API OAUTH2
Sandbox Manual

Change log

Date	Version	Description
10.02.2020	7	Fixed wrong password in Signing in
26.02.2020	8	Document and screen updates
22.05.2020	9	Document amendment
09.03.2021	10	Elimination of direct call variant - to register, change or delete application, incl. request for client secret via direct call – i.e. deleting sub-chapters from 2.1 to 2.5 compared to the previous document version No. 9 and renumbering the remaining sub-chapters in chapter 2.
29.05.2024	11	Addition of modifying the identity server link within 1 SCA flow on page 10.

Contents

Error reporting

Reporting quarantined errors or calling them always takes place via the mailbox api@kb.cz. The e-mail sent must contain the following information, in case the required information is missing, it will not be possible to process the query or error.

PSD2 API domain: CZ, SK
Environment: Sandbox, Production
Whether it was called from FE Sandbox incl. the type and version of the browser used or, in the case of a BE call, the name and version of the program for the BE call
Request type
Date and time of the call
IP address
The error and its most accurate description, which can be supplemented with the appropriate screenshot

Without the above values, it is not possible to solve the reported error.

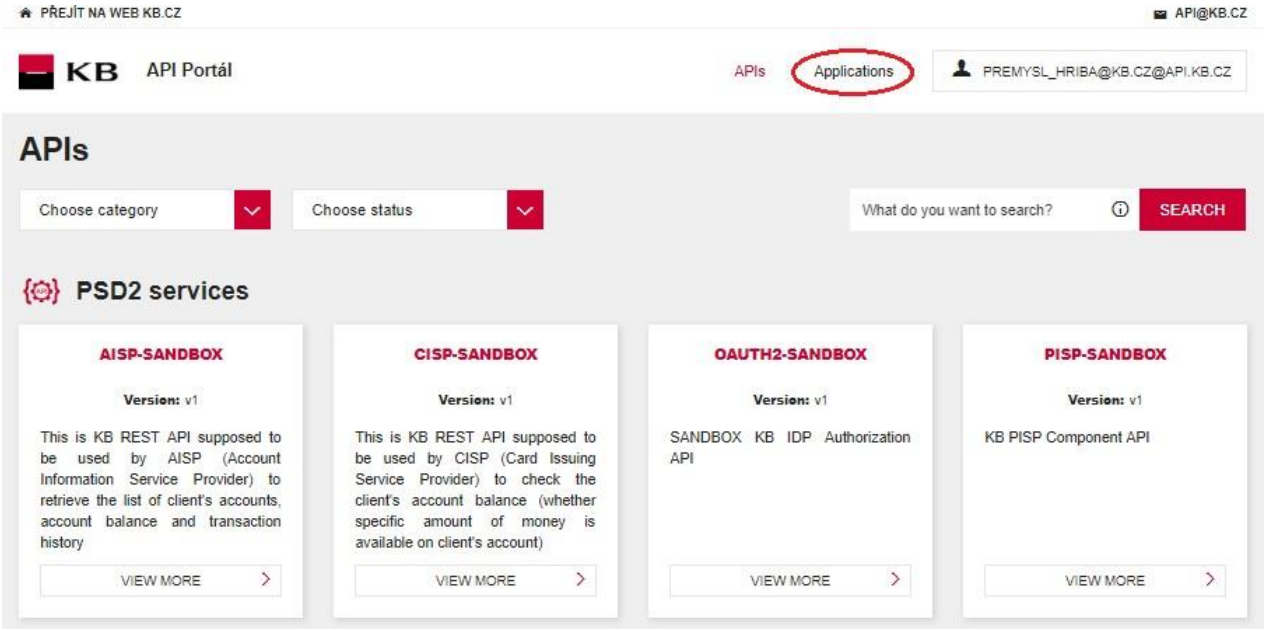
1 Procedure of Generating the authorization_code/refresh Token for the Application

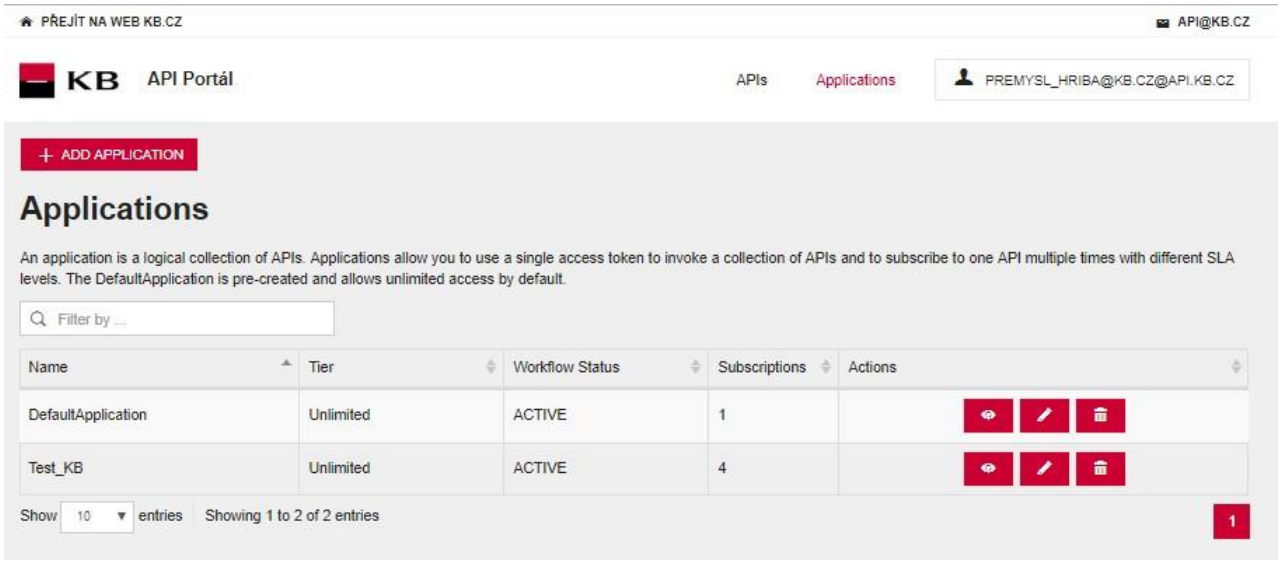
1.1 Prerequisites of access to applications

The user must be properly registered (see The Sandbox Manual on the Registering into the Sandbox).

1.2 Entering the application menu

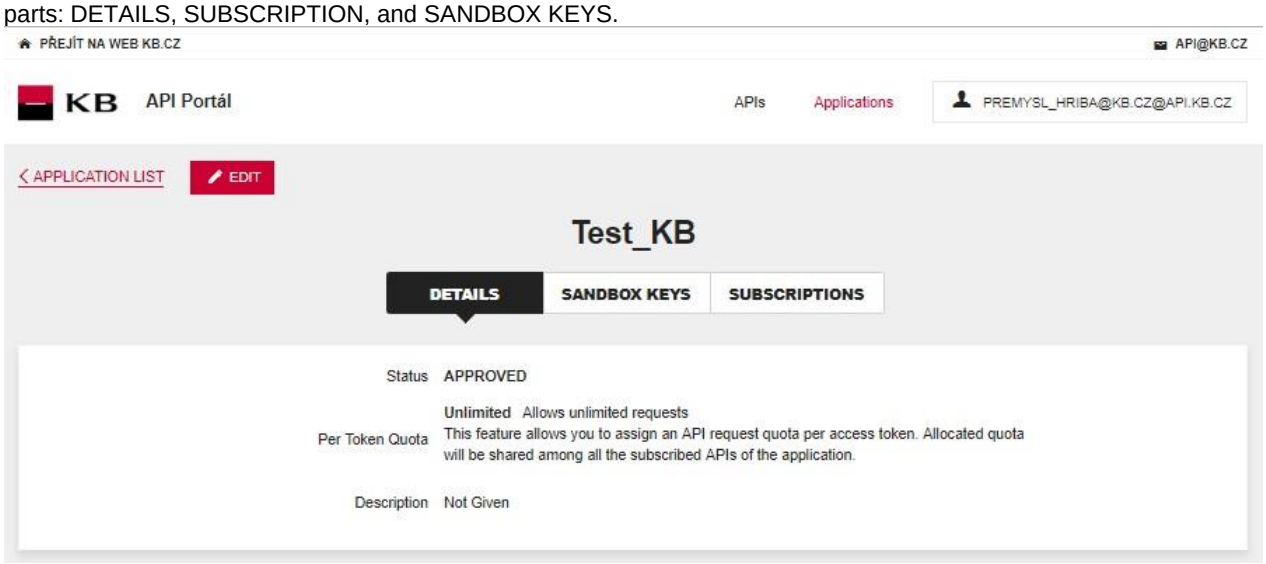
By clicking on the “Applications” button in the upper part of the screen, the signed-in user can enter the menu to register his/her application.





