BOC API Documentation

Table of Contents

Contents

How to register to API Store?

Introduction

BoC API Store offers a collection of APIs through which BoC opens the gateway to core banking functionality.

Through our PSD2 APIs and with the approval of the customer, third party providers (TPP) can access account information and initiate payments.

Using our B2B APIs, businesses can connect their ERP Systems to perform group/payroll and multiple payments.

For more information, please visit the API section of the portal.

Registration process

As a pre-requisite, to use any of our APIs you must first make sure that you are registered and subscribed to the required ones.

If you are a PSD2 TPP authorized by any European competent authority, you are required to have **digital eIDAS certificate (QSeal, QWAC)** and use it as transport certificate when calling the TPP Onboarding API for registration. The registration API needs the JSON body in JWS format sign with your **digital eIDAS QSeal certificate**.

Example Request:

```
{
  "payload":
"eyJkZXZlbG9wZXIiOnsiZmlyc3ROYW1lIjoiU2F2dmFzIiwibGFzdE5hbWUiOiJNaWNoYWVsIiwiZ
W1haWwiOiJlYXJ0aGZsYXgtZGlhc3BpZGlkYWUtdW5tb2RpZmllZEBsYW5kbWFyay5pbyIsInBob2
5lIjoiOTkwNTg2NzYifSwiY29udGFjdCI6eyJmaXJzdE5hbWUiOiJDaHJpc3RvcyIsImxhc3ROYW1lIjo
iTmljb2xhb3UiLCJlbWFpbCI6ImNuaWNvbGFvdTFAZ21haWwuY29tIiwicGhvbmUiOiI5OTY5NTE
5MSJ9LCJzaWduaW5nQ2VydGlmaWNhdGUiOiJNSUlIdmpDQ0JhYWdBd0lCQWdJUUNwZlVnSmM
vSWtMdTJpNngwTmQxYS4uLi4ifQ",
"signatures": [
    {
      "protected": "eyJhbGciOiJIUzI1NiJ9",
      "signature": "XkU8RonWMjBsrDrA9geT_UmpskDmolNx2wMKUP43Heg"
    }
  ]
```

On the other hand, if you are a B2B customer, you can register through your banker using this form. B2B providers do not need to provide a transport certificate but in case they will be requiring payments, then they will need Signing Certificate. In this case, during their registration they will be provided with instructions on how such certificate can be issued.

Note that for testing APIs on Sandbox, you just need to use the Sign-On functionality on the portal to create a developer account.

In case of a successful registration, the developer included in the API will receive appropriate credentials via email to get access to the developer portal. Through the Developer portal, a developer can create one or more applications and subscribe to APIs to be able to use them. For more information on this, please refer to How to access and call our APIs page.

How to access and call our APIs?

To access and call any of our APIs, you need to first *get **an access token*** that would be used in each subsequent call to authenticate your app to BoC resources, but also a ***subscription id (only for Accounts and B2B API families)*** that will define the access rights users have for a particular resource for prescribed amount of time.

Get Access Token

To retrieve an Access Token, token API needs to be called using Client ID and Client Secret, granted on the registration process. The Access token has a short lifetime of a few minutes.

Create Subscription

Before calling any of the APIs in the Accounts and B2B Payments API families, you must ensure that you follow the ‘Create Subscription’ process. The process is essentially the BoC implementation of an OAuth 2.0 Authorization workflow which will result in the acquisition of a Subscription ID.

BoC follows OAuth 2.0 which is the industry standard for delegating authorization for accessing resources via HTTP. This enables giving access rights to services and accounts to an app without explicitly providing a password. Instead, a Subscription ID is handed to the app/service which represents the access rights for a particular resource for a prescribed amount of time.

Within a banking context, this means that users have the granularity of choice in granting access to specific accounts for specific functions.

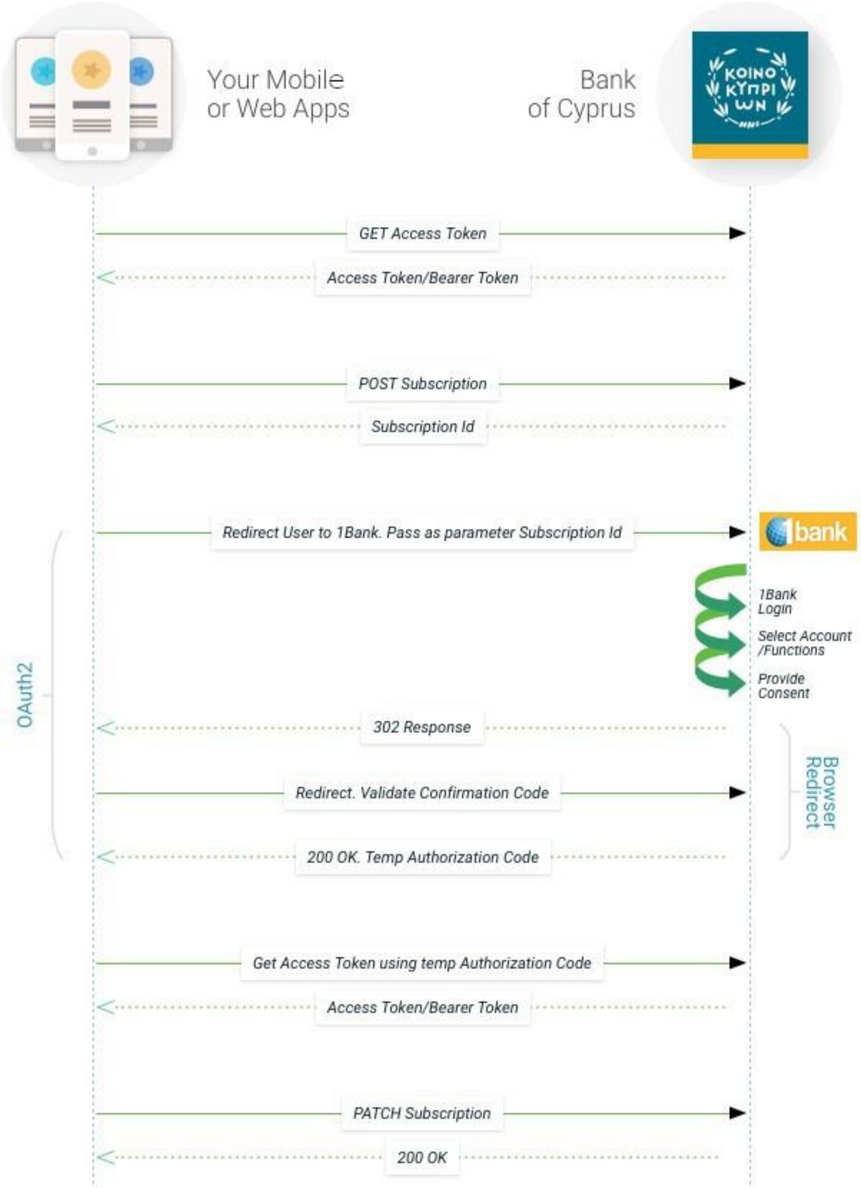
The client app should first call the POST Subscription API which will return the Subscription ID. Once we have the Subscription ID the client app re-directs the user to BoC Web banking login screen.

The user will be requested to login to 1Bank by supplying their UserID and Passcode. Following login, authorization will be requested for the Accounts and the appropriate functionality that will be provided to the subscription. As part of this Authorization workflow the client app would be provided with a specific temporary Authorization Code which will be used to get a second access token needed for the PATCH Subscription API to activate the subscription. The

PATCH Subscription API expects also the approved subscription details in the payload which the client app can retrieve using the Subscription API before calling the PATCH API.

