

# Sandbox příručka API OAUTH2

## Change log

Date	Version	Description
10.02.2020	7	Oprava chybně uvedeného hesla v Přihlášení
26.02.2020	8	Aktualizace dokumentu a printscreenů
22.05.2020	9	Úprava dokumentu
09.03.2021	10	Odstranění varianty přímého volání (vybraných resource) - pro registraci, změnu nebo smazání aplikace, vč. požadavku na client secret prostřednictvím přímého volání - tj. odstranění podkapitol 2.1 až 2.5 ve srovnání s předchozí verzí dokumentu č. 9 a přečíslování zbývajících podkapitol v kapitole 2.
30.05.2024	11	Doplnění úpravy odkazu na identity server v rámci 1 SCA flow na straně 10.
20.08.2024	12	Úprava příručky (zejména aktualizace printscreenů) pro nový API store – nové prostředí pro klienty.

## Obsah

Sandbox příručka .....	1
API OAUTH2.....	1
Hlášení chyb .....	4
1 Postup vygenerování authorization_code/refresh token pro aplikaci .....	4
1.1 Předpoklady přístupu k aplikacím .....	4
1.2 Vstup do aplikačního menu .....	4
1.3 Rozkliknutí aplikace .....	5
1.4 Zadání callback URL.....	6
1.5 Generování Consumer Key a Consumer Secret .....	7
1.6 Uložení Consumer Key a Consumer Secret .....	7
1.7 Odběr (subscription) API.....	8
1.8 Generování access tokenu .....	8
1.9 Získání code.....	9
1.10 API menu .....	10
1.11 Vybrání API OAUTH2.....	10
1.12 Vstoupení do API OAUTH2 .....	11
1.13 Vybrání operace „/token“ na otestování .....	11
1.14 Vyplnění požadovaných polí u operace „/token“ .....	11
1.15 Vypsání chyby u operace „/token“ .....	13
1.16 Vybrání operace „/revoke“ na otestování.....	14
1.17 Vyplnění požadovaných polí u operace „/revoke“ .....	15
1.18 Vypsání chyby u operace „/revoke“ .....	16
2 Postup přístupu do aplikace přes přímé volání.....	17
2.1 Charakteristika requestu Získání/vystavení tokenu.....	17
2.2 Charakteristika requestu Zneplatnění tokenu.....	19
2.3 Charakteristika requestu na Autorizační resource.....	20

## Hlášení chyb

**Hlášení chyb Sandboxu nebo jeho volání probíhá vždy pomocí emailové schránky [api@kb.cz](mailto:api@kb.cz). Odeslaný email musí obsahovat níže uvedené náležitosti. V případě chybějící požadované informace nebude možné dotaz nebo chybu zpracovat.**

PSD2 API: CZ, SK

Prostředí: Sandbox, Produkce

Zda bylo voláno z FE Sandbox vč. typu a verze použitého prohlížeče nebo v případě BE volání název a verzi programu pro BE volání

Typ volání

Datum a čas uskutečněného volání

IP adresu

Chybu a její co nejpřesnější popis, který může být doplněn o příslušný otisk obrazovky

**Bez výše uvedených hodnot se není možné hlášenou chybou zabývat.**

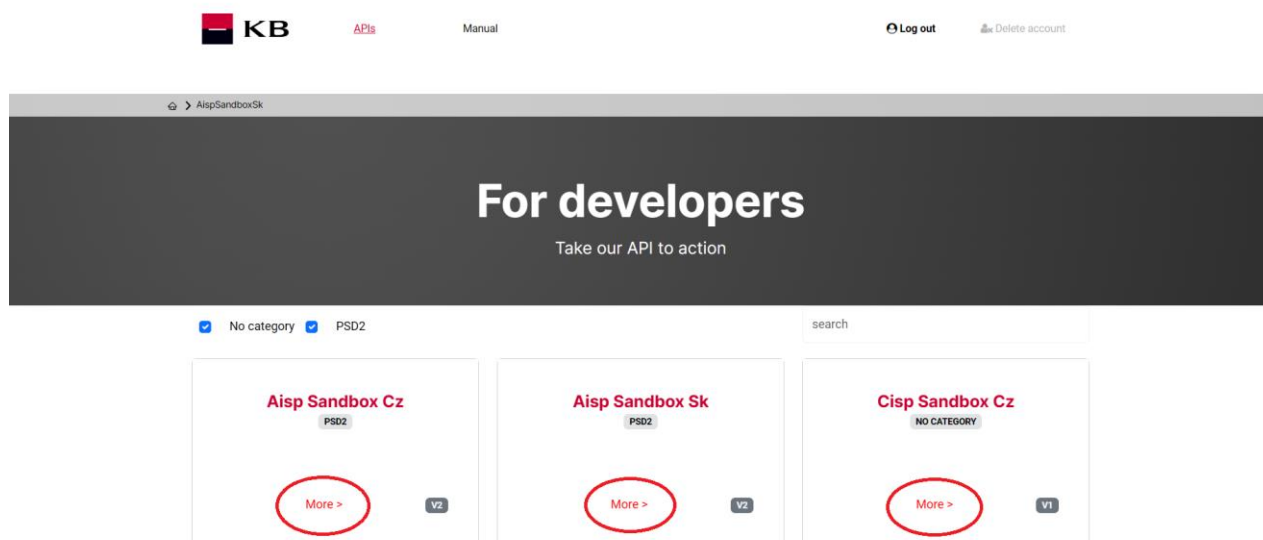
## 1 Postup vygenerování authorization\_code/refresh token pro aplikaci

### 1.1 Předpoklady přístupu k aplikacím

Uživatel je řádně zaregistrován a přihlášen. – viz Příručka Sandbox registrace uživatele.