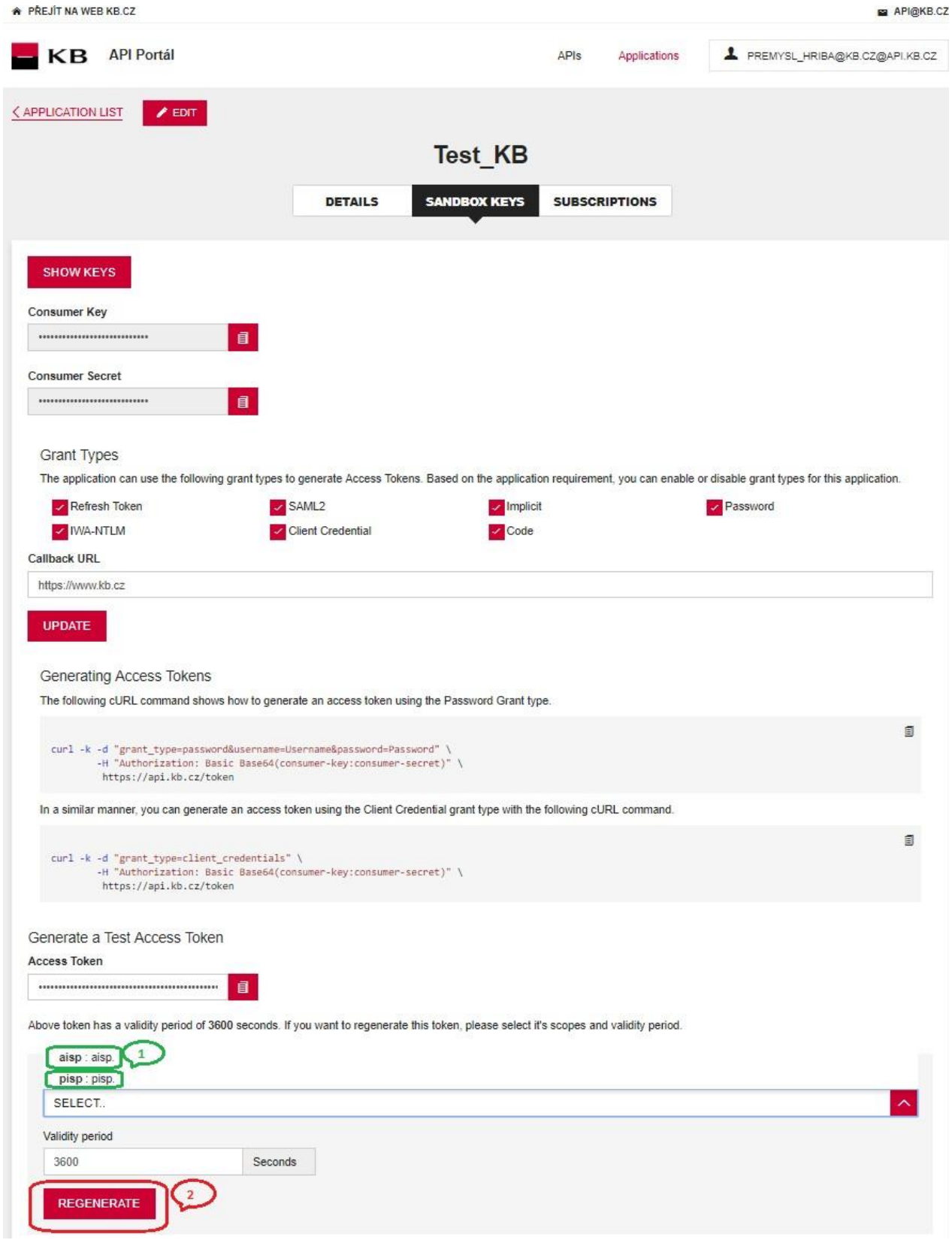
1.3 Viewing the application

The user clicks on the “View” option to display the preview of the selected operation. The function consists of 3 main parts: DETAILS, SUBSCRIPTION, and SANDBOX KEYS.



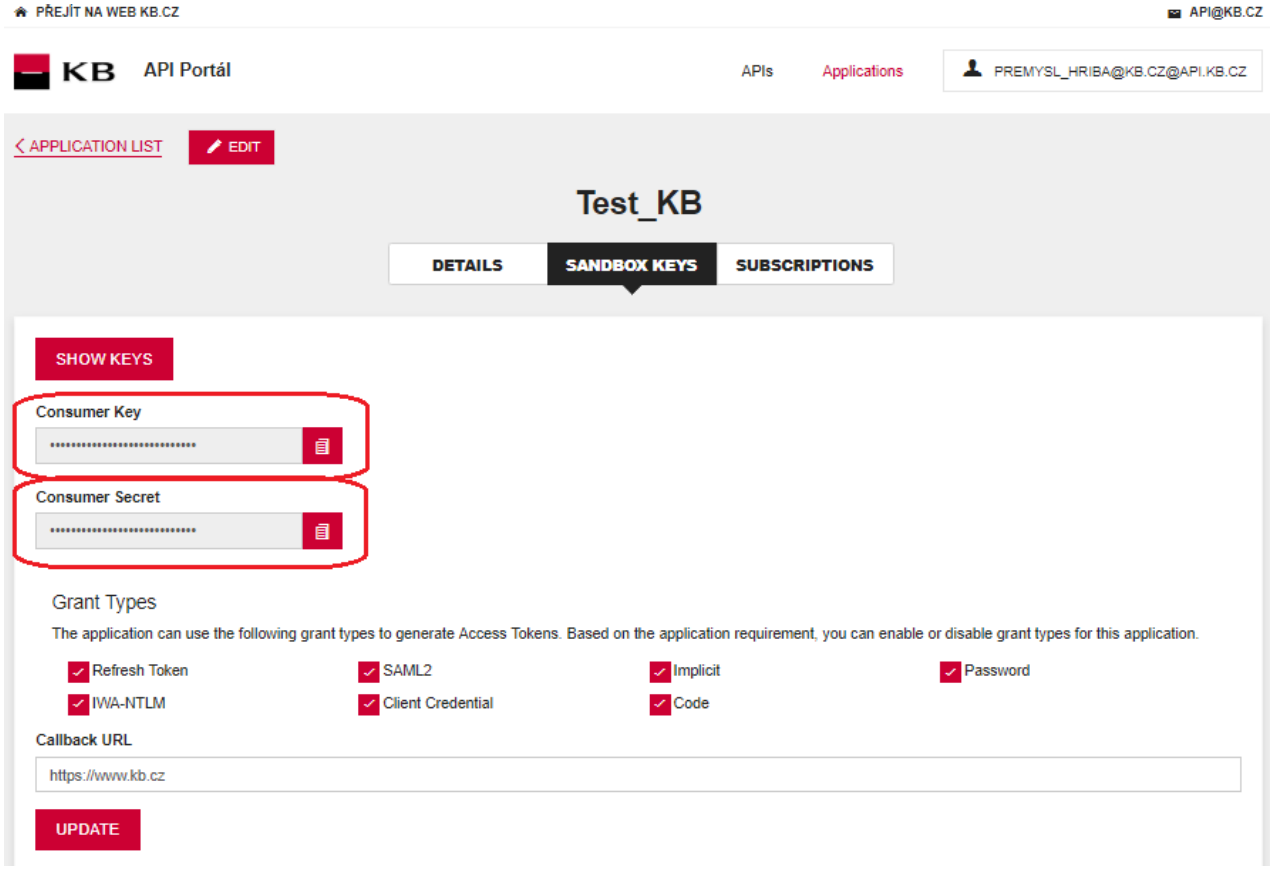
1.4 Sandbox keys

Subsequently, the user goes to the "Sandbox keys" section. If the conditions for generating the authorization_code/refresh token are met (see Section 1.**6**), the user can generate here a key/token securing access to a given scope and for the token as such (e.g. AISP, PISP, etc.) with properties set by the user here and with grant types selected by him/her.