The 'Create Subscription' process is described in the following sequence diagram:

Subscription Flow Sequence Diagram


Your Mobile or Web Apps
Bank of Cyprus
ΚΟΙΝΟ ΚΥΠΡΙ ΩΝ
GET Access Token
Access Token/Bearer Token
POST Subscription
Subscription Id
Redirect User to 1Bank. Pass as parameter Subscription Id
1bank
1Bank Login
Select Account /Functions
Provide Consent
OAuth2
302 Response
Redirect. Validate Confirmation Code
200 OK. Temp Authorization Code
Browser Redirect
Get Access Token using temp Authorization Code
Access Token/Bearer Token
PATCH Subscription
200 OK

NOTE: Subscription ID is valid for 180 days. For API calls you first need to get an Access Token (valid for 60mins), and along with each call you must pass the Token and the SubscriptionId. In case the subscription expires then re-send the subscription request with the same details i.e. functions and accounts to again get the consent of the customer.

Call an API

To call any API, an Access Token is required as indicated in section **Get access Token**. When calling APIs from the Accounts & B2B families you also need to have in hand a particular Subscription Id which is used as part of the API call. The SubscriptionId can be obtained as outlined in section **Create Subscription**.

You will then need to log onto the portal, **register your application** and subscribe to the required APIs you would like to use.

Account API

Follow these steps to test Accounts APIs.

Payments API

The payment APIs require an OAuth2 flow for authorizing a payment every time you initiate a payment as per the PSD2 Strong Customer Authorization (SCA) guidelines. The flow is similar to the subscription one above.

The client app should first call the initiate payment API with the details of the payment including the amount and the beneficiary. The ordering account is optional and in case it is not provided the customer will be asked to select it when redirected to BoC. All kinds of transfers are supported by this set of APIs including transfers within BoC, SEPA and SWIFT. Once the payment is initiated a Payment Id is returned which is passed as a query parameter when redirecting the user to BoC.

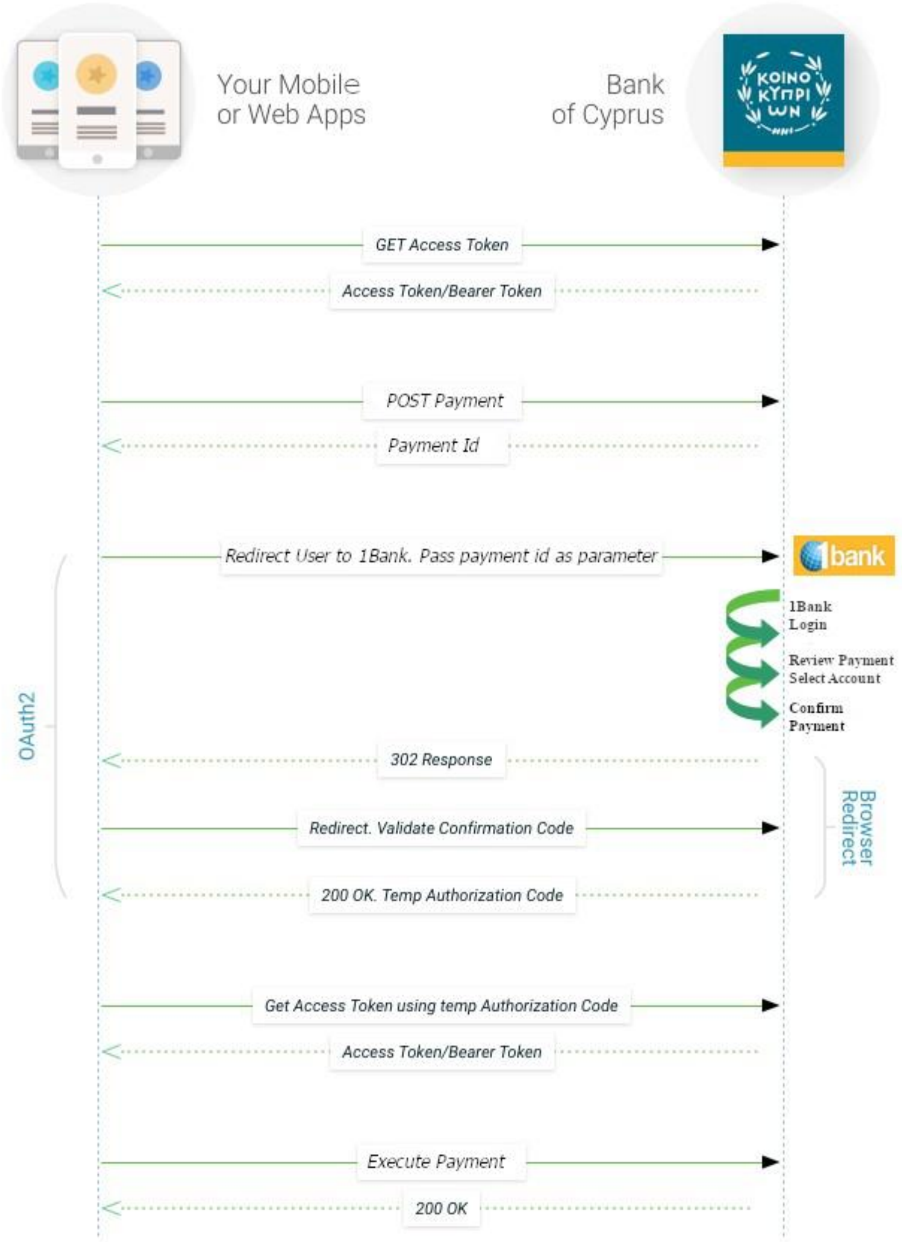
The users will be requested to login to 1Bank by supplying a UserID and a password. They will then be requested to select an ordering account if it was not provided, review the payment details and confirm the payment. As part of this Authorization workflow the Client would be provided with a specific Temporary Authorization Code which is used to get an access token and then call the POST Execute Payment API to complete the payment.

Follow **these steps** to call Payments APIs.

Note, that the initiate payment API requires the Client to sign the payment payload with your eIDAS QSeal Certificate and provide it as a JWS. The expected signing algorithm is RS256. JWS is a json format of JWT to be used in API payloads. You can find several libraries that do this at “https://jwt.io”. You can also find a few articles on JWS on the internet that explain this very well.

In sandbox an additional API is provided for simulating the API-Store signing functionality required for payments to assist the developers during implementation.

Payment Flow Sequence Diagram


Your Mobile or Web Apps
Bank of Cyprus
ΚΟΙΝΟ ΚΥΠΡΙΩΝ
GET Access Token
Access Token/Bearer Token
POST Payment
Payment Id
Redirect User to 1Bank. Pass payment id as parameter
1bank
1Bank Login
Review Payment Select Account
Confirm Payment
OAuth2
302 Response
Redirect. Validate Confirmation Code
Browser Redirect
200 OK. Temp Authorization Code
Get Access Token using temp Authorization Code
Access Token/Bearer Token
Execute Payment
200 OK