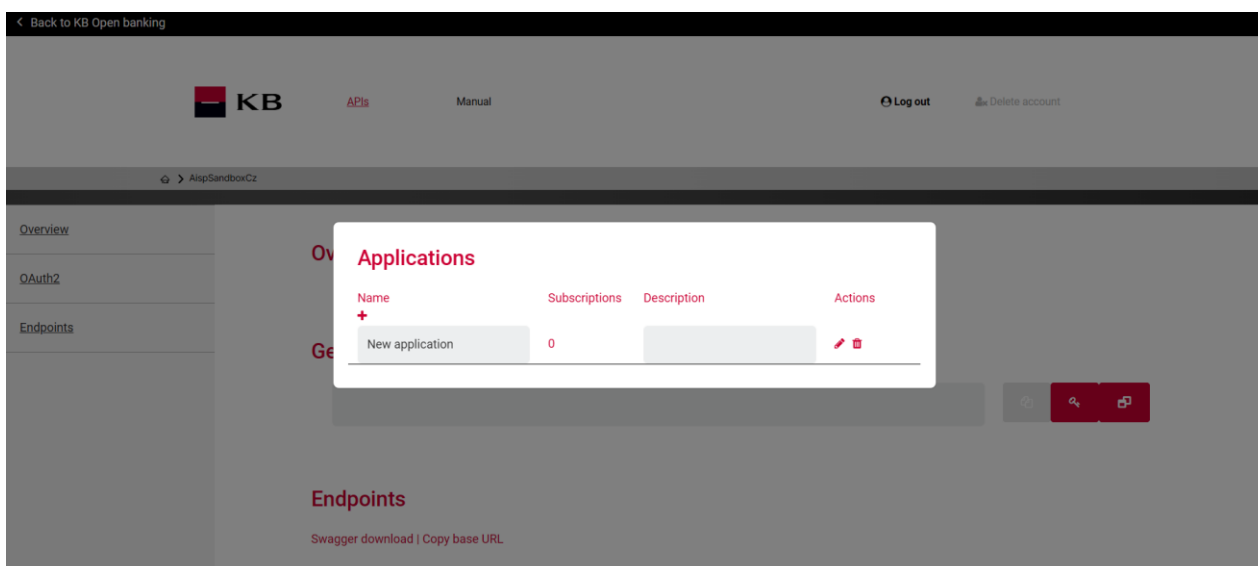
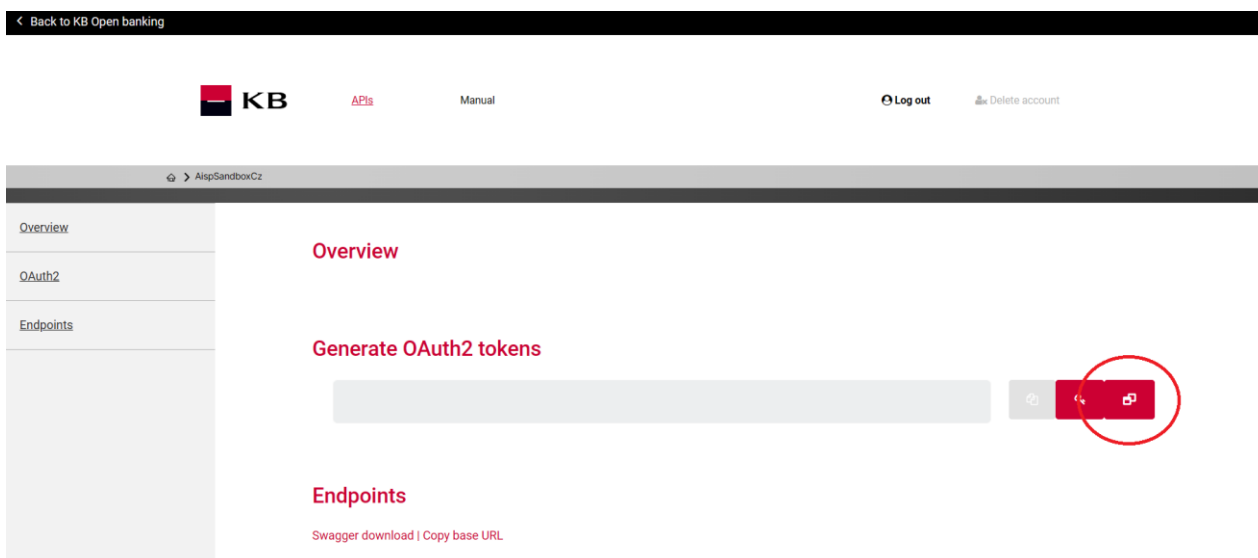
### 1.2 Vstup do aplikačního menu

Přihlášený uživatel může kliknutím na nápis „More“ přejít na detail vybraného PSD2 API a zde kliknutím na tlačítko „Applications“ vstoupit do menu, kde si může zaregistrovat své aplikace.