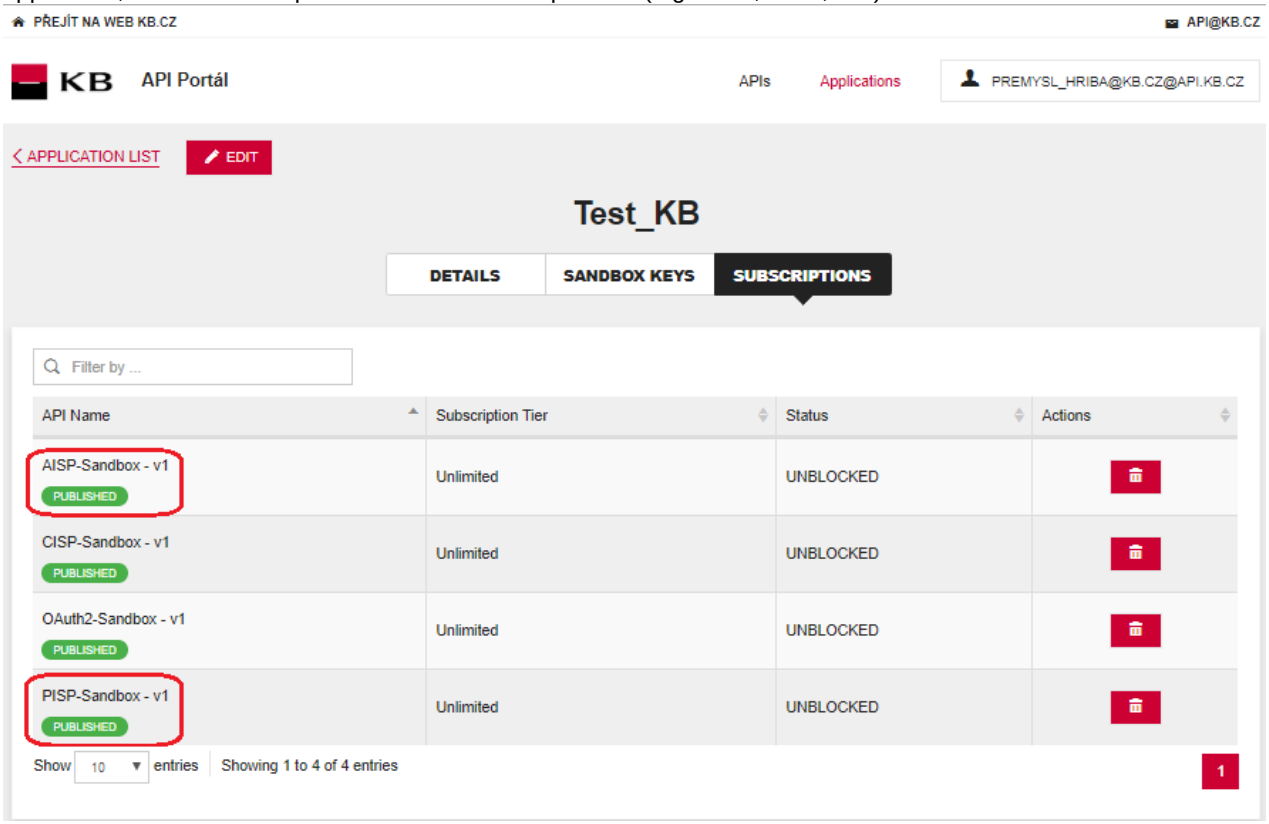
1.5 Storing the Consumer Key and Consumer Secret

The signed-in user can copy (for example to Notepad) the values from the “Consumer Key” and “Consumer Secret” fields (e.g. to the Notepad).



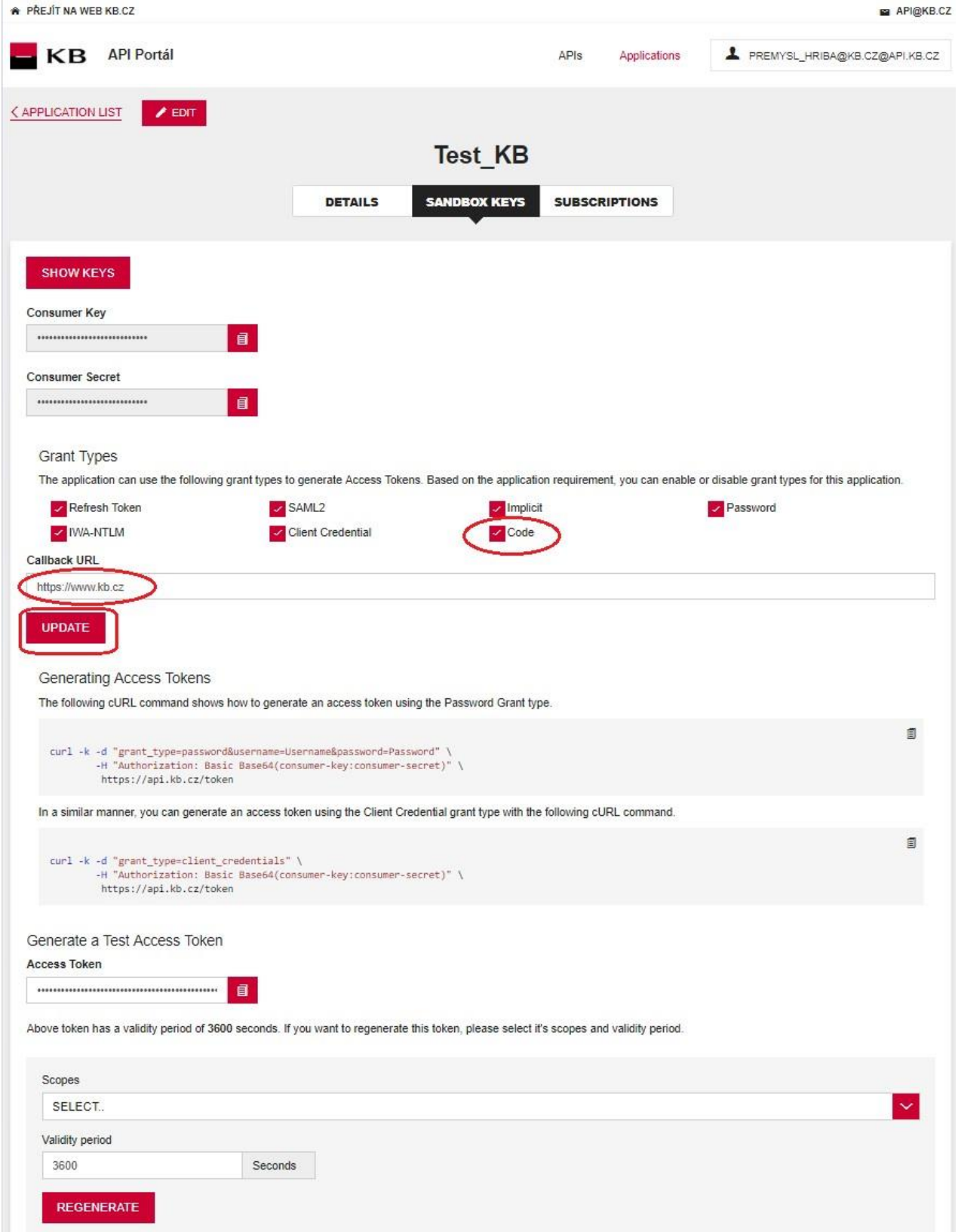
1.6 Preconditions for generating the authorization_code/refresh token

If the authorization_code/refresh token should be generated for a specific application, an API must be subscribed for this application, which makes it possible and uses this operation (e.g. AISP, PISP, etc.).



1.7 Entering the callback URL

The user enters the value https://www.kb.cz into the “Callback URL” text field in the “Grant Types” section and subsequently checks the “Code” checkbox. Then the user clicks on the “UPDATE” button.



