (rozdzielone dwukropkiem)
- CLIENT_ID to identyfikator modułu, z którego przychodzi żądanie o token dostępowy
- CLIENT_SECRET to "sekret"modułu, z którego przychodzi żądanie o token dostępowy
- Przykład poprawnej wartości nagłówka Authorization: *Basic czZCaGRSa3F0MzpnWDFmQmF0M2JW*

Odpowiedź zawiera HTTP status = 200 i treść odpowiedzi z tokenem dostępowym (**access_token**) np.


```
{
   "access_token": "eyNXQiOiJNell4TW1Ga09HWXdNV0kwWldObU5EY3hOR1l3WW1NNFpUQ(…)",
   "scope": "openid profile",
   "id_token": "eyJ4NXQiOiJNell4TW1Ga09HWXdNV0kwWldObU5EY3hOR1l3WW1NNFpV0(…)",
   "refresh_token": "7f3b9dc1-ff70-38b3-ab38-f21cd4508131",
   "token_type": "Bearer",
   "expires_in": 3600
}
```


W odpowiedzi zwracany jest także

- Token dostępowy (access_token)
- ID Token (id_token) , który zawiera dane zalogowanego użytkownika.
- Token służący do przedłużenia ważności tokenu dostępowego (refresh_token)

Rekomendowane jest wykorzystanie PKCE (Proof Key for Code Exchange) w procesie pozyskania tokenu dostępowego.

5.3. Usługa odczytu danych zalogowanego użytkownika

Adres usługi: https://ezamowienia.gov.pl/oauth2/userinfo

Do żądania dodajemy nagłówek HTTP: Authorization: Bearer <ACCESS_TOKEN> – ACCESS_TOKEN to token dostępowy otrzymany w odpowiedzi na żądanie z punktu 5.2.

Odpowiedź zawiera http status = 200 i treść odpowiedzi z danymi właściciela tokenu dostępowego np.


```
{
```

```
  "sub": "aplikacja",
  "userid": "b5b9b376-11dd-4a99-9996-a14608ec5022",
  "username": "aplikacja",
  "given_name": "Jan",
  "family_name": "Kowalski",
  "email": "jan.kowalski@example.com",
  "phone_number": "123456789",
  "iss": "https://ezamowienia.gov.pl/oauth2/token",
  "aud": "cvpjEBo8EsIolx9y4IwEtMm_q1Ea",
  "azp": "cvpjEBo8EsIolx9y4IwEtMm_q1Ea",
  "nbf": 1790947838,
  "exp": 1790951438,
  "iat": 1790947838,
  "jti": "f150ed0c-3504-4f40-ad4f-d81bafb9d26d",
  "multiple_organizations": "true",
  "organization": "aplikacja",
  "organization_id": "6047",
  "org_id": "10084a8d-113f-4211-a0d5-efe36b082211",
  "org_name": "Super",
  "org_handle": "carbon.super",
  "organization_role": [
    "BUYER",
    "SUPPLIER"
  ],
  "entitlements": [
    "MO_PPREAD",
    "SOZ_USER",
    "..."
  ],
  "groups": [
    "USER",
    "API_CALL",
    "..."
  ]
}
```


5.4. Usługa wylogowania

Adres usługi: https://ezamowienia.gov.pl/oidc/logout/

W żądaniu GET należy przekazać:

- identyfikator otrzymanego tokenu JWT (id_token_hint),
- adres strony, na którą nastąpi przekierowanie po wylogowaniu (post_logout_redirect_uri).

Odpowiedź zawiera HTTP status = 302 i nagłówek Location wskazujący na stronę wylogowania z Platformy np.

Location: https://localhost:9443/authenticationendpoint/oauth2_logout_consent.do?(...)

5.5. Usługa odczytu klucza publicznego do weryfikacji tokenu JWT

Adres usługi: https://ezamowienia.gov.pl/oauth2/jwks/

W odpowiedzi na żądanie zostanie zwrócony klucz publiczny wymagany do weryfikacji sygnatury tokenu JWT. Należy zadbać o to, by aplikacja zewnętrzna wykonywała weryfikację sygnatury otrzymanego tokenu JWT.