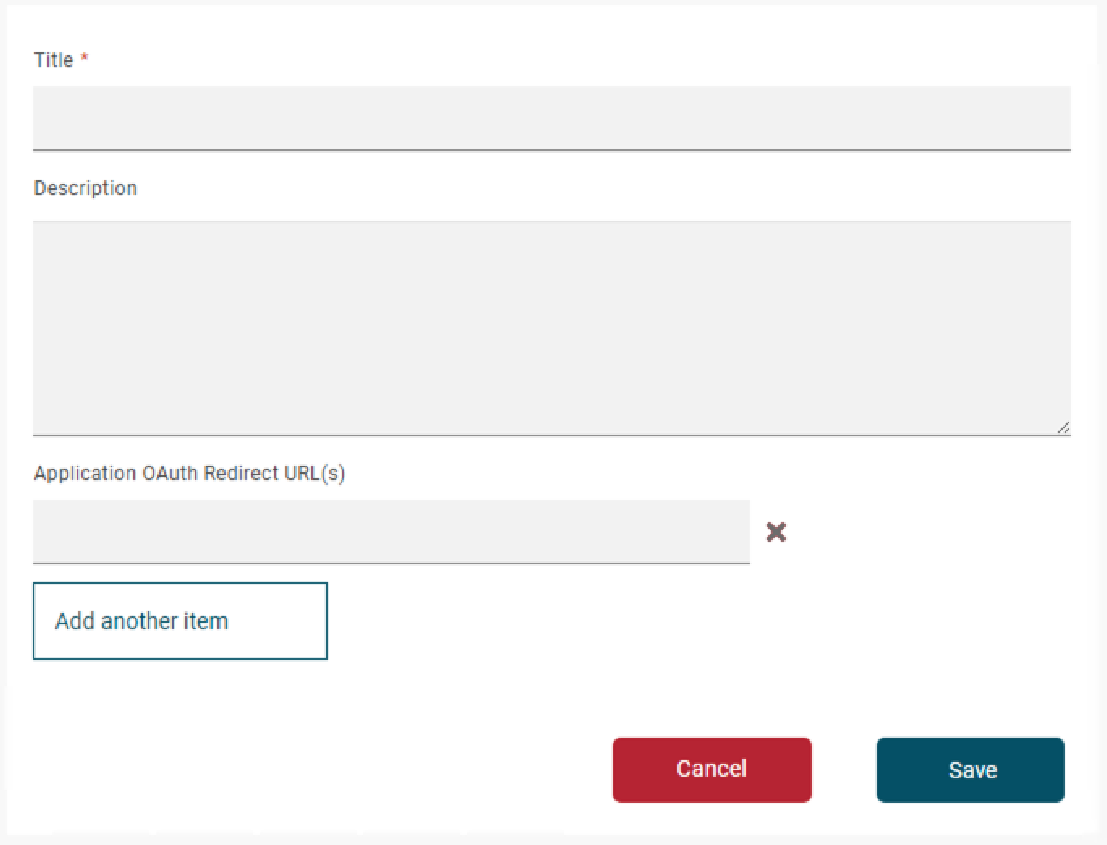
B2B APIs

Using B2B payments you can initiate in one call multiple payments which can then be authorized by an appropriate signatory at a later stage using the 1Bank

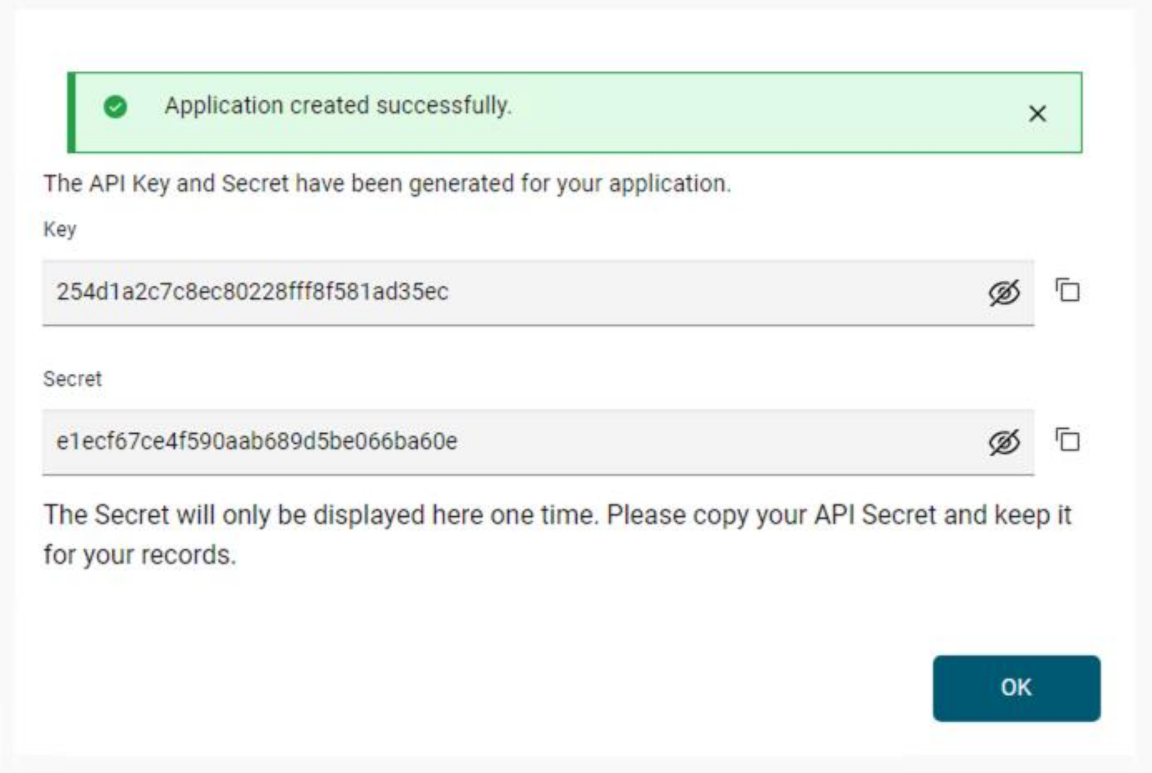
“Pending for Authorization Page”. There are two types of B2B payments. The first one is “Mass Payments” which can be used to initiate all kinds of transfers including transfers within BoC, SEPA and SWIFT. The other type is “Payroll/Group Transfers” which have a single ordering account for all credits and are handled as a single transaction so that only one transaction will appear in your payment account statement.

Register an application

Use "My Apps" menu to register an application. When you register an application, you need to provide the OAuth redirect URI. This is a mandatory field for the login/authentication mechanism to be able to return the oauth token back to your application.



After registering the application, a new screen will appear with the assigned unique client ID and client secret. You must verify the Client Secret by taking the value from the client secret in the beginning of the page.



Application created successfully.

The API Key and Secret have been generated for your application.

Key

254d1a2c7c8ec80228fff8f581ad35ec

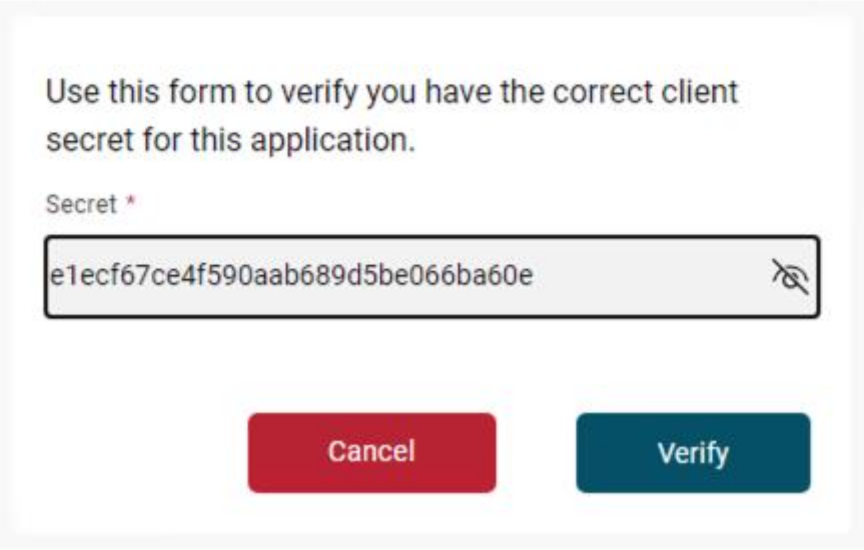
Secret

e1ecf67ce4f590aab689d5be066ba60e

The Secret will only be displayed here one time. Please copy your API Secret and keep it for your records.