1.8 Modifying the identity server link

Further, the user will change the string that follows client_id to the copied consumer key (the character “&” is not part of client_id) taken from the Sandbox Keys section of the given application. This is done in the following link:
https://api.kb.cz/sandbox/authfe?scope=aisp&redirect_uri=https://www.kb.cz&client_id=2dXmYO_yyYwLHW0yyaaCXTiAUy4a&state=123456&response_type=code

Modifying the identity server link within 1 SCA flow

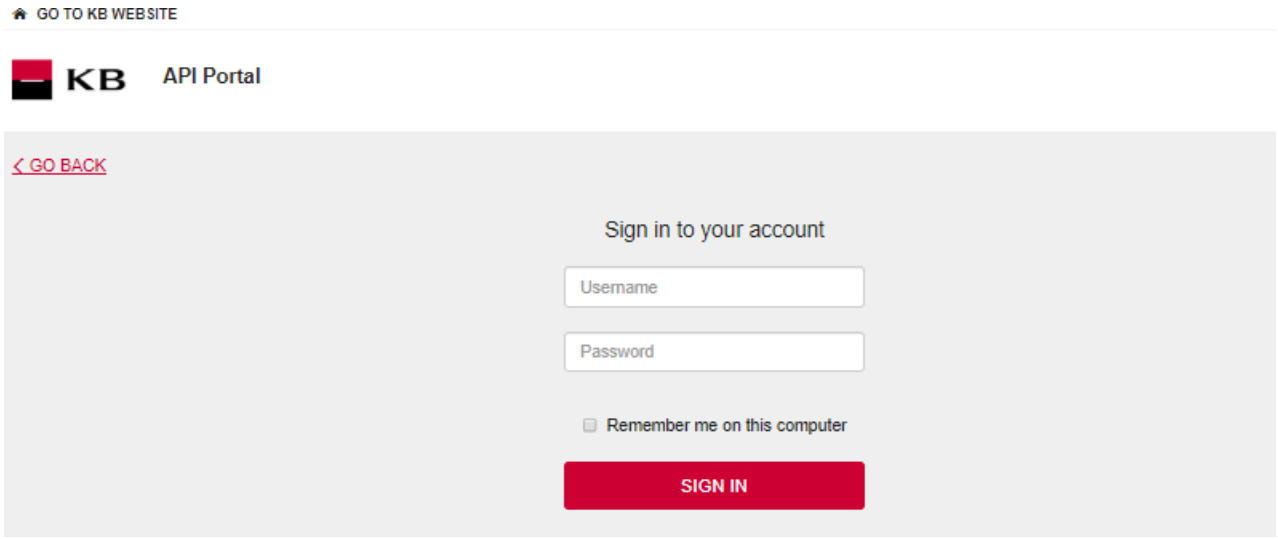
The user changes the client_id in the link below to the copied consumer key (the & character no longer belongs to the client_id) obtained in the Sandbox Keys section of the application and the request_uri obtained by calling the "Payment Authorization - starting the particular authorization method" operation (see the Sandbox Guide for the 1SCA Payment Initiation service). This is done in the following link:
https://api.kb.cz/sandbox/authfe?scope=pisp_1sca&redirect_uri=https://www.kb.cz&client_id=U5YePfGVtBBEWrTA6Nf5cp4Gf4Aa&response_type=code&request_uri=urn:uuid:94cc58a0-4ad9-4c77-95ca-b3832b3c72ff

1.9 Access to the identity server

The user opens an anonymous window in any browser and enters the address modified in the manner described in the foregoing paragraph.

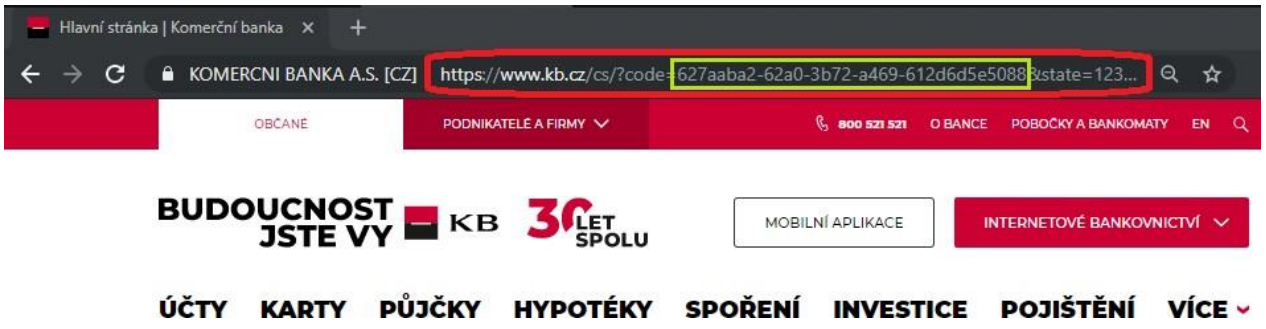
1.10 Signing in

When the foregoing step is completed, the sign-in screen is displayed. The User must enter the value “926637611” into the Username field and the password “sandbox” into the Password text field or use your credentials for Sandbox.



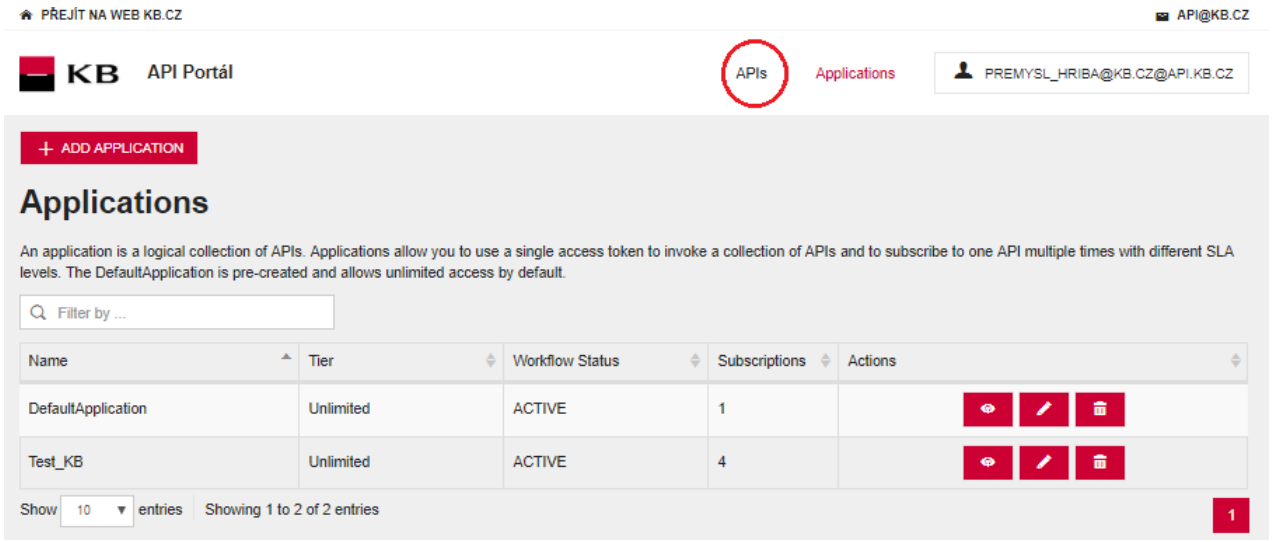
1.11 Obtaining the code

Now the user is redirected to the next page. The URL of this page contains the code value, which will later be used for generating the token.