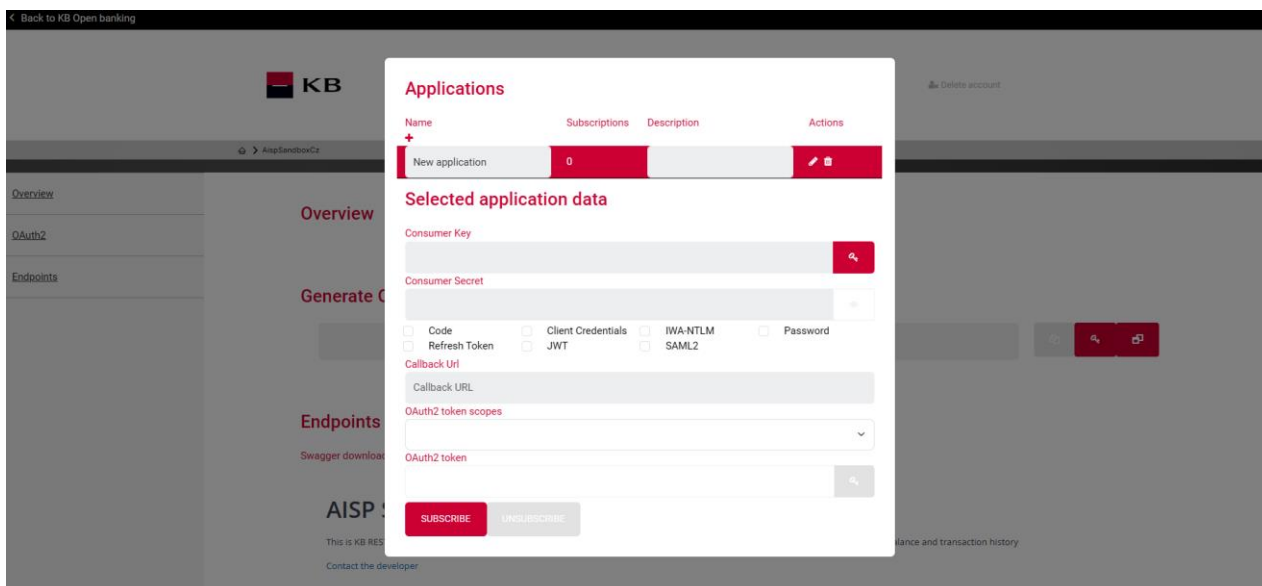


# Applications



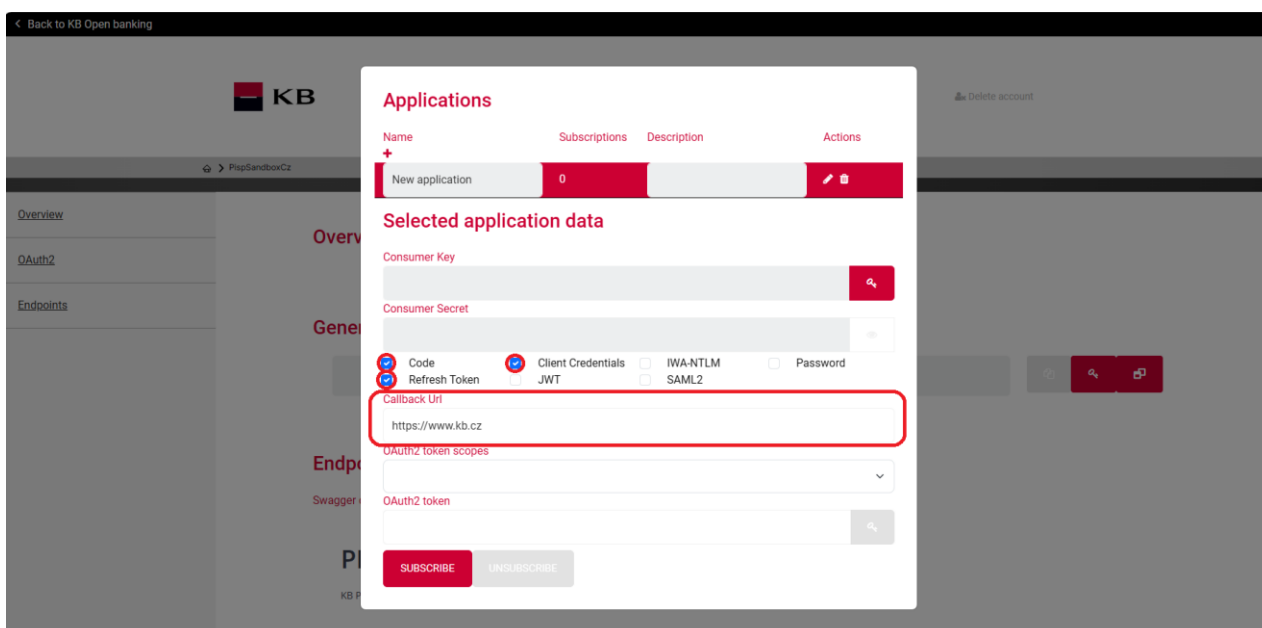
## 1.3 Rozkliknutí aplikace

Uživatel klikne na řádek s aplikací, načtež se mu zobrazí náhled vybrané aplikace.



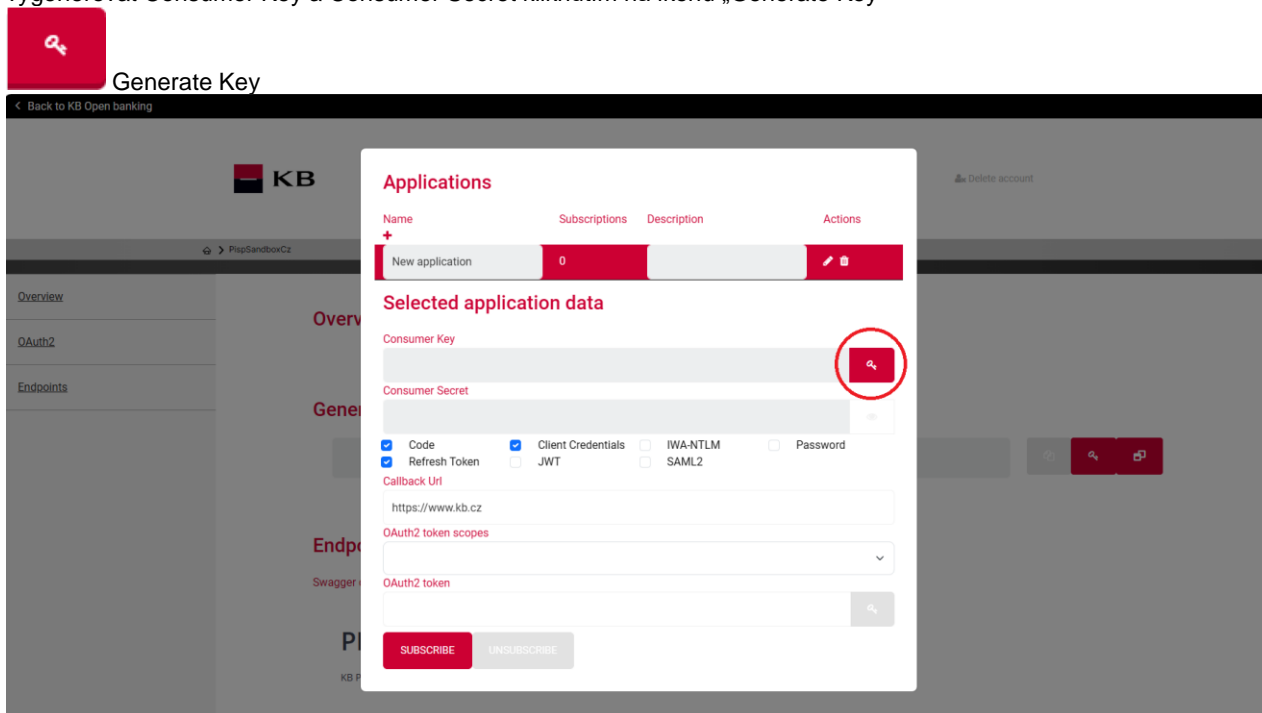
## 1.4 Zadání callback URL

Uživatel zaškrtně checkbox s názvem „Code“, „Refresh Token“ a „Client Credentials“ a pak zadá do textového pole „Callback URL“, např. hodnotu <https://www.kb.cz>.



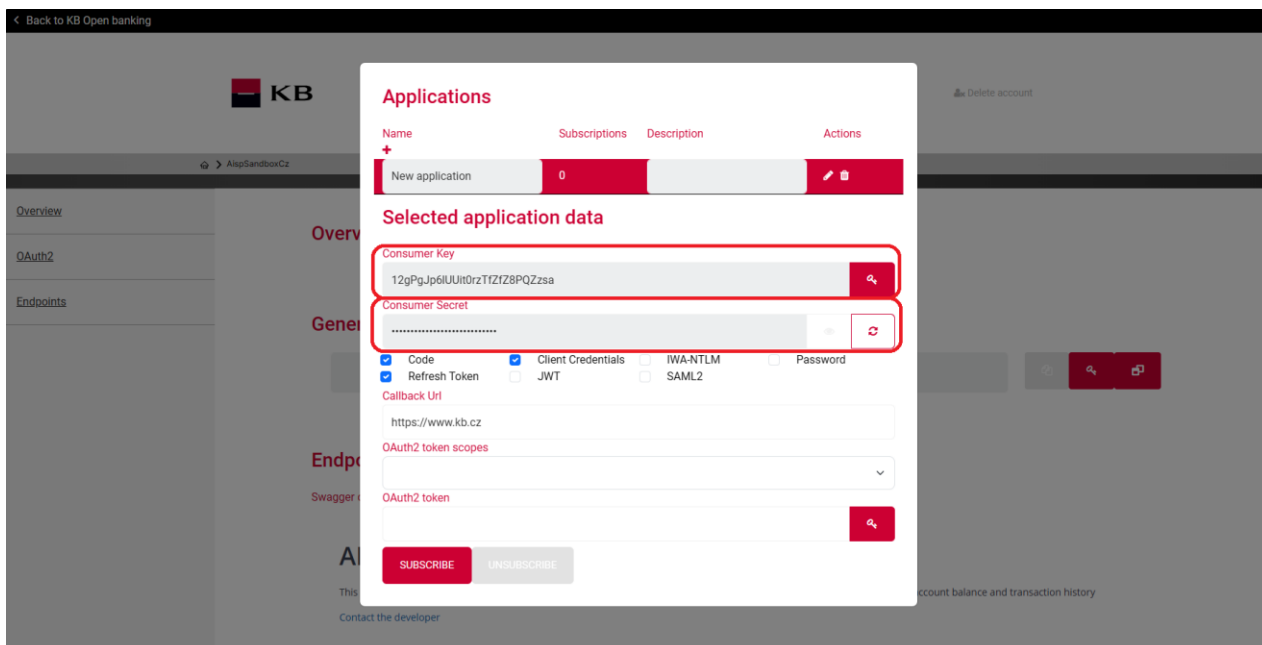
## 1.5 Generování Consumer Key a Consumer Secret

