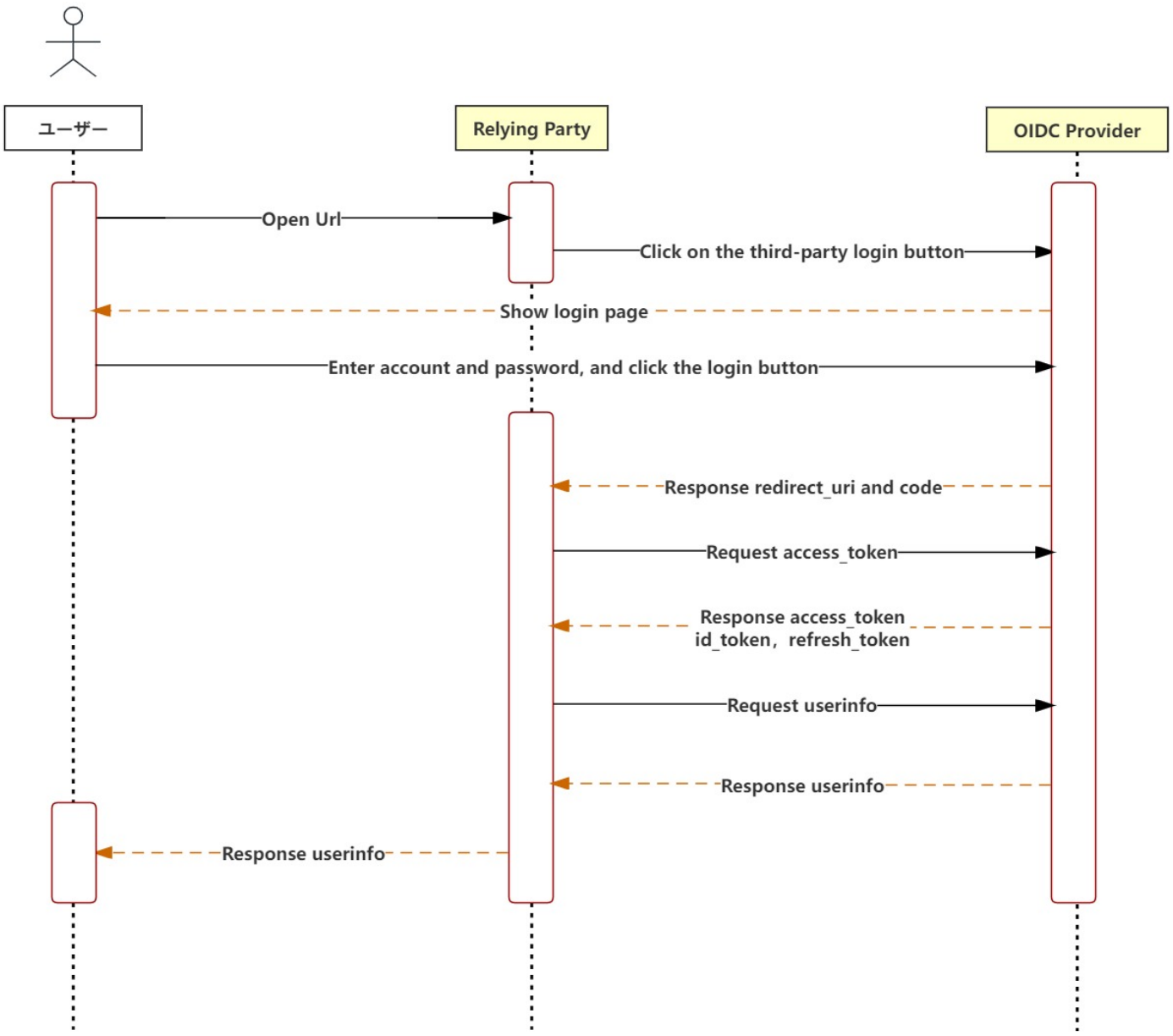
OIDC Provider 图文指南

made by pactera,2023.04.06

1. PC侧



1.1. 端点释义

- **ユーザー**
  End-User(EU)。人類参加者。以下*简称EU*。

- **Relying Party**
  Relying Party(RP),受信任的第三方应用(OAuth2.0中的Client)，需要*EU*完成*鉴*权，并从*OP处获*得*Claim*信息的*应*用。以下*简*称*RP*。