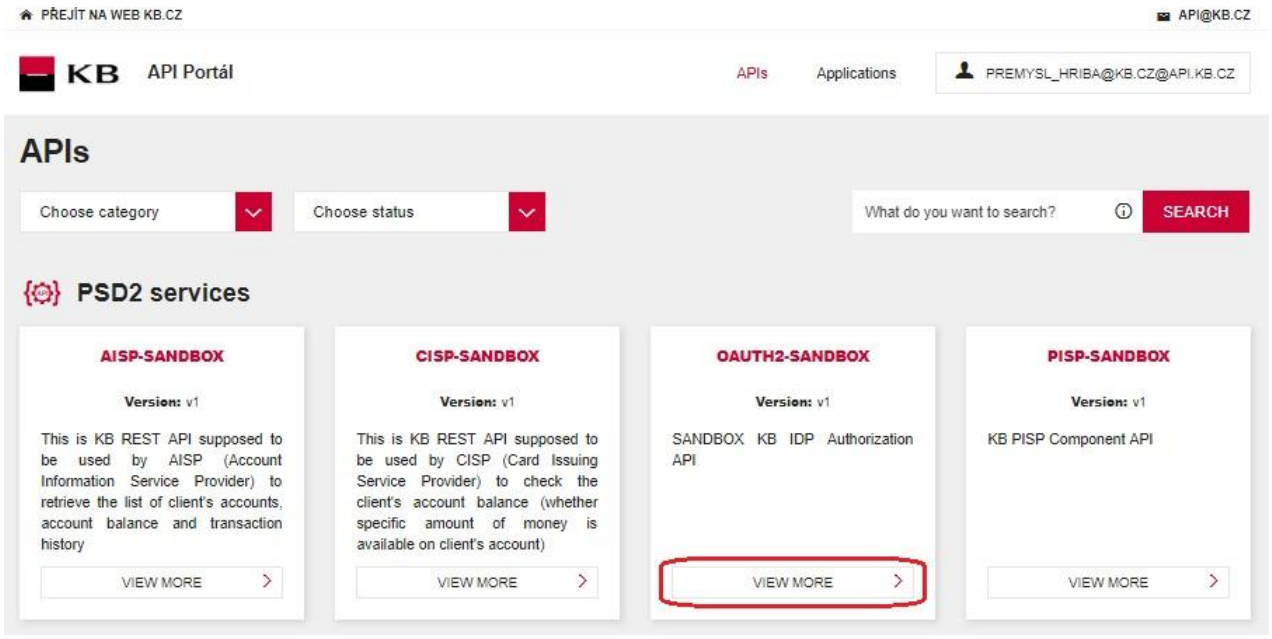
1.12 API menu

Subsequently, the user clicks on the “APIs” button in the upper part of the screen to enter the menu containing all APIs he/she is allowed to access.



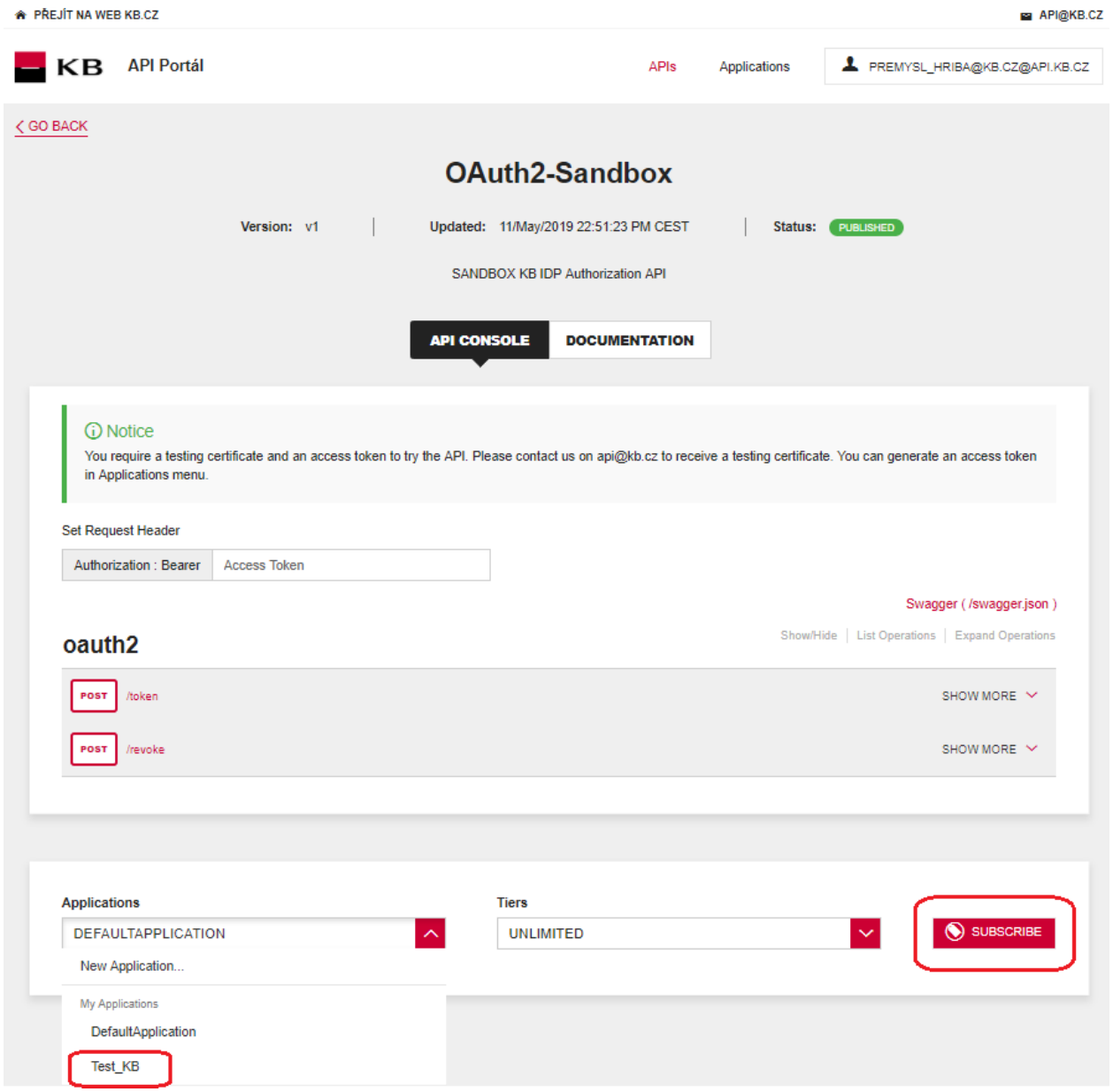
1.13 Selecting the API OAUTH2

The user can display the specific API by clicking on “VIEW MORE”.