Uživatel poté, pokud byly splněny podmínky pro vygenerování authorization\_code/refresh token (viz. 1.4), může vygenerovat Consumer Key a Consumer Secret kliknutím na ikonu „Generate Key“



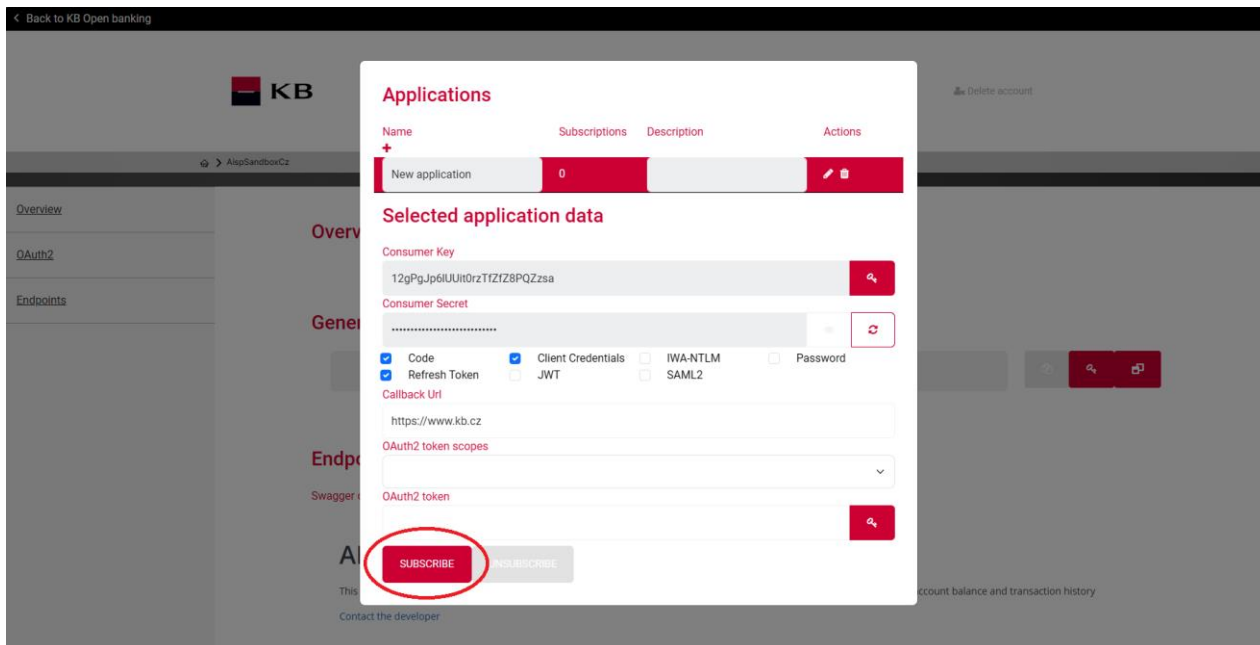
## 1.6 Uložení Consumer Key a Consumer Secret

Přihlášený uživatel si zkopíruje (např.: do Notepadu) hodnoty v polích „Consumer Key“ a „Consumer Secret“.



## 1.7 Odběr (subscription) API

Hlavním předpokladem, který musí být splněn pro to, aby se dané aplikaci mohl vygenerovat authorization\_code/refresh token je ten, že daná aplikace musí mít přihlášený odběr (subscription) k API, jenž tuto možnost umožňuje a využívá (např.: AISP, PISP, ...).

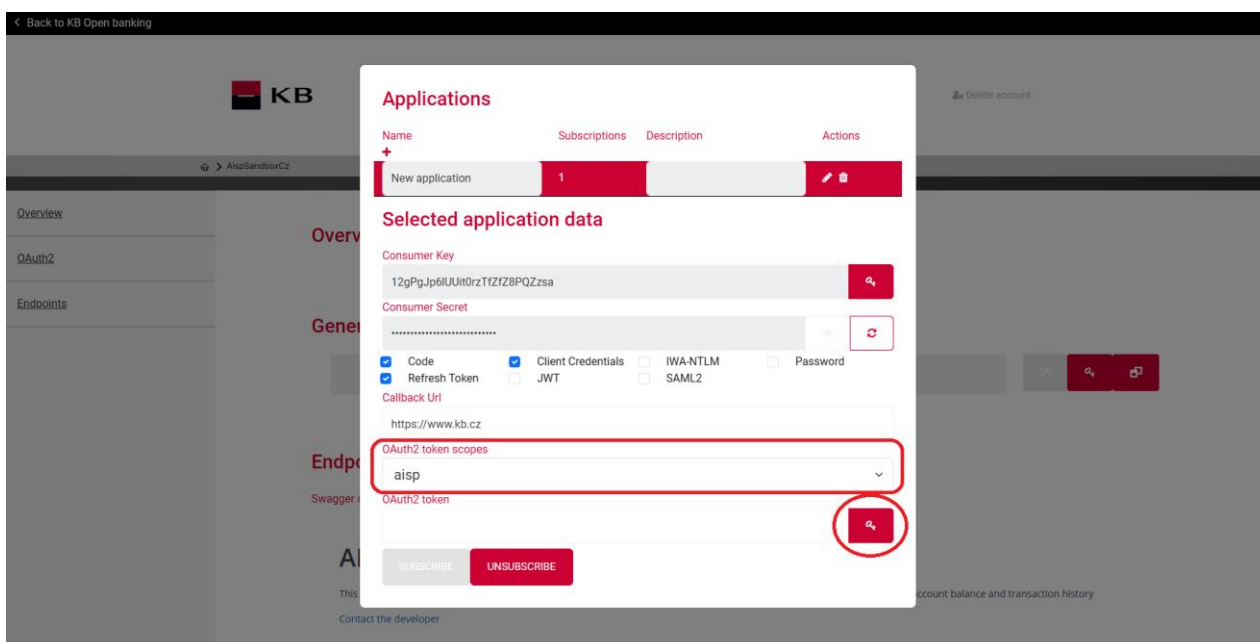


## 1.8 Generování access tokenu

Uživatel, pokud má vygenerován Consumer Key a Consumer Secret (viz. 1.6), má možnost si vygenerovat access token přímo ve webovém rozhraní. V poli „OAuth2 token scopes“ vybere požadovaný scope podle API, které bude s tokenem provolávat a poté klikne na tlačítko „Generate Token“. Výsledkem je vygenerovaný access token v poli „OAuth2 token“.