Make a note of your client ID and client secret. These will be needed for your application to access the API. Some APIs need only the client id, while others (the ones that need to authenticate the end user first) need both the client id and client secret for the oauth end user authentication.

Your client secret will only be displayed once. If you forget or lose it, you can verify the secret to see if it's correct or reset it to get a new one. The verification screen will ask to enter the secret Id.

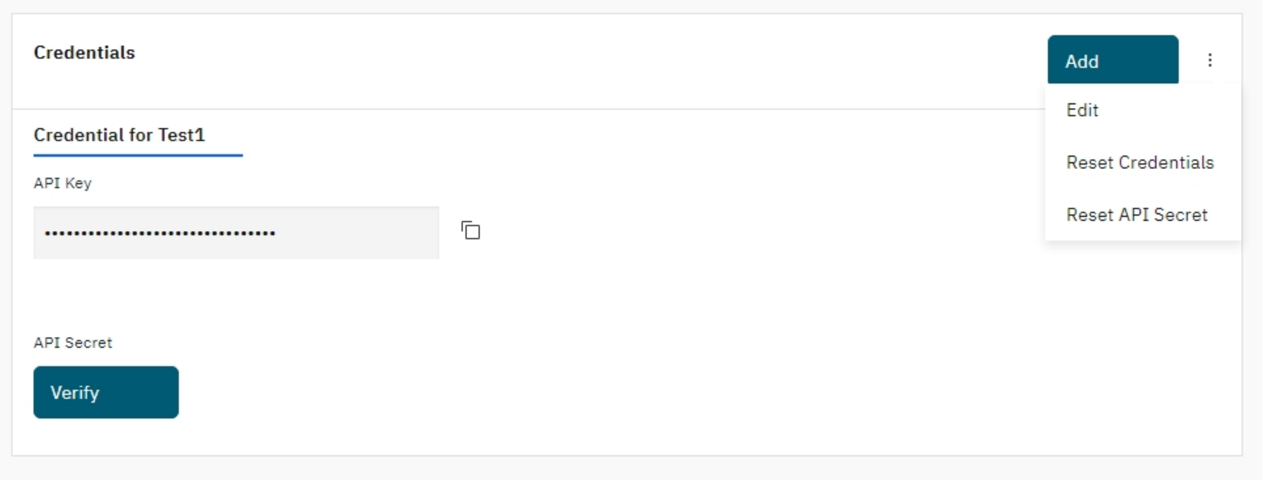


Use this form to verify you have the correct client secret for this application.

Secret *

e1ecf67ce4f590aab689d5be066ba60e

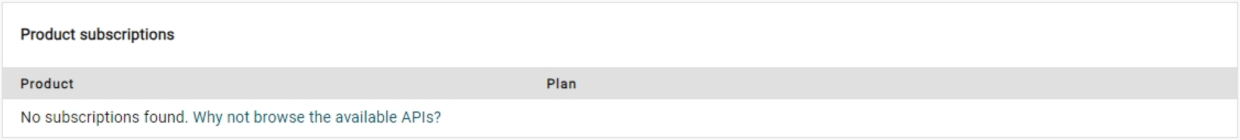
The credential screen will show the client Id and the option to reset the values.



API Key: ································

Select a product / plan

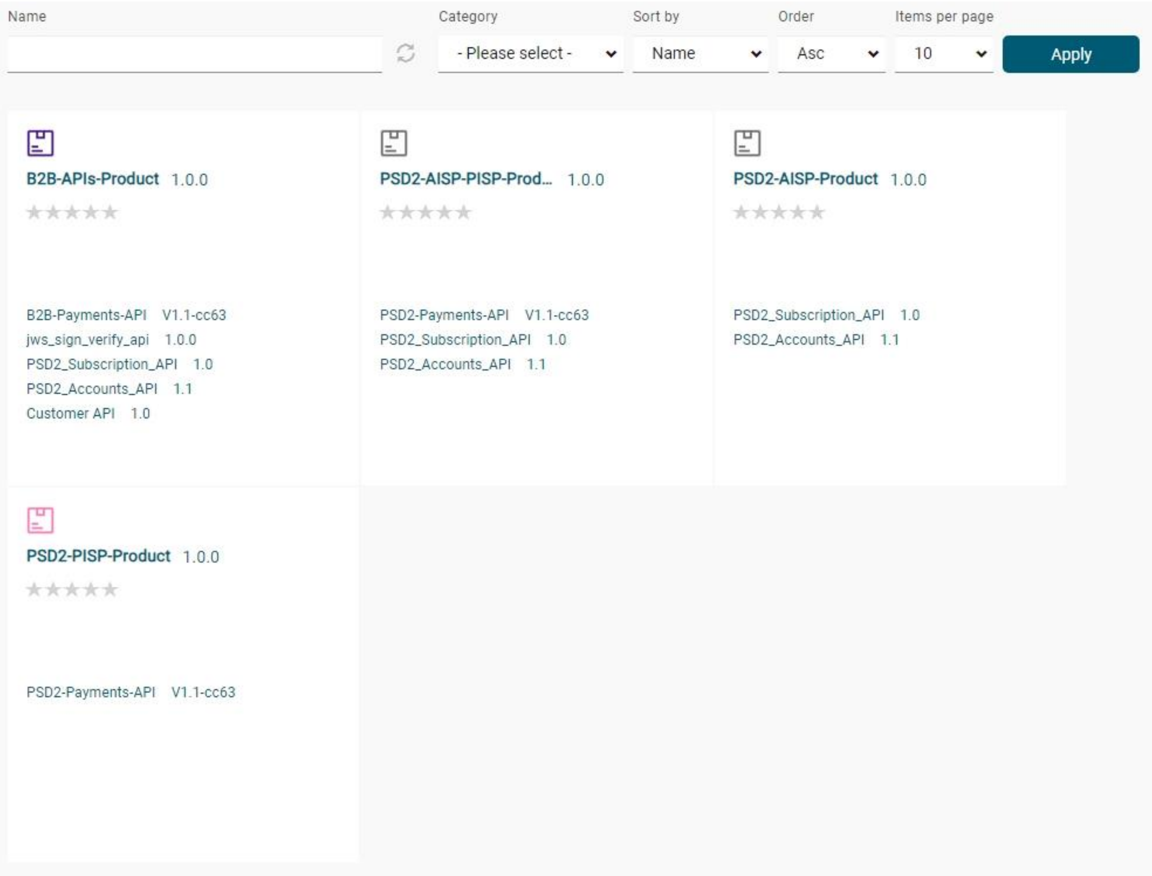
The registered application needs to be subscribed to a plan of one or more APIs. For the first time the link to available APIs will be in the Subscription Area of the Application page.



Product subscriptions

Product	Plan
No subscriptions found. Why not browse the available APIs?	

When selecting the Available APIs, the API list screen will appear. Any API can be selected for this application.