- **OIDC Provider**
  OIDC Provider(OP)，提供身份*认证*的服*务*方，*OAuth2.0*中的*Authorization Server(*授权服*务*器)，用来提供身份*认证*服*务*以及返回*Claim*信息*给*第三方*应*用*(Relying Party)*。以下*简*称*OP*。

1.2 流程

1. EU打开RP第三方登录网页。

2. 点击第三方登录按钮，跳转到OP登录页面。您需要调用「authorize」api，参数验证成功后会自动跳转到OP登录页面。详见「3.1_authorize」。

3. 输入账号密码，点击登录。

4. 授权成功，跳转到RP。（status code: 302 ）

   ```
   redirect_uri?code=xxx&state=xxxx
   ```

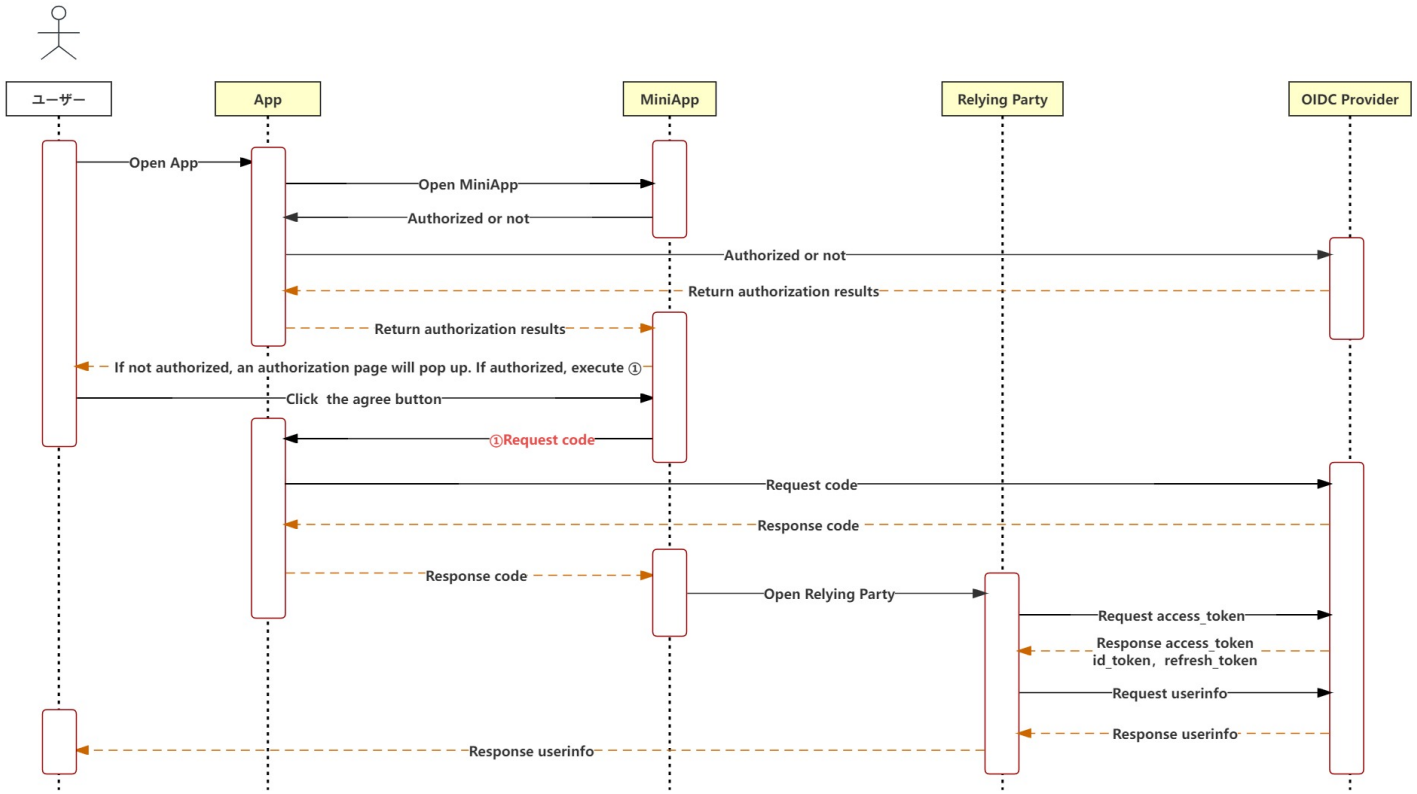
   ※*redirect_uri为*「*3.1_authorize*」*传递*参数，*请*使用 `response?.headers?.path` 接收返回路径。*详见*「*4.5_302route*」。