Generate Token





The screenshot displays the 'Applications' management interface in the KB Open banking developer portal. A modal window titled 'Selected application data' is open, showing configuration for a new application. The application is named 'New application' and has a subscription count of '1'. The configuration includes:

- Consumer Key:** 12gPgJp6IUUi0rzTfZfZ8PQZssa
- Consumer Secret:** [Redacted]
- Authentication Methods:** Code (checked), Refresh Token (checked), Client Credentials (checked), JWT (unchecked), IWA-NTLM (unchecked), SAML2 (unchecked), Password (unchecked).
- Callback Url:** https://www.kb.cz
- OAuth2 token scopes:** aisp
- OAuth2 token:** eyJ4NXQlOJpVGhsT1dFeU5qRTBOR0kxTXpnNFpUSTFOVGMxWidGak5HRXIPVFUwWkdNI [Redacted]
- Expiration:** Expires in 31. 7. 2024 11:39:38

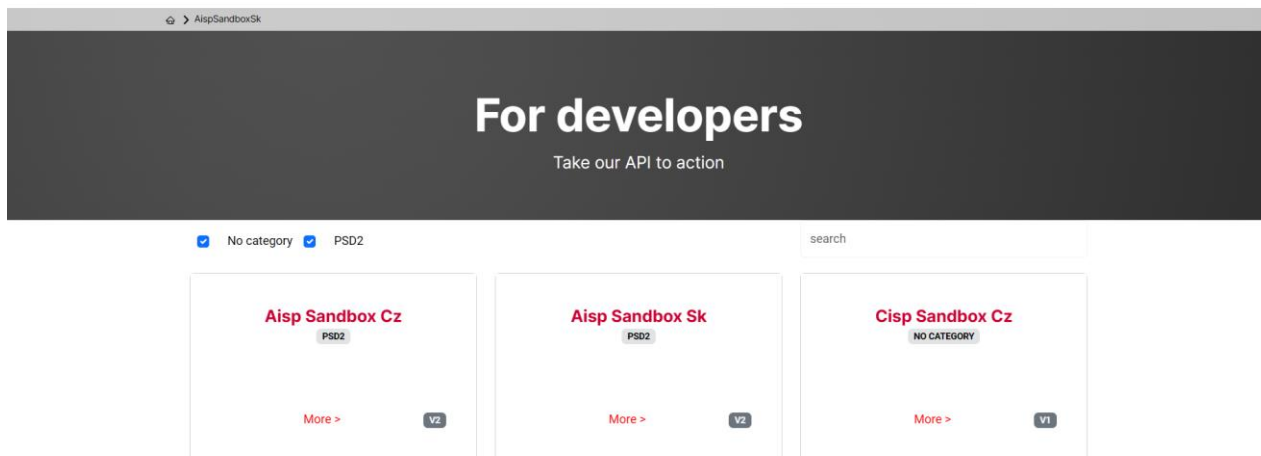
Buttons for 'SUBSCRIBE' and 'UNSUBSCRIBE' are visible at the bottom of the modal.

## 1.9 Získání code

Oproti produkčnímu řešení si klient nemusí generovat code, který je potřebný pro získání access tokenu přes OAuth2 API, ale použije statickou hodnotu „**b6487518-f321-4290-aa32-16fe5cc42b81**“. Tuto hodnotu používá při plnění polí OAuth2 API.

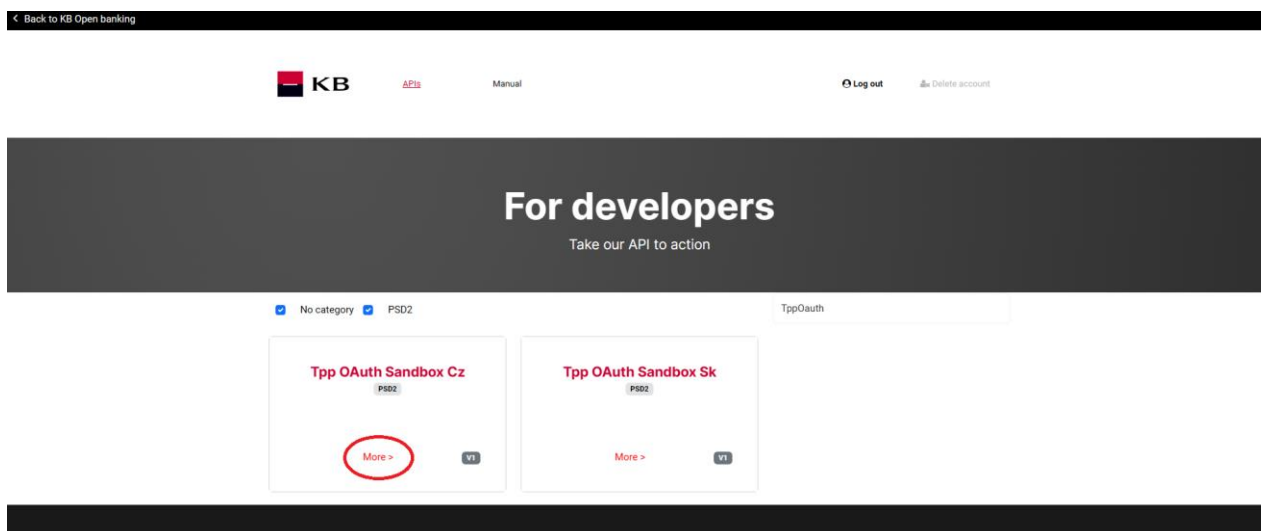
## 1.10 API menu

Uživatel poté kliknutím na nápis „APIs“ v horní části obrazovky vstoupí do menu, kde jsou všechna API, ke kterým má přístup.



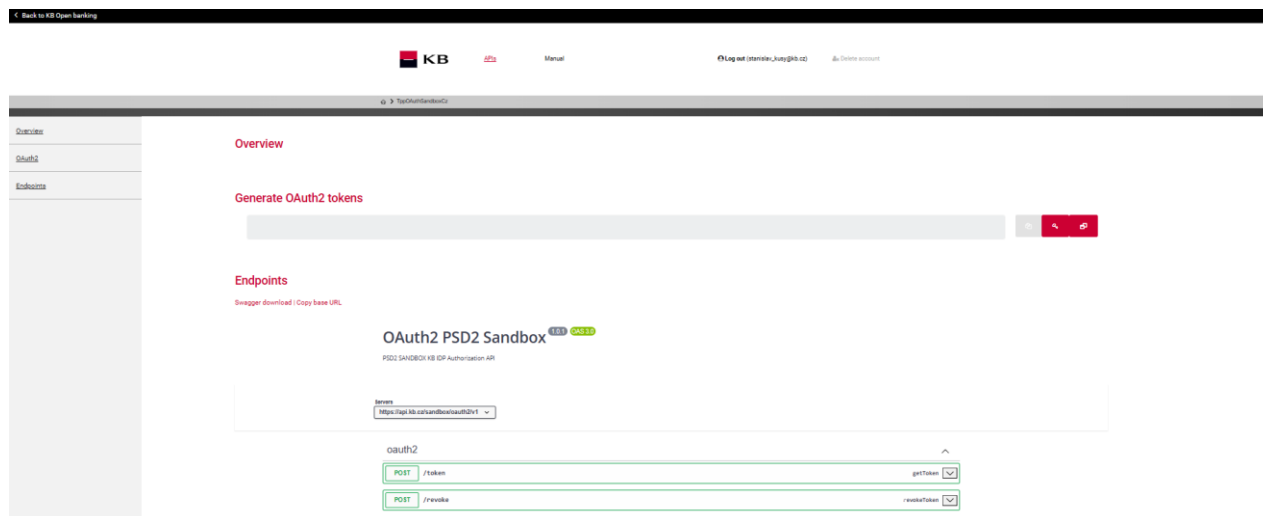
## 1.11 Vybrání API OAUTH2

Kliknutím na „More“ si uživatel zobrazí dané API.