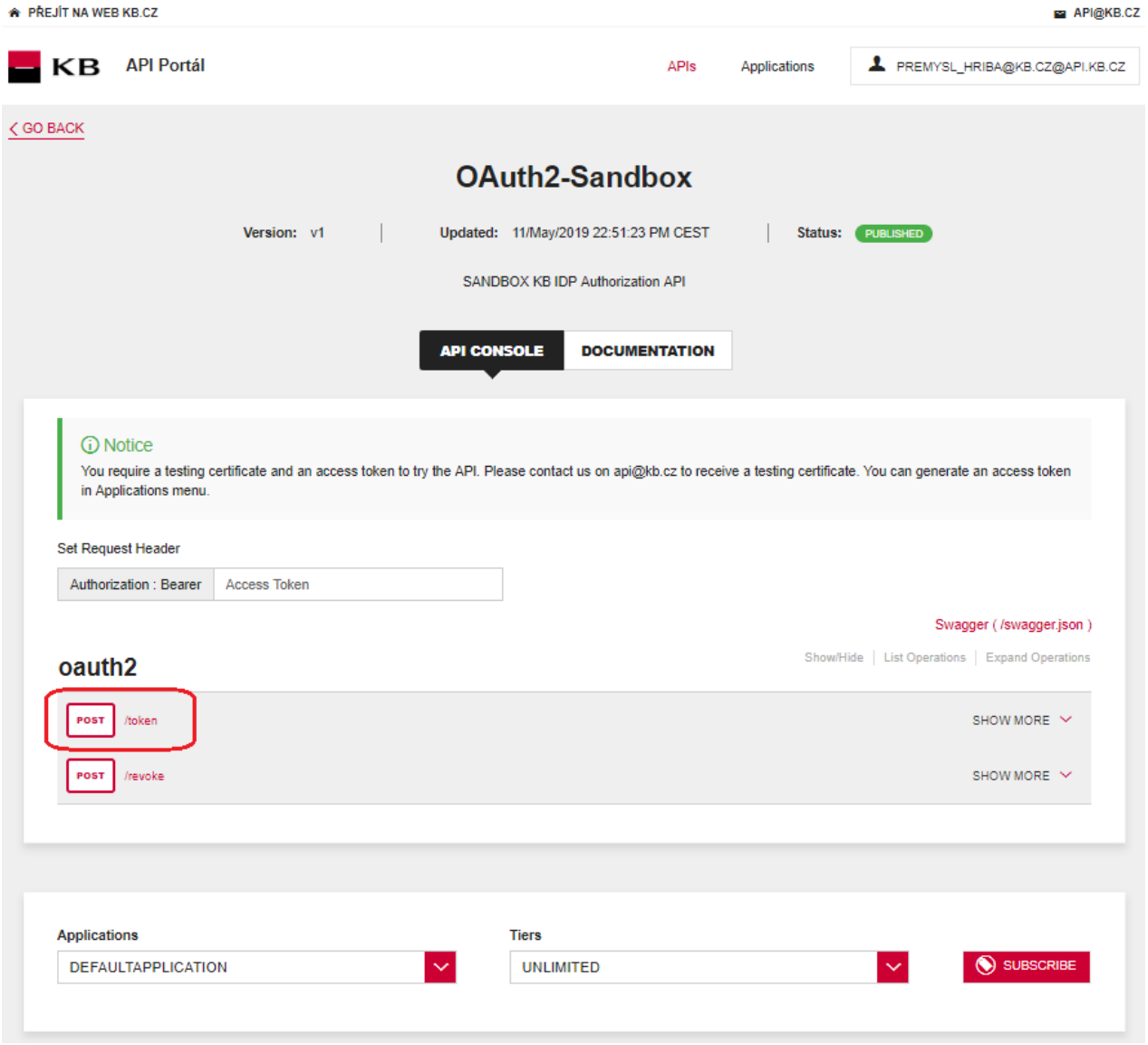
1.14 Entering the OAUTH2 API

API CONSOLE – a list of operations allowed by the specific API;
DOCUMENTATION – all available documentation concerning the specific application. The subscription of a selected API can be made here by clicking on the
“SUBSCRIBE” button, so that the given applications can use the API’s functions (as long as the user is properly signed in).



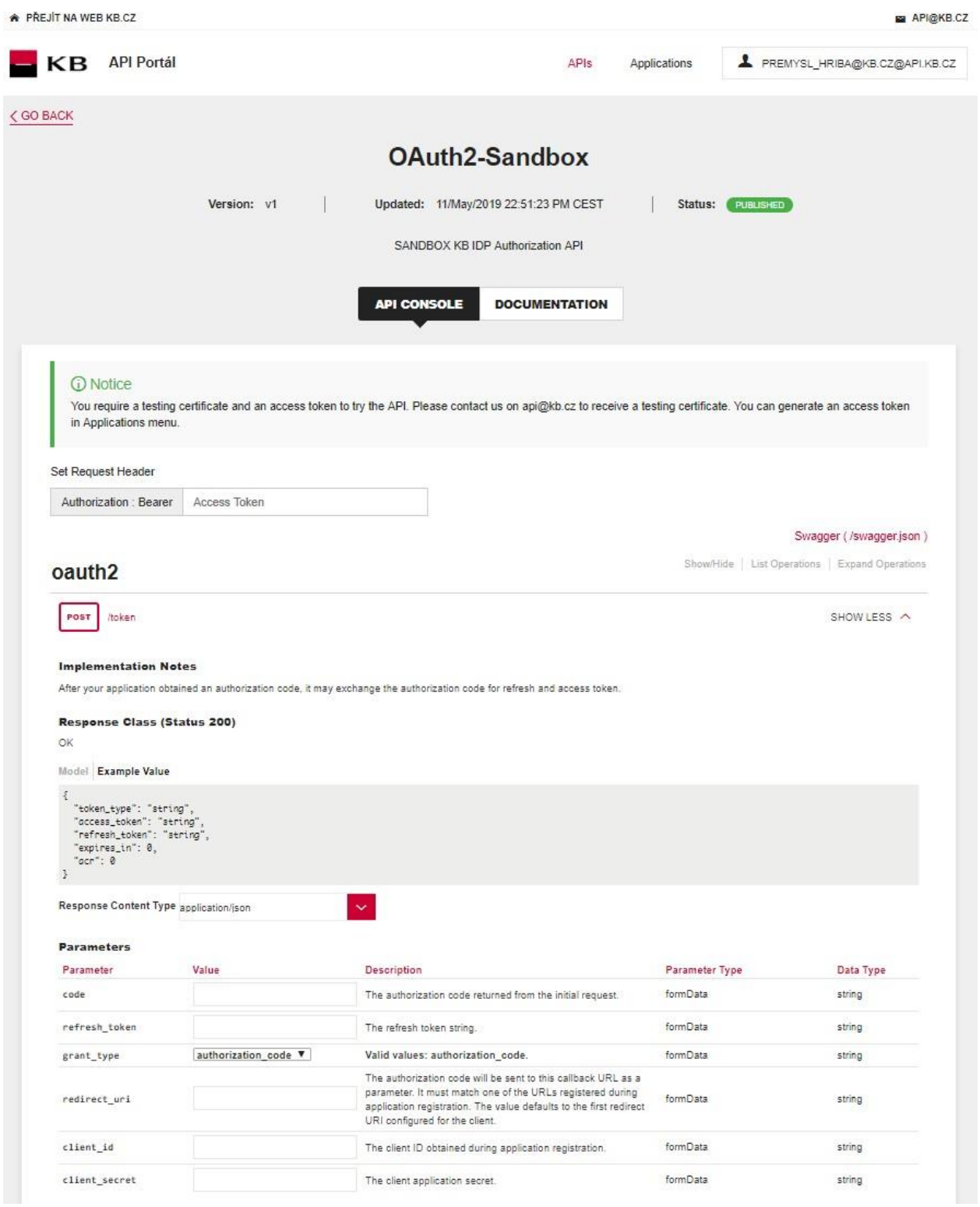
1.15 Selecting the “/token” operation

The user then goes to the “API CONSOLE” section and selects the “/token” operation to generate the access token or refresh token.



1.16 Filling in the required fields

The user wishing to get the access token generated fills in all mandatory fields with values in an appropriate format. The user will enter the code found in the URL in step 1.11 into the “code” field; the redirect_uri insert https://www.kb.cz ;the consumer key stored in step 0 to the “client_id” field; and the consumer secret stored in step 0 to the pole “client_secret” field. If everything is done properly, the specific token will be generated after pressing the "TRY IT OUT" button.


PŘEJÍT NA WEB KB.CZ
API@KB.CZ
KB API Portál
APIs
Applications
PREMYSL_HRIBA@KB.CZ@API.KB.CZ
GO BACK
OAuth2-Sandbox
Version: v1
Updated: 11/May/2019 22:51:23 PM CEST
Status: PUBLISHED
SANDBOX KB IDP Authorization API
API CONSOLE
DOCUMENTATION
Notice
You require a testing certificate and an access token to try the API. Please contact us on api@kb.cz to receive a testing certificate. You can generate an access token in Applications menu.
Set Request Header
Authorization : Bearer
Access Token
Swagger (/swagger.json)
oauth2
Show/Hide | List Operations | Expand Operations
POST
/token
SHOW LESS
Implementation Notes
After your application obtained an authorization code, it may exchange the authorization code for refresh and access token.
Response Class (Status 200)
OK
Model | Example Value
{
"token_type": "string",
"access_token": "string",
"refresh_token": "string",
"expires_in": 0,
"scr": 0
}
Response Content Type application/json
Parameters
Parameter
Value
Description
Parameter Type
Data Type
code
The authorization code returned from the initial request.
formData
string
refresh_token
The refresh token string.
formData
string
grant_type
authorization_code
Valid values: authorization_code.
formData
string
redirect_uri
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.
formData
string
client_id
The client ID obtained during application registration.
formData
string
client_secret
The client application secret.
formData
string

1.17 “/token” operation error message

If any value has been entered incorrectly, one of the following error messages will be displayed after pressing the "TRY IT OUT" button, otherwise the result statement will be displayed.