5. RP利用code获取access_token，refresh_token,id_token。详见「3.2_token」 。

6. RP利用access_token获取用户信息。详见「3.3_userinfo」。

7. RP展示用户信息。

2. 移动端侧



2.1. 端点释义

- **ユーザー**
  End-User(EU)。人類参加者。以下*简*称*EU*。

- **App**
  手机*App*。

- **MiniApp**
  *App*搭*载*的小程序。

- **Relying Party**
  Relying Party (RP)，受信任的第三方*应*用*(OAuth2.0*中的*Client)*，需要*EU*完成*鉴*权，并从*OP处获*得*Claim*信息的*应*用。此*处*在小程序中挂*载*。以下*简*称*RP*。

- **OIDC Provider**
  OIDC Provider(OP)，提供身份*认证*的服*务*方，*OAuth2.0*中的*Authorization Server(*授权服*务*器*)*，用来提供身份*认证*服*务*以及返回*Claim*信息*给*第三方*应*用*(Relying Party)*。以下*简*称*OP*。

2.2 流程

1. EU打开App并登录。

2. EU点击MiniApp图标，打开MiniApp。调用「isAuthorized」api，如果是初次授权则弹出授权页面（见3.），如果有授权记录则略过（见4.）。详见「3.5_isAuthorized」

3. 初次授权则弹出授权页面，EU点击同意按钮。

4. MiniApp通过App调用OP的code接口,获取到code。详见「3.6_authcode」

5. MiniApp跳转到RP，需传递code以及其他必要参数。

6. RP利用code获取到access_token，refresh_token，id_token等。详见「3.2_token」。

7. RP利用access_token获取用户信息。详见「3.3_userinfo」。

8. RP展示用户信息。

3 接口

3.1 アクセスコード取得（/oidc/authorize）

※*请使用 GET/POST 传递参数。当使用POST时*：

> Content-Type: application/x-www-form-urlencoded

请参考（GET）：

> http://www.oidc_example.com/oidc/authorize?client_id=abcde12345&response_type=code&redirect_uri=http%3A%2F%2Fwww.client_example.com%2Findex&scope=openid%2520email%2520profile%2520address&nonce=12345&state=abcde&code_challenge=12345abcdefghijk&code_challenge_method=S256&max_age=12000

パラメータ	タイプ	必須	説明
client_id	String	必須	Client IDの文字列。アプリケーション登録時に発行したClient IDを指定してください。
response_type	String	必須	固定値 `code` 。
redirect_uri	String	必須	アプリケーション登録時に設定したフルURL（もしくはカスタムURIスキーム）を指定してください。事前に登録されているクライアントのリダイレクト URI 値のひとつに正確に一致している必要があります。 ※GET请求时请进行URI编码。
scope	String	必須	UserInfo APIから取得できる属性情報を指定できます。詳見「4.1_scope」。 ※GET请求时请进行URI编码。
state	String	必須	要求と応答の間で維持されるランダム値。
nonce	String	任意	リプレイアタック (opens new window)を防止するための文字列。 この値はレスポンスで返されるIDトークンに含まれます。

パラメータ	タイプ	必須	説明
code_challenge	String	任意	一意のcode_verifierをSHA256で暗号化したうえで、 Base64URL形式にエンコードした値です。详见「4.2_code_challenge」。
code_challenge_method	String	任意	固定値 S256 。（ハッシュ関数SHA256を表します。） code_verifierからcode_challengeを算出する際の暗号化方式を指定します。
max_age	Number	任意	ユーザー認証後に許容される最大経過時間（秒）。max _Mageの場合、 返されるID Tokenにはauth _time Claim Value。