Odpowiedź zawiera HTTP status = 200 i w treści klucz publiczny do weryfikacji sygnatury tokenu JWT np.


```
{
```

```
    "keys": [
        {
            "kty": "RSA",
            "e": "AQAB",
            "use": "sig",
            "kid": "NDljMzc3ZD(…)",
            "alg": "RS256",
            "n": "s5N_PEAdgNnlr3NadgbjObz6aLVoPIlhEDzbYc_a9vE91JOl4AyanlbCzb"
        }
    ]
}
```


6. Przykładowa sekwencja żądań dla procesu uwierzytelnienia/autoryzacji:

Na potrzeby uwierzytelnienia i autoryzacji Platforma eZamówienia wystawia następujący endpoint który należy podać w konfiguracji aplikacji klienckiej:

`/oauth2/authorize`

Poniżej rzeczywista przykładowa sekwencja żądań (specyficzna dla danej implementacji oraz konfiguracji):

1. https://ezamowienia.gov.pl**/oauth2/authorize**?client_id=cvpjEBo8EsIolx9y4IwEtMm_q1Ea&redirect_uri=http://localhost:8083/login&response_type=code&scope=openid%20profile&state=VKog1S
Parametry:
client_id: cvpjEBo8EsIolx9y4IwEtMm_q1Ea
redirect_uri: http://localhost:8083/login
response_type: code
scope: openid profile
state: VKog1S

Status Code: 302

2. https://ezamowienia.gov.pl/authenticationendpoint/login.do?client_id=cvpjEBo8EsIolx9y4IwEtMm_q1Ea&commonAuthCallerPath=%2Foauth2%2Fauthorize&forceAuth=false&passiveAuth=false&redirect_uri=http%3A%2F%2Flocalhost%3A8083%2Flogin&response_type=code&scope=openid+profile&state=VKog1S&tenantDomain=carbon.super&sessionDataKey=58d6ea0e-b447-4509-b060-3e222c957863&relyingParty=cvpjEBo8EsIolx9y4IwEtMm_q1Ea&type=oidc&sp=AplikacjaTest&isSaaSApp=false&authenticators=BasicAuthenticator%3ALOCAL
Parametry:
client_id: cvpjEBo8EsIolx9y4IwEtMm_q1Ea
commonAuthCallerPath: /oauth2/authorize
forceAuth: false
passiveAuth: false
redirect_uri: http://localhost:8083/login
response_type: code

scope: openid profile
state: VKog1S
tenantDomain: carbon.super
sessionDataKey: 58d6ea0e-b447-4509-b060-3e222c957863
relyingParty: cvpjEBo8EsIolx9y4IwEtMm_q1Ea
type: oidc
sp: AplikacjaTest
isSaaSApp: false
authenticators: BasicAuthenticator:LOCAL

Status Code: 200

3. https://ezamowienia.gov.pl/logincontext?sessionDataKey=58d6ea0e-b447-4509-b060-3e222c957863&relyingParty=cvpjEBo8EsIolx9y4IwEtMm_q1Ea&tenantDomain=carbon.super&_=1588942634045
Parametry:
sessionDataKey: 58d6ea0e-b447-4509-b060-3e222c957863
relyingParty: cvpjEBo8EsIolx9y4IwEtMm_q1Ea
tenantDomain: carbon.super
_: 1588942634045

Status Code: 200

4. https://ezamowienia.gov.pl/logincontext?sessionDataKey=58d6ea0e-b447-4509-b060-3e222c957863&relyingParty=cvpjEBo8EsIolx9y4IwEtMm_q1Ea&tenantDomain=carbon.super&_=1588942634046
Parametry:
sessionDataKey: 58d6ea0e-b447-4509-b060-3e222c957863
relyingParty: cvpjEBo8EsIolx9y4IwEtMm_q1Ea
tenantDomain: carbon.super
_: 1588942634046

Status Code: 200

5. https://ezamowienia.gov.pl/commonauth
Parametry:
usernameUserInput: test1
username: test1@carbon.super
password: test1
sessionDataKey: 58d6ea0e-b447-4509-b060-3e222c957863