Name

Category: - Please select -

Sort by: Name

Order: Asc

Items per page: 10

B2B-APIs-Product 1.0.0

B2B-Payments-API V1.1-cc63
jws_sign_verify_api 1.0.0
PSD2_Subscription_API 1.0
PSD2_Accounts_API 1.1
Customer API 1.0

PSD2-AISP-PISP-Prod... 1.0.0

PSD2-Payments-API V1.1-cc63
PSD2_Subscription_API 1.0
PSD2_Accounts_API 1.1

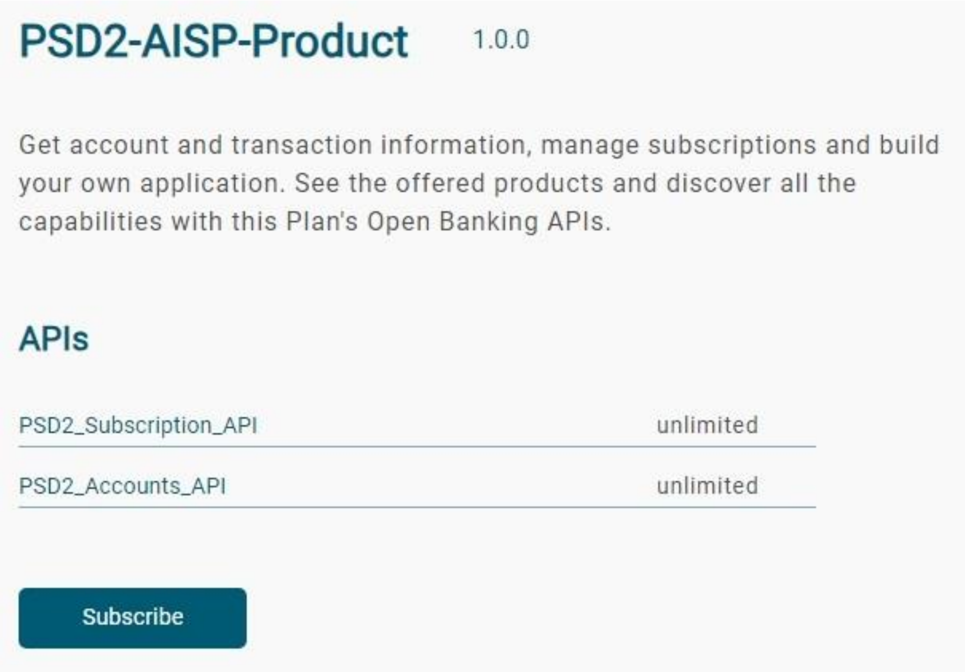
PSD2-AISP-Product 1.0.0

PSD2_Subscription_API 1.0
PSD2_Accounts_API 1.1

PSD2-PISP-Product 1.0.0

PSD2-Payments-API V1.1-cc63

After selecting any of the APIs, the subscription screen will appear to subscribe the API plan to the Application.



PSD2-AISP-Product 1.0.0

Get account and transaction information, manage subscriptions and build your own application. See the offered products and discover all the capabilities with this Plan's Open Banking APIs.

APIs

PSD2_Subscription_API unlimited
PSD2_Accounts_API unlimited

The subscribed API can be seen in the SUBSCRIPTION section of the Application.

All the APIs mentioned above can be found on the APIs catalogue. You can navigate to it using global tabs menu.

How to test our APIs?

The documentation that follows describes how the APIs can be tested with examples from the Sandbox environment where you can register for free to experiment and test our APIs. Once you finish your implementation you can switch to the production environment by changing the base URL of the examples. The URLs are listed below:

PSD2 Providers:
https://apis-secure.bankofcyprus.com/df-boc-org-prd/prod/psd2/v2

B2B Customers:
https://apis.bankofcyprus.com/df-boc-org-prd/prod/psd2

TEST SUBSCRIPTION APIS

The following steps are needed to test the BOC APIs for the creation of subscriptions. Samples are provided based on the Swagger file provided in the portal and on testing scripts in Postman.

Below you can find the link to a test suite with the steps required to test the Subscription APIs. It includes HTTP requests with sample headers and test data.

Postman is a free REST test tool; you can download the tool from here: https://www.getpostman.com/

Save the content of the link below as a .json file and then import it in the tool as a test collection.

PSD2-PROD-SAMPLES.postman_collection.json

PSD2-Sandbox Empty.postman_environment.json

The rest of this document explains the steps implemented in the test suite. You can use this information as guidance for the logic you need to implement in your application to be able to create a subscription ID.

Before calling any of the APIs, you must ensure that you follow the 'Create Subscription' process on [How to access and call our APIs](#) section.

Steps for initiating and activating a subscription

1. Obtain an access token to invoke the Subscription API:

In this step you use the TPP credentials to obtain the access token required to submit in each API call.

Call the POST /oauth2/token endpoint in the tppoauth2security API. The following parameters must be passed as headers in the HTTP request (Mandatory fields highlighted in Red).

Example Request

```
curl --request POST \ --url https://sandbox-apis.bankofcyprus.com/df-boc-org-sb/sb/psd2/oauth2/token \
 --header accept: application/json \
 --header content-type: application/x-www-form-urlencoded \
 --data grant_type=client_credentials&client_id={{your client id}}&client_secret={{your client secret}}&scope=TPPOAuth2Security
```

Example Response

POST https://sandbox-apis.bankofcyprus.com/df-boc-org-sb/sb/psd2/oauth2/token

```
{
"token_type": "bearer",
"access_token": "6334262361656275",
"expires_in": 3600,
"consented_on": 1542264419,
"scope": "TPPOAuth2Security",

}
```

The access token received will be used in all the API calls.

2. Obtain a Subscription Id

The subscription Id will be used in all the API calls. The Subscription ID must be authorized to be used. For the POST subscription API, you will need the Authorization Token from the previous call.

Example Request

```
curl --request POST \ --url https://sandbox-apis.bankofcyprus.com/df-boc-org-sb/sb/psd2/v1/subscriptions \--header Authorization: Bearer {{oauth_token}}\

--header Content-Type: application/json \

--header timeStamp: {{$timestamp}} \

--header journeyId: {{$guid}} \

--header Content-Type: application/json \

--data {

 "accounts": {

   "transactionHistory": true,

   "balance": true,

   "details": true,

   "checkFundsAvailability": true

  },

  "payments": {

   "limit": 99999999,

   "currency": "EUR",

   "amount": 999999999

  }

}'
```

Example Response

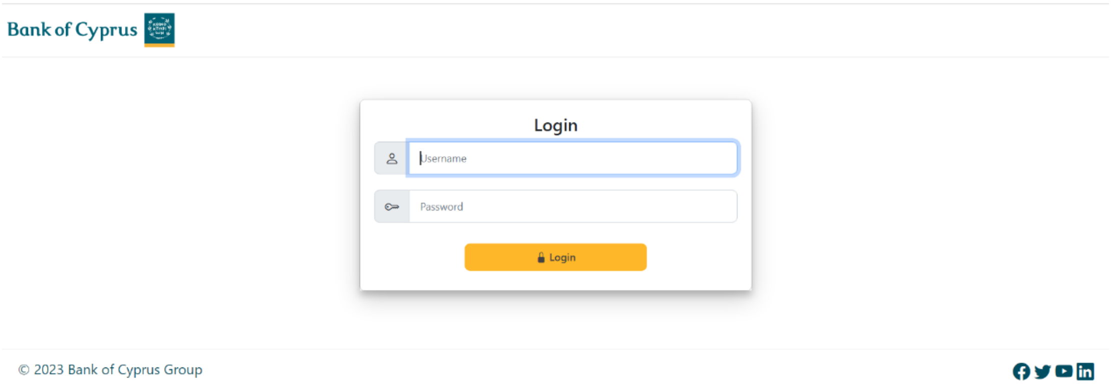
```
{
  "duration": {
  "startDate": "20/11/2017",
  "endDate": "20/11/2018"
  },
  "subscriptionId": "Subid000001-1590333432791",
  "status": "PENDING",
  "description": "SUBSCRIPTION",
  "selectedAccounts": [],
  "accounts": {
    "transactionHistory": true,
    "balance": true,
    "details": true,
    "checkFundsAvailability": true
  },
  "payments": {
    "limit": 99999999,
    "currency": "EUR",
    "amount": 999999999
  }
}
```