正常系示范：（status code: 302 ）
输入账号密码并完成登录操作。

> http://www.client_example.com/index?code=xz19ssafynqppsnosznhq3eob9bvxxjcqw5c&state=abcde

パラメータ	説明
code	アクセストークンの取得に使用される認可コード。有効期間は10分です。また、認可コードは1回のみ利用可能です。
state	リクエスト時に指定されたstate値。

异常系示范：（status code: 4xx ）

> http://www.oidc_example.com/error?error=invalid_request&error_description=パラメーター%7Bclient_id%2Credirect_uri%2Cresponse_type%7Dは必須です&state=abcde

パラメータ	説明
error	エラーコード。 - invalid_request：パラメータは必須です，或者参数的值无效。 - unsupported_response_type：response_typeの値が無効です。 - invalid_scope：scopeの値が無効です。 - unauthorized_client：指定されたclient_idはサポートされていません。
error_description	エラー内容。
state	リクエスト時に指定されたstate値。

3.2 コードよりトークンを取得（/oidc/token）

※*请使用 POST 传递参数。当使用POST时*：

> Content-Type: application/x-www-form-urlencoded

Headers:
请对client_secret进行Base64编码，并在Headers中传递参数。
※Client IDがクライアントサイド・アプリケーションとして発行された場合は指定する必要はありません。

> Authorization: Basic czZCaGRSa3F0MzpnWDFmQmF0M2JW

Body:

パラメータ	タイプ	必須	説明
client_id	String	必須	Client IDの文字列。
grant_type	String	必須	固定値 `authorization_code` 。
redirect_uri	String	必須	同意処理後の戻り先であるURLまたはCustom URI Scheme事前に登録されているクライアントのリダイレクト URI 値のひとつに正確に一致している必要があります。 请与请求code时传递的redirect_uri参数保持一致。
code	String	必須	Authorizationエンドポイントで取得した認可コードの文字列を指定してください。
code_verifier	String	必須	半角英数字（a～z、A～Z、0～9）および記号（-._~）からなるランダムな文字列。 建议50文字以下。 ※認可リクエスト時にcode_challengeを指定した場合は必須です。

正常系示范：（status code: 200 ）

```
{
    "access_token": "o1qjtqobs4j68nxwbytrxa80aj3xkut7uklg",
    "refresh_token": "60ubebp35maraih4qgrpfshov0qzdphbdk49",
    "token_type": "Bearer",
    "expires_in": "3600",
    "id_token": "eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCIsImtpZCI6Inlya2x3MmdnemwxNjc3ODMwODAyNTkzIn0.eyJhdF9oYXNoIjoiTVRBd01
    "scope": "openid email profile"
}
```

パラメータ	説明
access_token	APIへアクセスするのに使用します。
refresh_token	Access Tokenを更新するときに使用します。有効期限は4週間です。
token_type	Web APIへアクセスする際にAccess Tokenを適切に用いるために必要な情報を提供します。Bearer Token形式です。固定値 `Bearer` 。
expires_in	Access Tokenの有効期限を表す秒数です。
id_token	ユーザー認証情報を含む改ざん検知用の署名付きトークンです。详见「4.3_id_token」。
scope	アクセストークンに紐付く許諾されたスコープ ， 許可サーバーは「スコープ」応答パラメーターを使用して、 発行されたアクセストークンのスコープをクライアントに通知します。

异常系示范：（status code: 4xx ）

```
{
  "error": "invalid_grant",
  "error_description": "code無効またはタイムアウト"
}
```

パラメータ	説明
error	エラーコード。 - invalid_request：パラメータは必須です，或者参数的值无效。

パラメータ	説明
	- unsupported_grant_type：grant_typeの値が無効です。 - invalid_grant：codeが無効または期限切れです。 - invalid_client：指定されたclient_idはサポートされていません、client_secretの値が無効です。
error_description	エラー内容。