Response Messages

HTTP Status Code	Reason	Response Model	Headers
400	Invalid_request Invalid_scope	Model | Example Value	
401	Unauthorized_client Access_denied	Model | Example Value	
403	Forbidden_Insufficient_scope	Model | Example Value	
404	Not_Found	Model | Example Value	
409	Conflict	Model | Example Value	
415	Unsupported_Media_Type	Model | Example Value	
422	Unprocessable_Entity	Model | Example Value	
500	Server_Error	Model | Example Value	

```
{
  "errors": [
    {
      "error": "ERR_CODE_400"
    }
  ]
}
```

```
{
  "errors": [
    {
      "error": "ERR_CODE_401"
    }
  ]
}
```

```
{
  "errors": [
    {
      "error": "ERR_CODE_403"
    }
  ]
}
```

```
{
  "errors": [
    {
      "error": "ERR_CODE_404"
    }
  ]
}
```

```
{
  "errors": [
    {
      "error": "ERR_CODE_409"
    }
  ]
}
```

```
{
  "errors": [
    {
      "error": "ERR_CODE_415"
    }
  ]
}
```

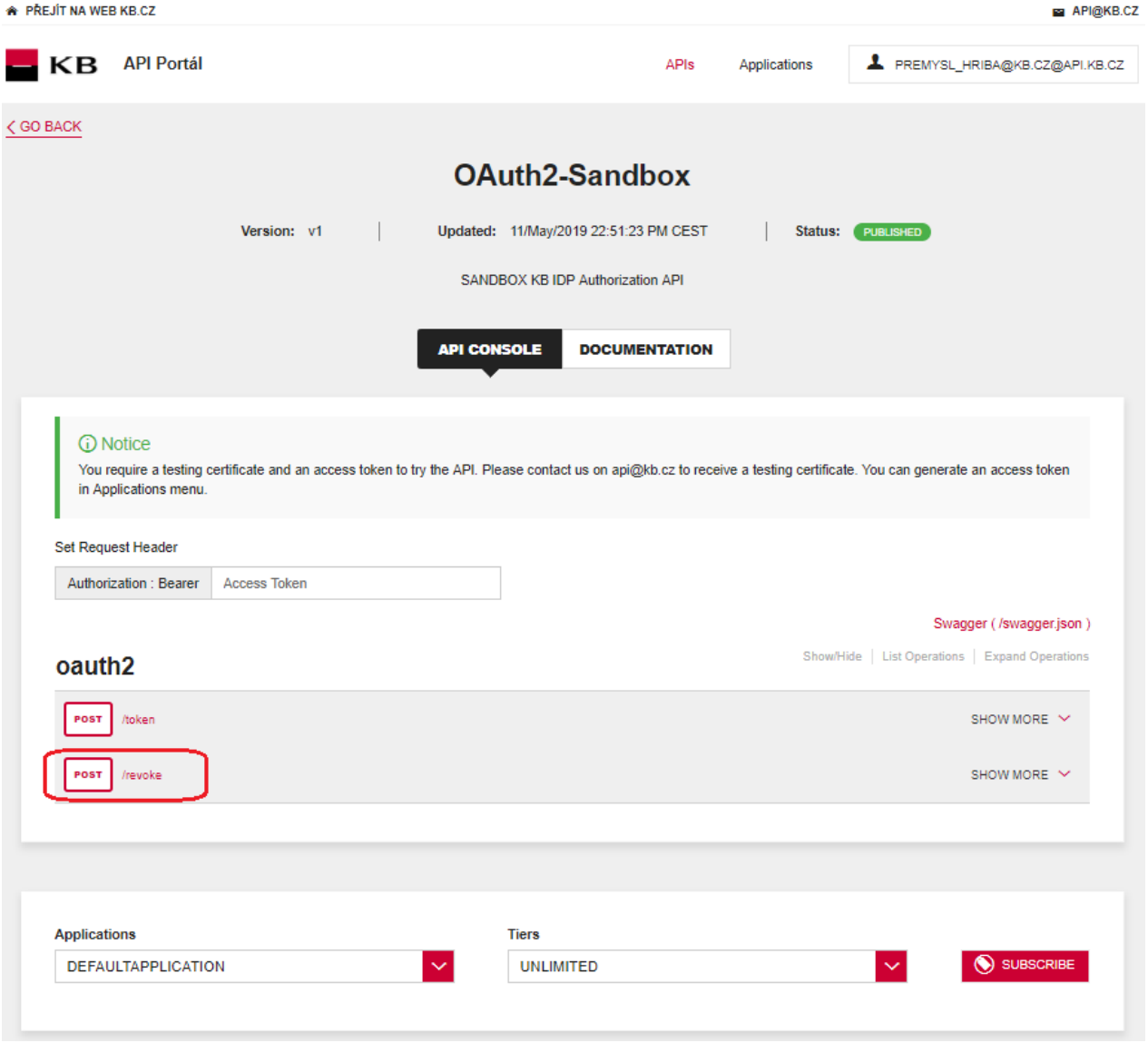
```
{
  "errors": [
    {
      "error": "ERR_CODE_422"
    }
  ]
}
```

```
{
  "errors": [
    {
      "error": "ERR_CODE_500"
    }
  ]
}
```

TRY IT OUT

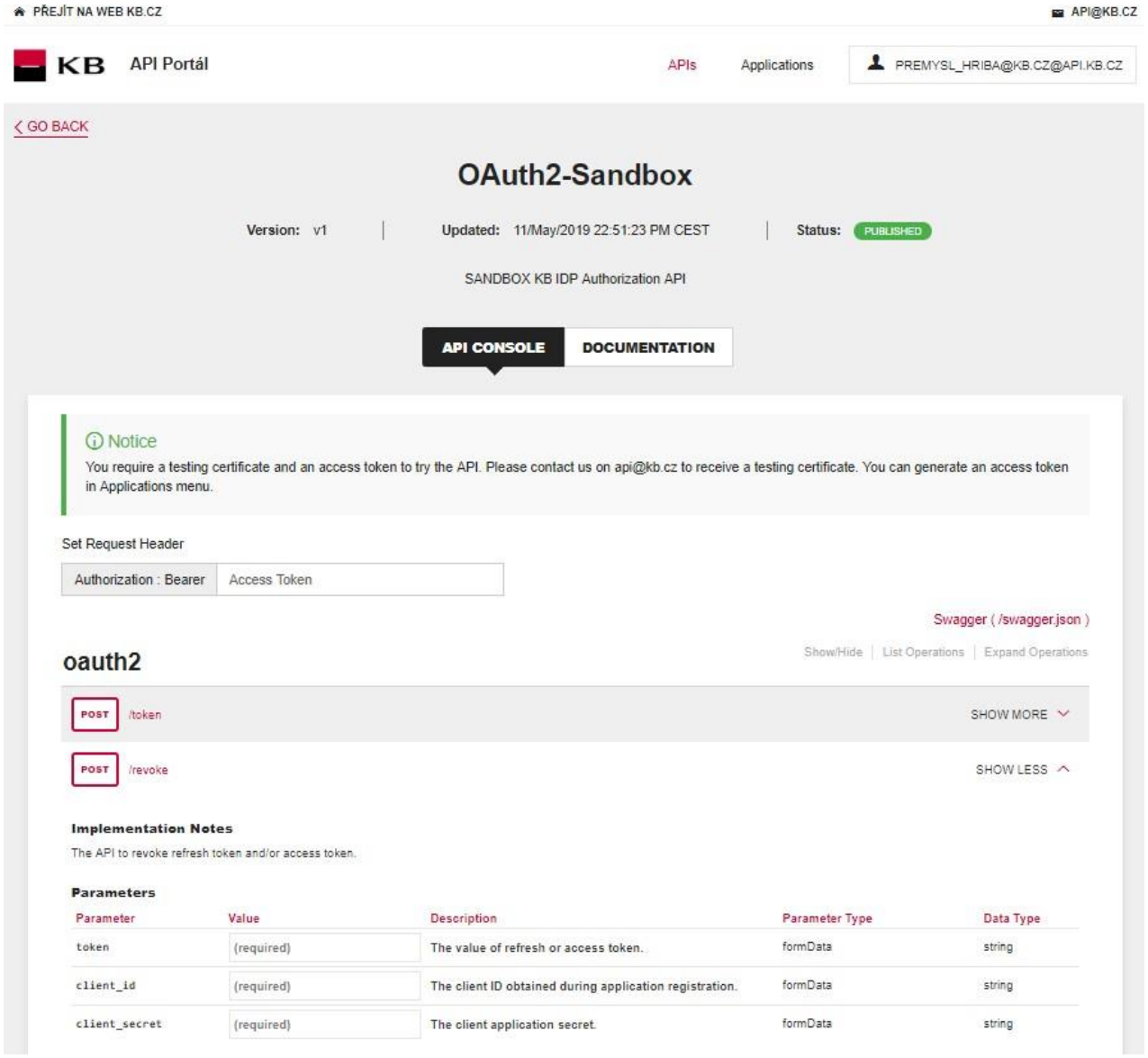
1.18 Selecting the “/revoke” operation for testing

The user chooses an operation he/she wishes to test. In this case, it is “/revoke”. The user can cancel the existing refresh token or access token using this operation.