3. Select Accounts for the Subscription Id

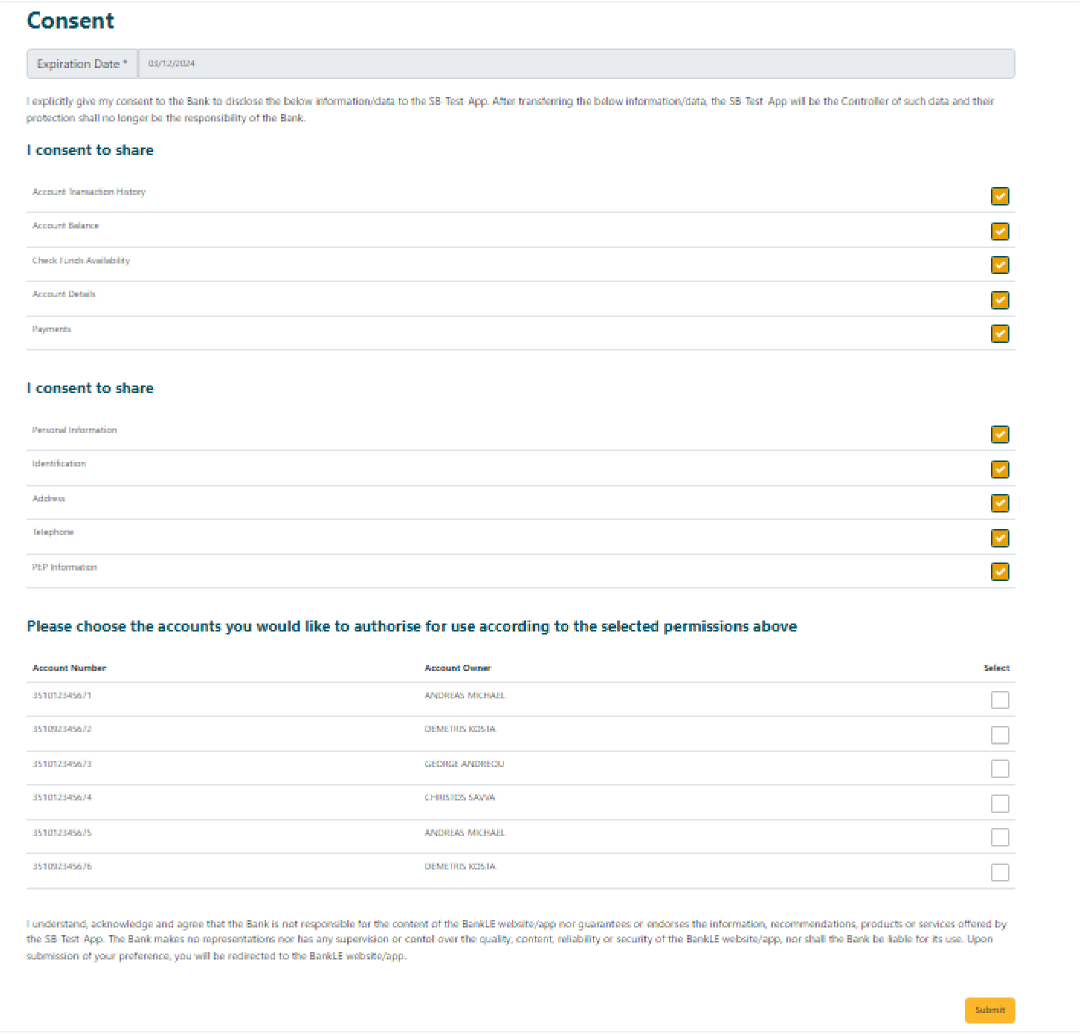
After initiating the subscription, the user needs to select the accounts to be used with this subscription and provide a consent. This can be done by redirecting the user to the below URL:

Example Request in Browser

https://sandbox-apis.bankofcyprus.com/df-boc-org-sb/sb/psd2/oauth2/authorize?response_type=code&redirect_uri={{yourAppRedirectionURL}}&scope=UserOAuth2Security&client_id={{yourClientId}}&subscriptionid={{subscriptionId}}



After entering the username=999999 and passcode=112233 we get the following screen.



Consent

Expiration Date *	03/12/2024

I explicitly give my consent to the Bank to disclose the below information/data to the SB Test App. After transferring the below information/data, the SB Test App will be the Controller of such data and their protection shall no longer be the responsibility of the Bank.

I consent to share

Account Transaction History	☑
Account Balance	☑
Check Funds Availability	☑
Account Details	☑
Payments	☑

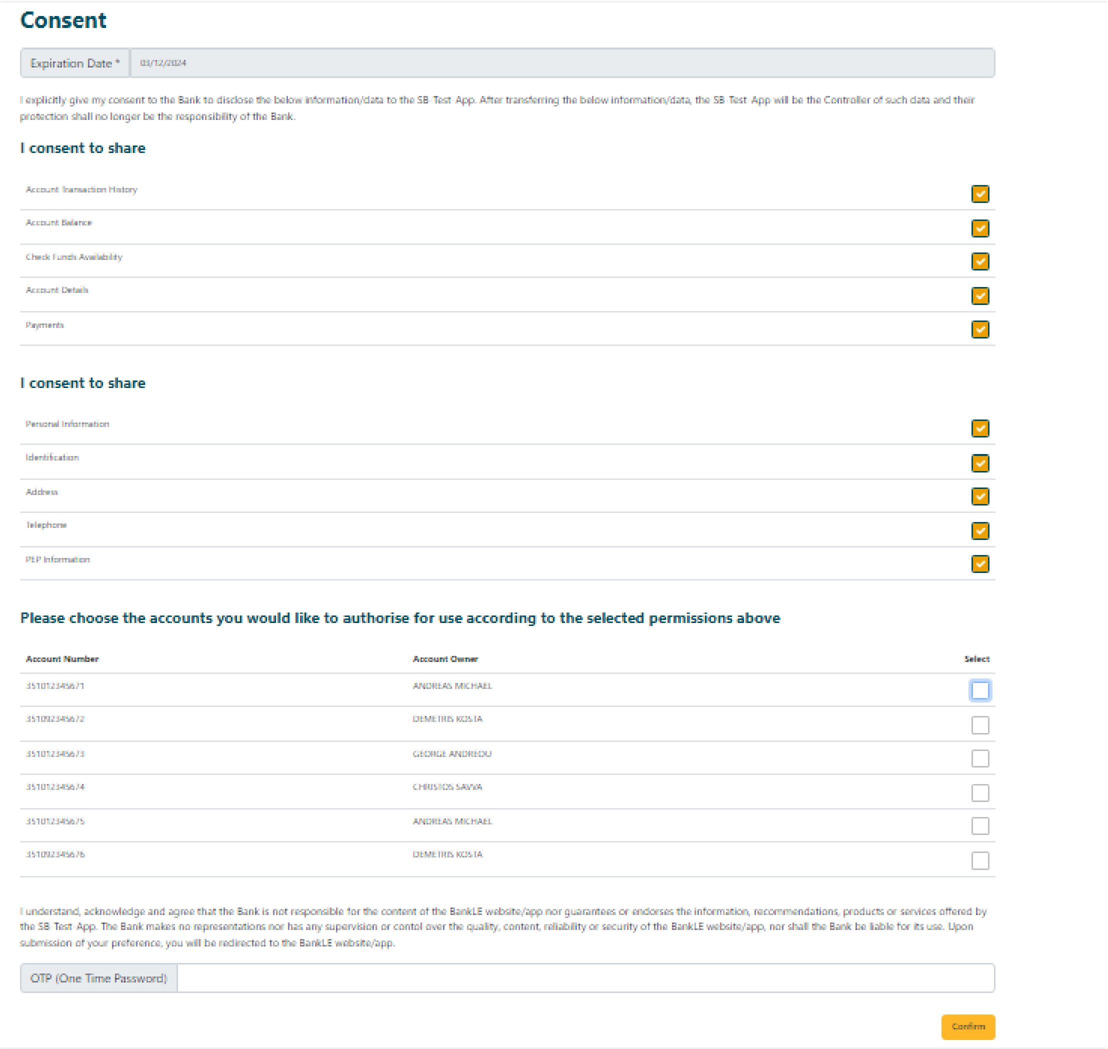
I consent to share

Personal Information	☑
Identification	☑
Address	☑
Telephone	☑
PEP Information	☑

Please choose the accounts you would like to authorise for use according to the selected permissions above