3.3 ユーザー情報照会、サードパーティーに取得できるように（/oidc/userinfo）

※*请*使用 *GET/POST 传递*参数。

请参考（GET）：

> http://www.oidc_example.com/oidc/userinfo

Headers:
Access Tokenの文字列を指定してください。（Bearer Token形式）。

> Authorization: Bearer 0bta4dzc5om5tcs5mqte5fpcacy5r83n1emj

正常系示范：（status code: 200 ）

```
{
  "sub": "4d4f7fafd2286f8dc7e8d7fffad424f6",
  "name": "太郎",
  "given_name": null,
  "family_name": null,
  "nickname": null,
  "picture": null,
  "birthdate": "1992-01-01",
  "gender": "1",
  "email": "jphxxx@chbank.co.jp",
  "phone_number": "080-6665-xxxx",
  "address": {
    "formatted": "千葉市中央区末広４丁目１−２−xxx",
    "country": null,
    "region": null,
    "locality": null,
    "postal_code": "260-0843"
  }
}
```

パラメータ	説明	対応するscope
sub	ユーザー識別子	openid
name	姓名	profile
given_name	名	profile
family_name	姓	profile
nickname	表示名	profile
picture	プロフィール画像URL	profile

パラメータ	説明	対応するscope
birthdate	生年	profile
gender	性別，1：男性2：女性	profile
email	メールアドレス	email
phone_number	電話番号	phone
address	- formatted：都道府県 + 市区町村 - country：国コード - region：都道府県 - locality：市区町村 - postal_code：郵便番号	address

异常系示范：（status code: 4xx ）

```
{
  "error": "invalid_token",
  "error_description": "accessToken無効またはタイムアウト"
}
```

パラメータ	説明
error	エラーコード。 - invalid_request：パラメータは必須です，或者参数的值无效。 - invalid_token：accessTokenが無効または期限切れです。
error_description	エラー内容。

3.4 利用kid取得public_key（/keys/{kid}）

※*请*使用 *GET 传递*参数。

请参考：

> http://www.oidc_example.com/keys/{kid}

path参数

パラメータ	タイプ	必須	説明
kid	String	必須	署名検証に用いる公開鍵のKey ID。存放于id_token的header中。

正常系示范：（status code: 200 ）

```
{
  "kid": "12345",
  "publicKey": "wwwwDQYJKoZIhvcNAQEBBQADSwAwSAJBAIdVddpMZCJVmnE+s7ZvRUwUiPsUBeLD12//q3Lo29G7fHWhMowGHOdN32rFuDhaaE4nhO/G
}
```

パラメータ	説明
kid	ID TokenのHeaderに含まれているKey ID。Signatureを生成したPrivate Keyの対になるPublic Keyを判定するために利用。
publicKey	Signatureを生成したPrivate Keyの対になるPublic Key文字列。

異常系示范：（status code: 4xx ）

```
{
  "error": "invaild_kid",
  "error_description": "公開鍵が取得できませんでした。kidの値をご確認ください。"
}
```

パラメータ	説明
error	エラーコード。 - invaild_kid：公開鍵が取得できませんでした。kidの値をご確認ください。
error_description	エラー内容。

3.5 小程序是否取得用户授权（/auth/isAuthorized）

※*请*使用 *GET/POST 传递*参数。当使用*POST时*：

> Content-Type: application/x-www-form-urlencoded

请参考：

> http://www.oidc_example.com/auth/isAuthorized?userId=00000&miniapp_id=321008553559111111

パラメータ	タイプ	必須	説明
userId	String	必須	appのユーザーID。
miniapp_id	String	必須	クライアントID（ mini app id）。

正常系示范：（status code: 200 ）

```
{
    "isAuthorized": true
}
```

パラメータ	説明
isAuthorized	true=承認済み，false=没有授权过。

異常系示范：（status code: 4xx ）

```
{
  "error": "unauthorized_client",
  "error_description": "サポートされていないクライアント"
}
```

パラメータ	説明
error	エラーコード。 - invalid_request：パラメータは必須です，或者参数的值无效。 - invalid_client：指定されたclient_idはサポートされていません。
error_description	エラー内容。