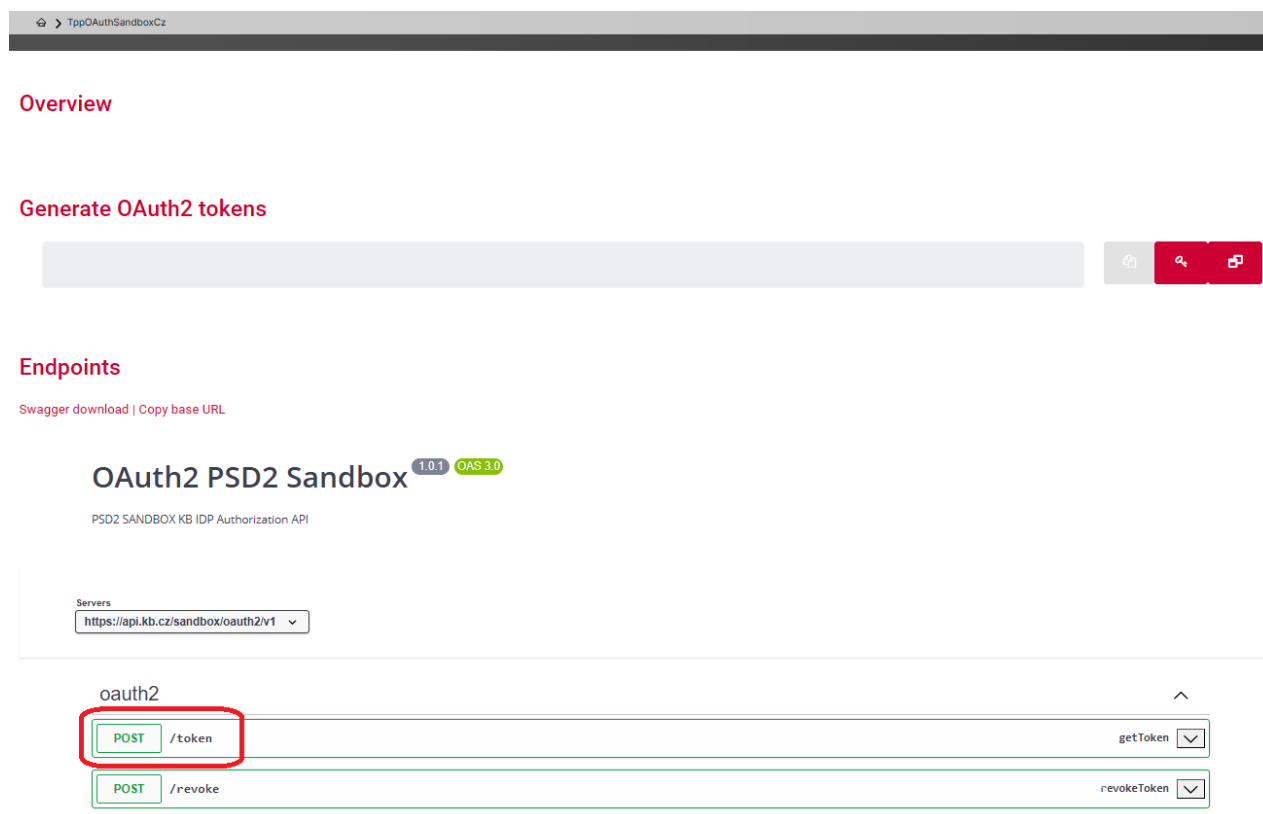
## 1.12 Vstoupení do API OAUTH2

Endpoints-seznam operací, které dané API umožňuje;



## 1.13 Vybrání operace „/token“ na otestování

Uživatel vybere operaci, kterou si přeje otestovat, v tomto případě vybere „/token“. Tato operace umožňuje zažádat a nechat si vygenerovat refreshovací nebo přístupový token.



## 1.14 Vyplnění požadovaných polí u operace „/token“

Uživatel, který si přeje nechat vygenerovat přístupový token, klikne na tlačítko „Try it out“ a vyplní všechna povinná pole hodnotami v příslušném formátu: Do pole s názvem „code“ zadá code, který získá v kroku 1.9. Do pole redirect\_uri zadá uživatel např. <https://www.kb.cz>, přičemž tato adresa musí být stejná jako u registrované aplikace. Do

pole s názvem „client\_id“ zadá consumer key, který si uložil v kroku 1.6. Do pole s názvem „client\_secret“ zadá consumer secret, které si uložil v kroku 1.6. Pokud vše provedl, jak měl, je mu po stisknutí tlačítka „Execute“, vygenerován token.

## Endpoints

[Swagger download](#) | [Copy base URL](#)

### OAuth2 PSD2 Sandbox 1.0.1 OAS 3.0

PSD2 SANDBOX KB IDP Authorization API

Servers  
<https://api.kb.cz/sandbox/oauth2/v1>

#### oauth2

**POST** /token getToken

After your application obtained an authorization code, it may exchange the authorization code for refresh and access token.

Parameters

No parameters

Request body **required** application/x-www-form-urlencoded

code string	The authorization code returned from the initial request.
refresh_token string	The refresh token string.
grant_type <b>required</b> string	Valid values: authorization_code.
redirect_uri string	The authorization code will be sent to this callback URL as a parameter. It must match one of the URLs registered during application registration. The value defaults to the first redirect URI configured for the client.

**Try it out**

#### oauth2

**POST** /token getToken

After your application obtained an authorization code, it may exchange the authorization code for refresh and access token.

Parameters

No parameters

Request body **required** application/x-www-form-urlencoded

code string	The authorization code returned from the initial request. code <input type="text"/> <input checked="" type="checkbox"/> Send empty value
refresh_token string	The refresh token string. refresh_token <input type="text"/> <input checked="" type="checkbox"/> Send empty value
grant_type <b>required</b> string	Valid values: authorization_code. authorization_code
redirect_uri string	The authorization code will be sent to this callback URL as a parameter. It must match one of the URLs registered during application registration. The value defaults to the first redirect URI configured for the client. redirect_uri <input type="text"/> <input checked="" type="checkbox"/> Send empty value
client_id string	The client ID obtained during application registration. client_id <input type="text"/> <input checked="" type="checkbox"/> Send empty value
client_secret string	The client application secret. client_secret <input type="text"/> <input checked="" type="checkbox"/> Send empty value

**Execute**

## 1.15 Vypsání chyby u operace „/token“

Pokud byla některá hodnota zadaná špatně, vypíše se po zmáčknutí tlačítka „Execute“ jedna z následujících chybových hlášek, jinak proběhne výpis výsledku.