Account Number	Account Owner	Select
351012345671	ANDREAS MICHAEL	□
351092345672	DEMETRIS KOSTA	□
351012345673	GEORGE ANDREOU	□
351012345674	CHRISTOS SAVVA	□
351012345675	ANDREAS MICHAEL	□
351092345676	DEMETRIS KOSTA	□

I understand, acknowledge and agree that the Bank is not responsible for the content of the BankLE website/app nor guarantees or endorses the information, recommendations, products or services offered by the SB Test App. The Bank makes no representations nor has any supervision or contol over the quality, content, reliability or security of the BankLE website/app, nor shall the Bank be liable for its use. Upon submission of your preference, you will be redirected to the BankLE website/app.



Consent

Expiration Date * 03/12/2024

I explicitly give my consent to the Bank to disclose the below information/data to the SB Test App. After transferring the below information/data, the SB Test App will be the Controller of such data and their protection shall no longer be the responsibility of the Bank.

I consent to share

Account Transaction History	☑
Account Balance	☑
Check Funds Availability	☑
Account Details	☑
Payments	☑

I consent to share

Personal Information	☑
Identification	☑
Address	☑
Telephone	☑
PEP Information	☑

Please choose the accounts you would like to authorise for use according to the selected permissions above

Account Number	Account Owner	Select
351012345671	ANDREAS MICHAEL	☐
351012345672	DEMETRIS KOSTA	☐
351012345673	GEORGE ANDREOU	☐
351012345674	CHRISTOS SAVVA	☐
351012345675	ANDREAS MICHAEL	☐
351012345676	DEMETRIS KOSTA	☐

I understand, acknowledge and agree that the Bank is not responsible for the content of the BankLE website/app nor guarantees or endorses the information, recommendations, products or services offered by the SB Test App. The Bank makes no representations nor has any supervision or contol over the quality, content, reliability or security of the BankLE website/app, nor shall the Bank be liable for its use. Upon submission of your preference, you will be redirected to the BankLE website/app.

OTP (One Time Password)

After you select the accounts and provide your consent the user will be redirected back to your app’s redirect URL with the authorization code as a query parameter.

Example Response

https://localhost/?code=AAIB7CW9gOFNa6c3B5l9GsfSHT

4. Get second Access token for the update of subscription

Example Request

```
curl --request POST \ --url https://sandbox-apis.bankofcyprus.com/df-boc-org-sb/sb/psd2/oauth2/token \

--header accept: application/json \

--header content-type: application/x-www-form-urlencoded \

--data
grant_type=authorization_code&client_id={{yourClientId}}&client_secret={(your
ClientSecret}}&code= {{code from redirect}}&scope=UserOAuth2Security
```

Example Response

```
  {
"token_type": "bearer",

"access_token": "6334262361656275",

"expires_in": 3600,

"consented_on": 1542264419,

"scope": "UserOAuth2Security",

}
```

The access token will be used for the authorization of the Subscription Id below

5. Update Subscription Id

The last step is the activation of the subscription which is done by calling PATCH Subscription. Before calling it, get the details of the subscription including the accounts as accepted by the user by calling the GET Subscription API and use the details when calling the PATCH API.

GET Subscription Sample Request:

```
  curl --location --request GET "https://sandbox-apis.bankofcyprus.com/df-boc-org-sb/sb/psd2/v1/subscriptions/{{subscriptionId}}" \

--header "Authorization: Bearer {{oauth_code}}"\

--header "Content-Type: application/json" \

--header "journeyid: {{$guid}}" \

--header "timestamp: {{$timestamp}}" \
```

GET Subscription Sample Response:

```
{
  "SubscriptionId": "Subid000001-1725429256148",
  "status": "PENDING",
  "description": "SUBSCRIPTION",
  "accounts": {
   "transactionHistory": true,
   "balance": true,
   "details": true,
   "checkFundsAvailability": true
  },
  "payments": {
   "limit": 50,
   "currency": "string",
   "amount": 50
  },
  "expirationDate": "03/12/2024"
 }
```

Example PATCH Subscription Request:

```
 curl —-location --request PATCH
‘https://sandbox-apis.bankofcyprus.com/df-boc-org-sb/sb/psd2/v1/subscriptions/{{subscriptionId}}’ \
--header ‘authorization: Bearer {{oauth_code}}’ \
--header ‘content-type: application/json’ \
--header ‘journeyI~~i~~d: {{$guid}}’ \
--header ‘timestamp: {{$timestamp}}’ \
--data '{
  "accounts": {
    "transactionHistory": true,
    "balance": true,
    "details": true,
    "checkFundsAvailability": true
  },
  "payments": {
    "limit": 50,
    "currency": "string",
    "amount": 50
  },
 }'
```

Example PATCH Subscription Response:

```
{
 "subscriptionId": "Subid000001-1725429256148",
 "status": "ACTV",
 "description": "SUBSCRIPTION",
 "selectedAccounts": [
 {
  "accountId": "351012345671"
 },
 {
  "accountId": "351092345672"
 },
 {
  "accountId": "351012345673"
 },
 {
  "accountId": "351012345674"
 },
 {
  "accountId": "351012345675"
 },
 {
  "accountId": "351092345676"
 }
],
"accounts": {
 "transactionHistory": true,
 "balance": true,
 "details": true,
 "checkFundsAvailability": true
},
"payments": {
 "limit": 50,
 "currency": "string",
   "amount": 50
  },
  "duration": {
   "startDate": "04/09/2024",
   "endDate": "03/12/2024"
  }
}
```

The Subscription is now ACTIVE to be used for all the other APIs (Accounts and B2B Payments).

TESTING ACCOUNT APIS

The following steps are needed to test the BOC APIs for the different actions on customer accounts. Samples are provided based on the Swagger file provided in the portal and on testing scripts in Postman.

Below you can find the link to a test suite with the steps required to test the following actions, it includes HTTP requests with sample headers and test data.

1. Get Accounts for specific subscription IDs
2. Get Accounts Details
3. Get Available Balance API
4. Get Account Statement API

1. Get Accounts for specific subscription IDs

Example Request

```
curl --location --request GET "https://sandbox-apis.bankofcyprus.com/df-boc-org-sb/sb/psd2/v1/accounts"

--header "Content-Type: application/json" \

--header "Authorization: Bearer {{oauth_code}}"\

--header "subscriptionId: {{subscription_id}}" \
--header 'journeyId: {{uuid}}' \
--header "timestamp: {{$timestamp}}" \
```

Example Response

```
[
  {
    "bankId": "12345671",
    "accountId": "351012345671",
    "accountAlias": "ANDREAS",
    "accountType": "CURRENT",
    "accountName": "ANDREAS MICHAEL",
    "IBAN": "CY11002003510000000012345671",
    "currency": "EUR",
    "infoTimeStamp": "1511779237",
    "interestRate": 0,
    "maturityDate": "19/11/2018",
    "lastPaymentDate": "19/11/2017",
    "nextPaymentDate": "19/12/2017",
    "remainingInstallments": 10,
    "balances": []
  }
]
```

2. Get Accounts Details

This API will retrieve the information for specific accounts.