3.6 認可コードを取得する（/oidc/authcode）

※*请使用 GET/POST 传递参数。当使用POST时*：

> Content-Type: application/x-www-form-urlencoded

请参考（GET）：

> http://www.oidc_example.com/oidc/authcode?userId=00000&client_id=abcde12345&response_type=code&redirect_uri=http%3A%2F%2Fwww.client_example.com%2Findex&scope=openid%2520email%2520profile%2520address&nonce=12345&state=abcde&code_challenge=12345abcdefghijk&code_challenge_method=S256&max_age=12000

パラメータ	タイプ	必須	説明
userId	String	必須	appのユーザーID。
client_id	String	必須	クライアントID（ mini app id）。
response_type	String	必須	固定値 `code` 。
redirect_uri	String	必須	アプリケーション登録時に設定したフルURL（もしくはカスタムURIスキーム）を指定してください。事前に登録されているクライアントのリダイレクト URI 値のひとつに正確に一致している必要があります。 ※GET请求时请进行URI编码。
scope	String	必須	UserInfo APIから取得できる属性情報を指定できます。详见「4.1_scope」。 ※GET请求时请进行URI编码。
state	String	必須	要求と応答の間で維持されるランダム値。
nonce	String	任意	リプレイアタック (opens new window)を防止するための文字列。 この値はレスポンスで返されるIDトークンに含まれます。
code_challenge	String	任意	一意のcode_verifierをSHA256で暗号化したうえで、 Base64URL形式にエンコードした値です。详见「4.2_code_challenge」。
code_challenge_method	String	任意	固定値 `S256` 。（ハッシュ関数SHA256を表します。） code_verifierからcode_challengeを算出する際の暗号化方式を指定します。
max_age	Number	任意	ユーザー認証後に許容される最大経過時間（秒）。max _Mageの場合、返されるID Tokenにはauth _time Claim Value。

正常系示范：（status code: 200 ）

```
{
    "code": "pnmdm7kl88ily6ghfjxyg3jmu6nfyy4bjpao",
    "state": "abcde"
}
```

パラメータ	説明
code	認可コード。
state	要求と応答の間で維持されるランダム値。

異常系示范：（status code: 4xx ）

```
{
    "error": "unsupported_response_type",
    "error_description": "response_typeの値が無効です",
    "state": "abcde"
}
```

パラメータ	説明
error	エラーコード。 - invalid_request：パラメータは必須です，或者参数的值无效。 - unsupported_response_type：response_typeの値が無効です。 - invalid_scope：scopeの値が無効です。 - unauthorized_client：指定されたclient_idはサポートされていません。
error_description	エラー内容。
state	要求と応答の間で維持されるランダム値。

3.7 トークンの有効期限が切れたあと、refresh tokenでトークンをリフレッシュする（/oidc/token）

※*请*使用 *POST 传递*参数。当使用*POST时*：

> Content-Type: application/x-www-form-urlencoded

Headers:
请对client_secret进行Base64编码，并在Headers中传递参数。
※Client IDがクライアントサイド・アプリケーションとして発行された場合は指定する必要はありません。

> Authorization: Basic czZCaGRSa3F0MzpnWDFmQmF0M2JW

Body:

パラメータ	タイプ	必須	説明
client_id	String	必須	Client IDの文字列。
grant_type	String	必須	固定値 `refresh_token` 。
refresh_token	String	必須	トークンエンドポイント（新規発行）より返却されたリフレッシュトークン。

正常系示范：（status code: 200 ）

```
{
    "access_token": "1u8d3kzf003xu96wfg9m5x9l4wia4wgcxhum8",
    "refresh_token": "1kki32g3khz3c60q17dsiuekp0jcevjvlq8ki",
    "token_type": "Bearer",
    "expires_in": "3600",
    "id_token": "1eyJhbGciOiJSUzI1NiIsInR5cCI6Imp3dCIsImtpZCI6Inlya2x3MmdnemwxNjc3ODMwODAyNTkzIn0.eyJhdF9oYXNoIjoiTVRBd0
    "scope": "openid email profile"
}
```