1.19 Filling in the required fields of the “/revoke” operation

The user wishing to cancel an existing token fills in all mandatory fields with values in an appropriate format. If everything is done properly, the specific token will be cancelled. If any of mandatory fields is not filled in, the report is not displayed and the blank fields are highlighted in red.



1.20 “/revoke” operation error message

If any value has been entered incorrectly, one of the following error messages will be displayed after pressing the "TRY IT OUT" button, otherwise the result statement will be displayed.

Response Messages

HTTP Status Code	Reason	Response Model	Headers
204	OK		
302	Invalid_request Invalid_client Access_denied		
400	Invalid_request Invalid_scope	Model | Example Value { "errors": [{ "error": "ERR_CODE_400" }] }	
401	Unauthorized_client Access_denied	Model | Example Value { "errors": [{ "error": "ERR_CODE_401" }] }	
403	Forbidden_Insufficient_scope	Model | Example Value { "errors": [{ "error": "ERR_CODE_403" }] }	
404	Not_Found	Model | Example Value { "errors": [{ "error": "ERR_CODE_404" }] }	
409	Conflict	Model | Example Value { "errors": [{ "error": "ERR_CODE_409" }] }	
415	Unsupported_Media_Type	Model | Example Value { "errors": [{ "error": "ERR_CODE_415" }] }	
422	Unprocessable_Entity	Model | Example Value { "errors": [{ "error": "ERR_CODE_422" }] }	
500	Server_Error	Model | Example Value { "errors": [{ "error": "ERR_CODE_500" }] }	

TRY IT OUT

2 Access to the application through direct calling

2.1 Obtaining/Issuing the Token – Request Characteristics

Having received the authorisation code, your application may subsequently swap it for an access token or refresh token.

URI: /token
HTTP Method: POST
Request URL: https://api.kb.cz/sandbox/oauth2/v1/token
Authorization: the request **requires** the user/client authorisation as part of the API calling
Certification: the request **requires** the use of the third party qualified certificate.

Supported encoding: `charset=UTF-8`

Request parameters:

Parameter	Values	Mandatory	Description
code	string	n (mandatory in the case of obtaining the access token)	An authorisation code returned from the original request.
refresh_token	string	n (mandatory in the case of refreshing the access token)	A refresh token string.
grant_type	string	y	Valid values of the authorisation code. Permitted values of authorization_code, refresh_token.
redirect_uri	string	n (mandatory in the case of obtaining the access token)	The authorisation code will be sent to this URL as a parameter. It should be identical to one URL registered during the application registering. By default, the value is set to the first URl that has been configured for the client.
client_id		n (mandatory in the case of obtaining the access token)	The Client_ID is obtained while the application is being registered, TPP application ID.
client_secret	string	n (mandatory in the case of obtaining the access token)	Client secret – a password/token issued by the bank IDP for the (`client_id`) TPP application.

Example of a request:

```
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
```

Response parameters:

Parameter	Values	Mandatory	Description
token_type	string	y	The inputted token type. The value does not distinguish between capital and lower-case letters. An example of the token type: “Bearer”
access_token	string	y	An access token issued by the authorising server.
refresh_token	string	n	Refresh tokens are authorisations used for obtaining new access tokens after they have been authorised.
expires_in	integer($int64)	y	A life time of the access token expressed in seconds.
acr	integer($int64)	n	The authentication security level. The value van range from 0 to 4, the default value is 3. “0” means nonSCA.

Example of an error-free response:
A successfully processed request generates a response with the JSON payload defined as follows:

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

Error codes:

HTTP Status	Code	Description
400	**invalid_request**	Invalid request. It is missing a mandatory field or its format is inappropriate / invalid.
401	**Unauthorized_client Access_denied**	Erroneous client-side authorisation, access denied.
403	**Forbidden**	The client is not authorised to execute this query.
404	**Not found**	The entered query has not been found.
429	**Too many requests**	The system capacity has been exceeded by inputting too many requests.
500	**Internal server error**	Server error.

2.2 Invalidating the Token – Request Characteristics

The API invalidating the refresh token or access token.

URI: /revoke
HTTP Method: POST
Request URL: https://api.kb.cz/sandbox/oauth2/v1/revoke
Authorization: the request **requires** the user/client authorisation as part of the API calling
Certification: the request **requires** the use of the third party qualified certificate.

Supported encoding: `charset=UTF-8`

Request parameters:

Parameter	Description
token	OAuth2 access or refresh token obtained during the authentication process after its exchange (swap) for the code or refresh token (in the case of the access_token).

Example of a request:

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

token=be9eef9b0af42c674d0b1c1128c37c2g
```

Error codes:

HTTP Status	Code	Description
302	**Invalid_request** **Invalid_client** **Access_denied**	Invalid request or invalid client; access denied.
400	**invalid_request**	Invalid request. It is missing a mandatory field or its format is inappropriate / invalid.
401	**Invalid_client** **Invalid_grant** **Invalid_token**	Invalid client, invalid grant, or invalid token.
403	**Forbidden**	The client is not authorised to execute this query.
404	**Not found**	The entered query has not been found.
429	**Too many requests**	The system capacity has been exceeded by inputting too many requests.
500	**Internal server error**	Server error.

2.3 Authorising Resource – Request Characteristics

If your client/application has not been authorised, it must obtain an authorising code before applying for an access token. Your application may launch the authorising process by redirecting its user's web browser to the bank authorisation server. The server will then require user's data from the user. Authorisations specified by the scope and a list of bank services and payment services from which to choose will be displayed to the user. If the user makes it possible for your application to access any of them, the server will send an authorising code to the callback URL by redirecting the browser to redirect_uri.

URI: /ssologin
HTTP Method: GET
Request URL: https://api.kb.cz/sandbox/oauth2/autfe/ssologin
Authorization: the request **requires** the user/client authorisation as part of the API calling
Certification: the request **does not require** the use of the third party qualified certificate.

Supported encoding: `charset=UTF-8`

Request parameters:

Parameter	Values	Mandatory	Description
response_type	code	y	A mandatory parameter determining the authentication flow that has been used (code grant in this case). In terms of the authentication process it means that a one-time code is expected instead of the access_token as a result of a successful identification and authentication.
client_id	ID of TPP application	y	A unique identifier of the TPP application issued by the bank, or the bank IDP, e.g., by using the "*0. Initializing/registering resource"*.
redirect_uri	URL	y	An URL to which the authentication flow is redirected in the end. This URL is already determined while the client_id is issued, and this parameter is validated as part of the authentication against the URL introduced for the client_id on the bank's IDP system. The value should be identical to one of the values introduced by using the "*0. Initializing/registering resource".*
scope	List of authoris-ations separated by a space	n	A field of scopes (authorisations) required by the application. For PSD2, it may be the aisp and pisp roles. E.g., if the TPP is a holder of both authorisations, it may require here either one or both for its application (see the example of the request).
state	Arbitrary string	n	Redirect_uri can be supplemented with this parameter when redirected. It conveys information from the application via the authentication flow.

Example of a request:

```
GET /oauth2/authfe/ssologin HTTP/1.1
Host: idp.banka.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
```

Response parameters:

Pole	Description
code	Authorisation code
state	A state parameter from the TPP request.

Example of an error-free response:

```
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
```

Error codes:

HTTP Status	Code	Description
302	**invalid_request**	Invalid request. It is missing a mandatory field or its format is inappropriate / invalid.
302	**unauthorized_client**	The client is not authorised to execute this query.
302	**access_denied**	Access denied by the authorising server.
500, 503	**server_error**	Authorising server error.
302	**invalid_scope**	Invalid request scope.

Example of an error response:

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