400	<b>Invalid_request Invalid_scope</b> Media type: <input type="text" value="application/json"/> Example Value   Schema <pre>{   "errors": [     {       "error": "ERR_CODE_400"     }   ] }</pre>	No links
401	<b>Unauthorized_client Access_denied</b> Media type: <input type="text" value="application/json"/> Example Value   Schema <pre>{   "errors": [     {       "error": "ERR_CODE_401"     }   ] }</pre>	No links
403	<b>Forbidden_Insufficient_scope</b> Media type: <input type="text" value="application/json"/> Example Value   Schema <pre>{   "errors": [     {       "error": "ERR_CODE_403"     }   ] }</pre>	No links
404	<b>Not_Found</b> Media type: <input type="text" value="application/json"/> Example Value   Schema <pre>{   "errors": [     {       "error": "ERR_CODE_404"     }   ] }</pre>	No links
409	<b>Conflict</b> Media type: <input type="text" value="application/json"/> Example Value   Schema <pre>{   "errors": [     {       "error": "ERR_CODE_409"     }   ] }</pre>	No links
415	<b>Unsupported_Media_Type</b> Media type: <input type="text" value="application/json"/> Example Value   Schema <pre>{   "errors": [     {       "error": "ERR_CODE_415"     }   ] }</pre>	No links
422	<b>Unprocessable_Entity</b> Media type: <input type="text" value="application/json"/> Example Value   Schema <pre>{   "errors": [     {       "error": "ERR_CODE_422"     }   ] }</pre>	No links
500	<b>Server_Error</b> Media type: <input type="text" value="application/json"/> Example Value   Schema <pre>{   "errors": [     {       "error": "ERR_CODE_500"     }   ] }</pre>	No links

## 1.16 Vybrání operace „/revoke“ na otestování

Uživatel vybere operaci, kterou si přeje otestovat, v tomto případě vybere „/revoke“. Tato operace umožňuje zrušit platnost existujícího refreshovacího anebo přístupového tokenu.

↳ TppOAuthSandboxCz

**Overview**

**Generate OAuth2 tokens**

**Endpoints**

Swagger download | Copy base URL

**OAuth2 PSD2 Sandbox** <sup>1.0.1</sup> OAS 3.0

PSD2 SANDBOX KB IDP Authorization API

Servers

https://api-ref.kb.cz/sandbox/oauth2/v1

oauth2

POST	/token	getToken
POST	/revoke	revokeToken

## 1.17 Vyplnění požadovaných polí u operace „/revoke“

Uživatel, který si přeje zrušit už existující token, klikne na tlačítko „Try it out“ a vyplní všechna povinná pole hodnotami v příslušném formátu. Pokud vše provedl, jak měl, je daný token zrušen. V případě, že nejsou vyplněna všechna povinná pole, výpis se neprovede a nevyplněná pole jsou rudě zvýrazněna.

### Endpoints

[Swagger download](#) | [Copy base URL](#)

#### OAuth2 PSD2 Sandbox 1.0.1 OAS 3.0

PSD2 SANDBOX KB IDP Authorization API

Servers  
<https://api.kb.cz/sandbox/oauth2/v1>

oauth2

POST /token getToken

POST /revoke revokeToken

The API to revoke refresh token and/or access token.

Parameters Cancel Reset

No parameters

Request body required application/x-www-form-urlencoded

<b>token</b> <span>required</span>	The value of refresh or access token.
string	<input type="text" value="token"/>
<b>client_id</b> <span>required</span>	The client ID obtained during application registration.
string	<input type="text" value="client_id"/>
<b>client_secret</b> <span>required</span>	The client application secret.
string	<input type="text" value="client_secret"/>

Execute

## 1.18 Vypsání chyby u operace „/revoke“

Pokud byla některá hodnota zadána špatně, vypíše se po zmáčknutí tlačítka „TRY IT OUT“ jedna z následujících chybových hlášek, jinak proběhne výpis výsledku.

302	<p>Invalid_request Invalid_client Access_denied</p> <p>Media type</p> <p>application/json</p> <p>Example Value   Schema</p> <pre>{   "errors": [     {       "error": "ERR_CODE_400"     }   ] }</pre>	No links
400	<p>Invalid_request Invalid_scope</p> <p>Media type</p> <p>application/json</p> <p>Example Value   Schema</p> <pre>{   "errors": [     {       "error": "ERR_CODE_401"     }   ] }</pre>	No links
401	<p>Unauthorized_client Access_denied</p> <p>Media type</p> <p>application/json</p> <p>Example Value   Schema</p> <pre>{   "errors": [     {       "error": "ERR_CODE_403"     }   ] }</pre>	No links
403	<p>Forbidden_Insufficient_scope</p> <p>Media type</p> <p>application/json</p> <p>Example Value   Schema</p> <pre>{   "errors": [     {       "error": "ERR_CODE_404"     }   ] }</pre>	No links
404	<p>Not_Found</p> <p>Media type</p> <p>application/json</p> <p>Example Value   Schema</p> <pre>{   "errors": [     {       "error": "ERR_CODE_409"     }   ] }</pre>	No links
409	<p>Conflict</p> <p>Media type</p> <p>application/json</p> <p>Example Value   Schema</p> <pre>{   "errors": [     {       "error": "ERR_CODE_415"     }   ] }</pre>	No links
415	<p>Unsupported_Media_Type</p> <p>Media type</p> <p>application/json</p> <p>Example Value   Schema</p> <pre>{   "errors": [     {       "error": "ERR_CODE_422"     }   ] }</pre>	No links
422	<p>Unprocessable_Entity</p> <p>Media type</p> <p>application/json</p> <p>Example Value   Schema</p> <pre>{   "errors": [     {       "error": "ERR_CODE_500"     }   ] }</pre>	No links
500	<p>Server_Error</p> <p>Media type</p> <p>application/json</p> <p>Example Value   Schema</p> <pre>{   "errors": [     {       "error": "ERR_CODE_500"     }   ] }</pre>	No links



## 2 Postup přístupu do aplikace přes přímé volání

### 2.1 Charakteristika requestu Získání/vystavení tokenu

Poté co vaše aplikace obdrží autorizační kód, může ho následně vyměnit za access nebo refresh token.

**URI:** /token  
**HTTP Metoda:** POST  
**Request URL:** <https://api.kb.cz/sandbox/oauth2/v1/token>  
**Authorization:** request **vyžaduje** autorizaci uživatele/klienta jako součást volání API  
**Certification:** request **vyžaduje** použití kvalifikovaného certifikátu třetí strany.

**Podporované kódování:** charset=UTF-8

Parametry requestu:

Parametr	Hodnoty	Povinný	Popis
code	string	n (pokud se jedná o získání access tokenu pak je povinný)	Autorizační kód vrácený z původního requestu.
refresh_token	string	n (pokud se jedná o obnovení access tokenu pak je povinný)	Řetězec rešeršovacího tokenu..
grant_type	string	y	Validní hodnoty autorizačního kódu Povolené hodnoty : authorization_code, refresh_token
redirect_uri	string	n (pokud se jedná o získání access tokenu pak je povinný)	Autorizační kód bude zaslán na toto URL jako parametr. Musí se shodovat s jedním URL zaregistrovaným během registrace aplikace. Hodnota je defaultně nastavena na první URI, které bylo klientovi nakonfigurováno.
client_id		n (pokud se jedná o získání access tokenu pak je povinný)	Client_ID je získáno během registrace aplikace, ID aplikace TPP.
client_secret	string	n (pokud se jedná o získání access tokenu pak je povinný)	Client secret - password/token vydaný IDP banky/bance pro aplikaci (client_id) TPP

**Příklad requestu:**

```
POST /oauth2/token HTTP/1.1
Host: idp.banka.cz
Content-Type: application/x-www-form-urlencoded
```

```
code=a200234062baa2ada828bbd33c1f6054&
client_id=MyPFM&
client_secret={client_secret}&
redirect_uri=https://www.mypfm.cz/start&
grant_type=authorization_code
```

**Parametry response:**

Parametr	Hodnoty	Povinný	Popis
token_type	string	y	Typ zadaného tokenu. Hodnota nerozlišuje velká písmena. Typ tokenu například "Bearer"
access_token	string	y	Přístupový token vydaný autorizačním serverem.
refresh_token	string	n	Refreshovací tokeny jsou pověřeni užívaná k obstarání nových přístupových tokenů když už byly autorizovány.
expires_in	integer(\$int64)	y	Životnost přístupového tokenu, uvádí se v sekundách.
acr	integer(\$int64)	n	Úroveň zabezpečení autentizace. Nabývá hodnot 0 až 4. Default 3. Hodnota „0“ znamená nonSCA.