パラメータ	説明
access_token	APIへアクセスするのに使用します。
refresh_token	Access Tokenを更新するときに使用します。有効期限は4週間です。
token_type	Web APIへアクセスする際にAccess Tokenを適切に用いるために必要な情報を提供します。Bearer Token形式です。固定値 `Bearer` 。
expires_in	Access Tokenの有効期限を表す秒数です。
id_token	ユーザー認証情報を含む改ざん検知用の署名付きトークンです。详见「4.3_id_token」。
scope	アクセストークンに紐付く許諾されたスコープ ， 許可サーバーは「スコープ」応答パラメーターを使用して、発行されたアクセストークンのスコープをクライアントに通知します。

异常系示范：（status code: 4xx ）

```
{
    "error": "invalid_grant",
    "error_description": "refresh_tokenが無効または期限切れ"
}
```

パラメータ	説明
error	エラーコード。 - invalid_request：パラメータは必須です，或者参数的值无效。 - unsupported_grant_type：grant_typeの値が無効です。 - invalid_grant：refresh_tokenが無効または期限切れ。 - invalid_client：指定されたclient_idはサポートされていません、client_secretの値が無効です。
error_description	エラー内容。

4. 附录

4.1 scope

*scope*パラメータに指定できるスコープは以下のとおりです。複数のスコープを指定するには、*URL*エンコードされた空白文字（*%20*）で区切って指定します。

範囲値	説明
openid	（必須）ユーザー識別子を返却します。
profile	姓名・生年・性別を返却します。
email	メールアドレスと確認済みフラグを返却します。
address	ユーザー登録住所情報を返却します。
phone	連絡先番号。

4.2 code_challenge

code_verifierおよびcode_challengeの生成方法は下記をご確認ください。
※コードはJava言語を使用する

```
    /**
     * @param code_verifier
     * @Description: code_challengeの生成
     * @return: java.lang.String
     */
    public String createChallenge(String code_verifier) throws NoSuchAlgorithmException {
        MessageDigest digest = MessageDigest.getInstance("SHA-256");
        byte[] encodedhash = digest.digest(code_verifier.getBytes(StandardCharsets.UTF_8));
        String hexString  = bytesToHex(encodedhash);
        return java.util.Base64.getUrlEncoder().withoutPadding().encodeToString(hexString .getBytes(StandardCharse
    }
    /**
     * @param hash
     * @Description: バイトから16進
     * @return: java.lang.String
     */
    private String bytesToHex(byte[] hash) {
        StringBuilder hexString = new StringBuilder(2 * hash.length);
        for (int i = 0; i < hash.length; i++) {
            String hex = Integer.toHexString(0xff & hash[i]);
            if (hex.length() == 1) {
                hexString.append('0');
            }
            hexString.append(hex);
        }
        return hexString.toString();
    }
```

4.3 id_token

ID Tokenとはユーザー認証情報を含む改ざん検知用の署名付きTokenであり、JWT（JSON Web Token）フォーマットでエンコードされています。
JWTはピリオド（"."）区切りのHeader、Payload、Signatureから構成されます。

Header

署名のアルゴリズムなどJWTを取り扱うために必要なパラメーターが含まれます。

範囲値	説明
typ	固定値 `JWT` 。
alg	固定値 `RS256` 。
kid	Public KeysエンドポイントのPublic Keyのキー名に一致します。详见「4.4_public_key」。

Payload

ユーザー識別子や認証時刻などユーザーの認証情報のパラメーターが含まれます。

範囲値	説明
iss	ID Tokenの発行元（固定値）。
sub	ユーザー識別子,openid。
aud	请求时传入的Client ID。
iat	ID Tokenの発行時刻のUNIXタイムスタンプ。
nbf	生效时间。
nonce	RP送信要求時に提供されるランダム文字列。
auth_time	EUが認証を完了した時間。RPがAuthN要求を送信するときにmax _Mageのパラメータを指定するには、 このClaimが必要です。
exp	ID Tokenの有効期限のUNIXタイムスタンプ。
at_hash	Access Tokenのハッシュ値。Access TokenをID TokenのSignature生成時と同じハッシュアルゴリズム（SHA256）でハッシュ化し、オクテッドの前部をBase64URLエンコードした文字列。详见「4.4_at_hash」。