Status Code: 302

6. https://ezamowienia.gov.pl**/oauth2/authorize**?sessionDataKey=13423803-c1ff-4ef8-92a5-1e9c51003fae
Parametry:
sessionDataKey: 13423803-c1ff-4ef8-92a5-1e9c51003fae

Status Code: 302

7. https://ezamowienia.gov.pl/authenticationendpoint/oauth2_consent.do?application=AplikacjaTest&tenantDomain=carbon.super&scope=openid+profile&sessionDataKeyConsent=6b670699-19cd-40a0-b158-9accb6800253&spQueryParams=client_id%3DcvpjEBo8EsIolx9y4IwEtMm_q1Ea%26redirect_uri%3Dhttp%3A%2F%2Flocalhost%3A8083%2Flogin%26response_type%3Dcode%26scope%3Dopenid%2520profile%26state%3DVKog1S
Parametry:
application: AplikacjaTest
tenantDomain: carbon.super
scope: openid profile
sessionDataKeyConsent: 6b670699-19cd-40a0-b158-9accb6800253
spQueryParams:
client_id=cvpjEBo8EsIolx9y4IwEtMm_q1Ea&redirect_uri=http://localhost:8083/login&response_type=code&scope=openid%20profile&state=VKog1S

Status Code: 200

8. https://ezamowienia.gov.pl**/oauth2/authorize**
Parametry:
scope-approval: approve
sessionDataKeyConsent: 6b670699-19cd-40a0-b158-9accb6800253
consent: approve

Status Code: 302

9. http://localhost:8083/login?code=1eacea07-decf-31e7-85d7-d33909d9ae76&state=VKog1S&session_state=5cbfb16acb6be7e6db7aa9c3ed409cc15ee70f9135fa60676dba013a33008694.YsLguvaEPvf-kzVg1ZTT_g
Parametry:
code: 1eacea07-decf-31e7-85d7-d33909d9ae76
state: VKog1S
session_state:
5cbfb16acb6be7e6db7aa9c3ed409cc15ee70f9135fa60676dba013a33008694.YsLguvaEPvf-kzVg1ZTT_g

POZYSKANIE TOKENU

W 9 punkcie do aplikacji klienckiej wysyłany jest parametr code - jest to authorization_code nadany przez serwer.
Następnie aplikacja kliencka używa tego authorization_code aby wysłać go do serwera i uzyskać token JWT. Szczegóły tej komunikacji zostały opisane w punkcie 5.2.
Endpoint WSO2 IS na potrzeby pozyskania tokenu który należy podać w konfiguracji aplikacji klienckiej:

`/oauth2/token`

W Spring Security pozyskanie tokenu jest realizowane w OAuth2ClientAuthenticationProcessingFilter za pomocą OAuth2RestTemplate.

7. Zawartość tokenu dostępowego

Token dostępowy wydany przez platformę eZamówienia jest zgodny ze specyfikacją JWT i w sekcji payload zawiera następujące dane:

- sub - login użytkownika
- aut - typ autentykacji / kontekst uwierzytelniania (np. APPLICATION_USER)
- iss - adres, z którego pochodzi token
- client_id - identyfikator aplikacji klienckiej
- aud - odbiorca tokenu (identyfikator aplikacji klienckiej)
- nbf - początek ważności tokenu (timestamp)
- azp - identyfikator autoryzowanej strony / aplikacji klienckiej
- org_id - identyfikator techniczny organizacji (podmiotu)
- scope - zakres dostępu
- exp - koniec ważności tokenu (timestamp)
- org_name - nazwa organizacji
- iat - czas utworzenia tokenu (timestamp)
- jti - unikalny identyfikator tokenu
- org_handle - uchwyt/identyfikator domeny organizacji (np. carbon.super)

Token dostępowy nie zawiera identyfikatora technicznego użytkownika (userid), danych osobowych ani uprawnień (entitlements). Dane te są dostępne w ID tokenie oraz w usłudze odczytu danych zalogowanego użytkownika (punkt 5.3).