**Příklad response bez chyby:**

Úspěšně zpracovaný request odpoví response s takto definovaným JSON payloadem:

```
{
  "expires_in": 3600,
  "token_type": "Bearer",
  "access_token": "ae9eef9b0af42c674d0b1c1128c37c2d"
  "refresh_token": "be9eef9b0af42c674d0b1c1128c37c2g",
  "acr": "0"
}
```

**Chybové kódy:**

HTTP Status	Kód	Popis
400	invalid_request	Nevalidní request. V dotazu chybí povinné pole nebo je v nevhodném / nevalidním formátu.
401	Unauthorized_client Access_denied	Chybná autorizace na straně klienta, přístup odepřen
403	Forbidden	Klient není oprávněný provádět tento dotaz.
404	Not found	Zadaný dotaz se nepodařilo najít
429	Too many requests	Kapacita systému byla překročena zadáním přílišného množství requestů.
500	Internal server error	Chyba serveru

## 2.2 Charakteristika requestu Zneplatnění tokenu

API sloužící ke zrušení platnosti refresh nebo access tokenu.

**URI:** /revoke  
**HTTP Metoda:** POST  
**Request URL:** <https://api.kb.cz/sandbox/oauth2/v1/revoke>  
**Authorization:** request **vyžaduje** autorizaci uživatele/klienta jako součást volání API  
**Certification:** request **vyžaduje** použití kvalifikovaného certifikátu třetí strany.

**Podporované kódování:** charset=UTF-8

### Parametry requestu:

Parametr	Popis
token	OAuth2 access nebo refresh token získaný na základě autentizačního procesu po výměně za code resp. refresh token (v případě access_tokenu)

### Příklad requestu:

```
POST /oauth2/revoke HTTP/1.1
Host: idp.banka.cz
Content-Type: application/x-www-form-urlencoded

token=be9eef9b0af42c674d0b1c1128c37c2g
```

### Chybové kódy:

HTTP Status	Kód	Popis
302	Invalid_request Invalid_client Access_denied	Nevalidní request nebo nevalidní klient, přístup odepřen.
400	invalid_request	Nevalidní request. V dotazu chybí povinné pole nebo je v nevhodném / nevalidním formátu.
401	Invalid_client Invalid_grant Invalid_token	Nevalidní klient, nevalidní oprávnění nebo nevalidní token.
403	Forbidden	Klient není oprávněn provádět tento dotaz.
404	Not found	Zadaný dotaz se nepodařilo najít
429	Too many requests	Kapacita systému byla překročena zadáním přílišného množství requestů.
500	Internal server error	Chyba serveru

## 2.3 Charakteristika requestu na Autorizační resource

Když váš klient/aplikace není autorizován, musí si obstarat autorizační kód, předtím než si zažádá o přístupový token. Vaše aplikace může zahájit autorizační proces tím, že přesměruje webový prohlížeč svého uživatele na bankovní autorizační server. Server poté bude po uživateli požadovat jeho údaje. Uživateli se zobrazí povolení specifikovaná scopem a seznam bankovních a platebních služeb a účtů, ze kterých může uživatel vybírat. Pokud uživatel povolí vaši aplikaci přístup k něčemu z toho, server zašle autorizační kód na callback URL tím, že přesměruje prohlížeč na `redirect_uri`.

**URI:** /ssologin  
**HTTP Metoda:** GET  
**Request URL:** <https://api.kb.cz/sandbox/authfe/ssologin>  
**Authorization:** request **vyžaduje** autorizaci uživatele/klienta jako součást volání API  
**Certification:** request **vyžaduje** použití kvalifikovaného certifikátu třetí strany.

**Podporované kódování:** charset=UTF-8

**Parametry requestu:**

Parametr	Hodnoty	Povinný	Popis
<b>response_type</b>	code	y	Povinný parametr. Určuje použité autentizační flow. V tomto případě code grant. Pro autentizační proces to znamená, že jako výsledek úspěšné identifikace a autentizace je očekáván jednorázový code na místo access_tokenu.
<b>client_id</b>	ID aplikace TPP	y	Jedinečný identifikátor aplikace TPP vydaný bankou, resp IDP banky/bance. Např. použitím resource „0. Inicializační/registrační resource“
<b>redirect_uri</b>	URL	y	URL kam je na konci přesměrováno flow autentizace. Toto URL je stanoveno již při vydání client_id a v rámci autentizace je tento parametr validován proti URL zavedenému k client_id v systému IDP banky. Hodnota by se měla shodovat s jednou z hodnot zavedených použitím resource „0. Inicializační/registrační resource“.
<b>scope</b>	Seznam oprávnění oddělený mezerou	n	Jedná se o pole aplikací požadovaných scope (oprávnění). V případě PSD2 to mohou být role aisp a pisp. Např. pokud je TPP držitelem obou oprávnění může zde pro svoji aplikaci požádat jen o jedno z nich nebo oboje viz příklad requestu
<b>state</b>	Libovolný string	n	Tímto parametrem je možné obohatit redirect_uri při přesměrování. Slouží k předání informací z aplikace přes autentizační flow.

**Příklad requestu:**

```
GET /sandbox/authfe/ssologin HTTP/1.1
Host: api.kb.cz
Content-Type: application/x-www-form-urlencoded

client_id=MyPFM&
redirect_uri=https://www.mypfm.cz/start&
response_type=code&
scope=aisp pisp&
state=balance
```

**Parametry response:**

Pole	Popis
code	Autorizační code
state	Parametr state z requestu TPP

**Příklad response bez chyby:**

```
content-type: application/x-www-form-urlencoded
date: Wed, 8 Mar 2017 20:56:28 GMT
location: https://www.mypfm.cz/start?
        code=a200234062baa2ada828bbd33c1f6054&
        state=balance
status: 302
```

**Chybové kódy:**

HTTP Status	Kód	Popis
302	invalid_request	Nevalidní request. V dotazu chybí povinné pole nebo je v nevhodném / nevalidním formátu.
302	unauthorized_client	Klient není oprávněný provádět tento dotaz.
302	access_denied	Autorizační server odmítl přístup.
500, 503	server_error	Chyba autorizačního serveru.
302	invalid_scope	Neplatný scope požadavku.

**Příklad error response:**

```
HTTP/1.1 302 Found
Location: https://www.mymultibank.com/login?
        error=invalid_request
        &error_description=Unsupported%20response_uri
        &state=login_cz
```