Signature

HeaderとPayloadから生成された改ざん防止のための署名です。

Base64URLエンコードされたHeader + "." + Payloadを入力値としてRSA-SHA256で署名した結果をBase64URLエンコードした文字列です。

検証手順

- 取得したJWTをピリオド（"."）で区切りHeader、Payload、Signatureの3つに分割します。
- Headerに含まれるkid値を用いて、Public Keysエンドポイントからkid値（またはキー名）から対になるPublic Keyを取得します。详见「3.4_public_key」。
- 验证签名是否正确，参照以下代码：
  ※コードはJava言語を使用する

```java
    /**
     * @param idToken
     * @param publicKey
     * @Description: Verify signature
     * @return: boolean
     */
    public boolean verify(String idToken, String publicKey) throws Exception {
        try {
            Algorithm algorithm = Algorithm.RSA256((RSAPublicKey) generateRsaPublic(publicKey), null);
            JWTVerifier verifier = com.auth0.jwt.JWT.require(algorithm).withIssuer("固定值，请与OP确认").build();
            verifier.verify(idToken);
            return true;
        } catch (Exception e) {
            e.printStackTrace();
            return true;
        }
    }
    /**
     * @param key
     * @Description: Generate a public key
     * @return: java.security.PublicKey
     */
    private PublicKey generateRsaPublic(String key) throws Exception {
        KeyFactory rsa_fact = KeyFactory.getInstance("RSA");
        byte[] data = Base64.decode((key.getBytes()));
        X509EncodedKeySpec spec = new X509EncodedKeySpec(data);
        return rsa_fact.generatePublic(spec);
    }
```

- 如果请求传递了nonce参数，请检查Payload中是否存在nonce，并检查值是否与传递的相同。
- 请检查Headers中的iss值，是否与OP提供的值相同。
- 请检查Payload中的aud值，确认是否当前客户端的client_id。
- Access TokenとID Tokenを同時に発行する場合には、Access TokenをHeaderのalgと同じハッシュアルゴリズム（SHA256）でハッシュ化し、オクテッドの前部をBase64URLエンコードした文字列とPayloadのat_hash値が一致していることを検証します。详见「4.4_at_hash」。
- ID Tokenの有効期限を確認するために、Payloadのexp値が検証時のUNIXタイムスタンプの値よりも大きいことを検証します。

4.4 at_hash

Access Tokenのハッシュ値，Access TokenをID TokenのSignature生成時と同じハッシュアルゴリズム（SHA256）でハッシュ化し、オクテッドの前部をBase64URLエンコードした文字列。
※コードはJava言語を使用する，iss请向OP确认。

```java
    /**
     * @param accessToken
     * @Description: use accessToken to create at_hash
     * @return: java.lang.String
     */
    public String createHash(String accessToken) throws Exception {
        MessageDigest digest = MessageDigest.getInstance("SHA-256");
        byte[] encodedhash = digest.digest(accessToken.getBytes(StandardCharsets.UTF_8));
        String hexString = bytesToHex(encodedhash);
        BigInteger sint = new BigInteger(hexString, 16);
        //10進2進
        return java.util.Base64.getUrlEncoder().withoutPadding().encodeToString(sint.toString(2).substring(0, 128)
    }
    /**
     * @param hash
     * @Description: バイトから16進
     * @return: java.lang.String
     */
    private String bytesToHex(byte[] hash) {
        StringBuilder hexString = new StringBuilder(2 * hash.length);
        for (int i = 0; i < hash.length; i++) {
            String hex = Integer.toHexString(0xff & hash[i]);
            if (hex.length() == 1) {
                hexString.append('0');
            }
            hexString.append(hex);
        }
        return hexString.toString();
    }
```

4.5 302route

以下代码是axios的示范代码，用来接收OP返回的302地址，请参照。

```javascript
request.interceptors.response.use(
  function (response: AxiosResponse<any>) {
    return Promise.reject(response.data)
  },
  function (error) {
    if (error.response?.status === 302) {
      location.href = error.response?.headers?.path
    } else {
      return Promise.reject(error.response)
    }
  }
)
```

The End