Przykładowy token JWT wygenerowany przez serwer uwierzytelnienia platformy eZamówienia:


```
{
 "sub": "aplikacja",
 "aut": "APPLICATION_USER",
 "iss": "https://ezamowienia.gov.pl/oauth2/token",
 "client_id": "cvpjEBo8EsIolx9y4IwEtMm_q1Ea",
 "aud": "cvpjEBo8EsIolx9y4IwEtMm_q1Ea",
 "nbf": 1790947838,
 "azp": "cvpjEBo8EsIolx9y4IwEtMm_q1Ea",
 "org_id": "10084a8d-113f-4211-a0d5-efe36b082211",
 "scope": "openid profile",
```

```
 "exp": 1790951438,
 "org_name": "Super",
 "iat": 1790947838,
 "jti": "8b9ff201-992d-486f-a8ca-da8c2d8d0930",
 "org_handle": "carbon.super"
}
```


ID Token wydany przez platformę eZamówienia jest zgodny ze specyfikacją JWT i w sekcji payload zawiera następujące dane:

- sub - login użytkownika
- iss - adres, z którego pochodzi token
- userid - identyfikator techniczny użytkownika
- aud - odbiorca tokenu (identyfikator aplikacji klienckiej)
- nbf - początek ważności tokenu (timestamp)
- azp - identyfikator autoryzowanej strony / aplikacji klienckiej
- exp - koniec ważności tokenu (timestamp)
- iat - czas utworzenia tokenu (timestamp)
- jti - unikalny identyfikator tokenu
- isk - klucz sesji / identyfikator sesji w systemie uwierzytelniania
- sid - identyfikator sesji użytkownika
- at_hash - skrót tokenu dostępowego (Access Token Hash)
- c_hash - skrót kodu autoryzacyjnego (Code Hash)
- amr - metody uwierzytelniania zastosowane przy logowaniu
- given_name - imię użytkownika
- family_name - nazwisko użytkownika
- username - nazwa użytkownika
- email - adres email użytkownika
- phone_number - numer telefonu użytkownika
- organization - nazwa podmiotu wskazanego w procesie uwierzytelniania użytkownika
- organization_id - identyfikator podmiotu
- org_id - techniczny identyfikator organizacji
- org_name - nazwa organizacji
- org_handle - uchwyt/identyfikator domeny organizacji
- organization_role - role posiadane w organizacji (np. BUYER, SUPPLIER)
- entitlements - uprawnienia użytkownika (tablica wartości)
- groups - role/grupy użytkownika (tablica wartości)
- multiple_organizations - flaga wielopodmiotowości (true / false)

Przykładowy ID token JWT wygenerowany przez serwer uwierzytelnienia platformy eZamówienia:


```
{
  "isk":
"89213bed97262e78aab30de30e4cd0c72619a46ed792bdca19aaac8e7a423f09",
  "at_hash": "263VW4M2rkkDtd_ww7znEA",
  "entitlements": [
    "MO_PPREAD",
    "SOZ_USER",
    "..."
  ],
  "sub": "aplikacja",
  "amr": [
    "BasicAuthenticator"
  ],
  "iss": "https://ezamowienia.gov.pl/oauth2/token",
  "multiple_organizations": "true",
  "userid": "b5b9b376-11dd-4a99-9996-a14608ec5022",
  "sid": "8a244079-a68d-4a12-8931-b05a6a5d7177",
  "azp": "cvpjEBo8EsIolx9y4IwEtMm_q1Ea",
  "org_name": "Super",
  "exp": 1790951438,
  "iat": 1790947838,
  "jti": "f150ed0c-3504-4f40-ad4f-d81bafb9d26d",
  "email": "jan.kowalski@example.com",
  "groups": [
    "USER",
    "API_CALL",
    "..."
  ],
  "given_name": "Jan",
  "c_hash": "k1zVbDoZ5QzTDsvpzNSLHQ",
  "aud": "cvpjEBo8EsIolx9y4IwEtMm_q1Ea",
  "nbf": 1790947838,
  "org_id": "10084a8d-113f-4211-a0d5-efe36b082211",
  "organization": "aplikacja",
  "organization_id": "6047",
  "phone_number": "123456789",
  "organization_role": [
    "BUYER",
    "SUPPLIER"
  ],
  "family_name": "Kowalski",
  "org_handle": "carbon.super",
  "username": "aplikacja"
}
```