Example Request

```
curl --location --request GET "https://sandbox-apis.bankofcyprus.com/df-boc-org-sb/sb/psd2/v1/accounts/{{account_number}}’ \
--header "Content-Type: application/json" \
--header "Authorization: Bearer {{oauth_code}}” \
--header "subscriptionId: {{subscription_id}}" \
--header 'journeyId: {{uuid}}' \
--header " timestamp: {{$timestamp}}" \
```

Example Response

```
[
  {
    "bankId": "12345671",
    "accountId": "351012345671",
    "accountAlias": "ANDREAS",
    "accountType": "CURRENT",
    "accountName": "ANDREAS MICHAEL",
    "IBAN": "CY11002003510000000012345671",
    "currency": "EUR",
    "infoTimeStamp": "1511779237",
    "interestRate": 0,
    "maturityDate": "19/11/2018",
    "lastPaymentDate": "19/11/2017",
    "nextPaymentDate": "19/12/2017",
    "remainingInstallments": 10,
    "balances": [
      {
        "amount": 1000,
        "balanceType": "AVAILABLE"
      },
      {
        "amount": 1000,
        "balanceType": "CURRENT"
      }
    ]
  }
]
```

3. Get Available Balance API

The GetBalance API uses the same headers as AccountDetails API.

Example Request

```
curl --location --request GET "https://sandbox-apis.bankofcyprus.com/df-boc-org-sb/sb/psd2/v1/accounts/{{account_number}}/balance" \
--header "Content-Type: application/json" \
--header "Authorization: Bearer {{oauth_code}}"\
--header "subscriptionId: {{subscription_id}}" \
--header 'journeyId: {{uuid}}' \
--header "timestamp: {{$timestamp}}" \
```

Example Response

```
[
  {
    "bankId": "12345671",
    "accountId": "351012345671",
    "accountAlias": "ANDREAS",
    "accountType": "CURRENT",
    "accountName": "ANDREAS MICHAEL",
    "IBAN": "CY11002003510000000012345671",
    "currency": "EUR",
    "infoTimeStamp": "1511779237",
    "interestRate": 0,
    "maturityDate": "19/11/2018",
    "lastPaymentDate": "19/11/2017",
    "nextPaymentDate": "19/12/2017",
    "remainingInstallments": 10,
    "balances": [
      {
        "amount": 1000,
        "balanceType": "AVAILABLE"
      },
      {
        "amount": 1000,
        "balanceType": "CURRENT"
      }
    ]
  }
]
```

4. Get Account Statement API

The GetAccStatement API retrieves the transactions of an account:

Example Request:

```
curl --location --request GET "https://sandbox-apis.bankofcyprus.com/df-boc-org-sb/sb/psd2/v1/accounts/{{account_number}}/statement?startDate=16/04/2024&endDate=16/05/2024&maxCount=10"\

--header "Content-Type: application/json" \

--header "Authorization: Bearer {{oauth_code}}"\
--header "subscriptionId: {{subscription_id}}" \

--header 'journeyId: {{uuid}}' \

--header "timestamp: {{$timestamp}}" \
```

Example Response:

```
{
"account": {
  "bankId": "12345671",
  "accountId": "351012345671",
  "accountAlias": "ANDREAS",
  "accountType": "CURRENT",
  "accountName": "ANDREAS MICHAEL",
  "IBAN": "CY11002003510000000012345671",
  "currency": "EUR",
  "infoTimeStamp": "1511779237"
},
"transaction": [
  {
    "id": "663c9d26de9162079842ce59",
    "dcInd": "DEBIT",
    "transactionAmount": {
      "amount": 30,
      "currency": "EUR"
    },
    "description": "SWIFT Transfer",
    "postingDate": "09/05/2024",
    "valueDate": "09/05/2024",
    "transactionType": "PAYMENT"
  },
  {
   "id": "664307a7de9162079842cef8",
   "transactionAmount": {
    "amount": 10,
    "currency": "EUR"
   },
   "postingDate": "14/05/2024",
   "valueDate": "14/05/2024"
  },
  {
   "id": "6644a7c9d983f403982a0b1e",
   "transactionAmount": {
    "amount": 10,
    "currency": "EUR"
   },
   "postingDate": "15/05/2024",
   "valueDate": "15/05/2024"
  }
 ]
}
```

TESTING PAYMENTS APIs

The following steps are needed to test BOC APIs for the creation of a payment.

1. Sign Request (JWS_Sign_Verify API)

When creating a payment, the payload must be signed with a Digital Certificate. To simplify this step in Sandbox an API to sign the payment payload had been provided. In the actual production environment this step must be done by the app and this API is not available.

Example Request

```
curl --location --request POST "https://sandbox-apis.bankofcyprus.com/df-boc-org-sb/sb/jwssignverifyapi/sign" \

--header "Content-Type: application/json" \

 --header 'tppId: singpaymentdata' \

--data-raw "{

        "debtor": {

                "bankId": "",

                "accountId": "351012345671"

        },

        "creditor": {

                "bankId": "CITIUS33",

                "accountId": "48193222324233"

        },

        "transactionAmount": {

                "amount": 30,

                "currency": "EUR"

        },

        "paymentDetails": "SWIFT Transfer"

}"
```

Example Response

```
{

  "payload":
"eyAiZGVidG9yIjp7ICJiYW5rSWQiOiIiLCAiYWNjb3VudElkIjoiMzUxMDEyMzQ1Njc
xIiB9LCAiY3JlZGl0b3IiOnsgImJhbmtJZCI6IkNJVElVUzMzIiwgImFjY291bnRJZCI6Ij
Q4MTkzMjIyMzI0MjMzIiB9LCAidHJhbnNhY3Rpb25BbW91bnQiOnsgImFtb3VudC
I6MzAsICJjdXJyZW5jeSI6IkVVUiIgfSwgInBheW1lbnREZXRhaWxzIjoiU1dJRlQgVHJhbnNmZXIiIH0",
  "signatures": [
    {
      "protected": "eyJhbGciOiJSUzI1NiJ9",
      "signature": "s9vy53hGobNDeuQGyQI1J4-Kopo7AsVPMNYuyku9PLV2UXSAzkEfPQQPHYsAHe4ZnArv06XDp2Qsnqti5v88IWIDQe1AlVmNLEiVmkIBwXjsSWcRaNqVPWVas70SuO6ddrqH1Vz_UbvBJD02e49iDhuuCnsKZYBU7jvo4o-JvHyWXneXFElQvXKSCA-iddaivXdKWEuv7R2pkDr3xOJKJ4xS8Ugt5vKUVWMVQhDK6fOfzh50VeCSxC0v-XByMC4wLZcb4HbPtH9YEtP0MqF_AkqFRGD8v5OBBYr6pQdQ7oBRe1N6a9UkAhG0UDrfZFPoD6m1Gbdd9__RspWOU7fMDA"
    }
  ]
}
```

In case you want to test the JWS signing in the Sandbox Environment you can use the below key which is the same used by the “Sign Request” API.

sandbox-cert.pem

sandbox-key.pem

The JWS signature string will be provided on the body of the API call to initiate the payment.

Example Request

```
curl --location --request POST "https://sandbox-apis.bankofcyprus.com/df-boc-org-sb/sb/psd2/v1/payments/initiate’\

--header "Content-Type: application/json" \

--header "Authorization: Bearer {{oauth_code}}” \

--header 'journeyId: {{uuid}}' \

--header ‘timestamp: {{$timestamp}}’ \

--header 'customerDevice: {{yourDevice}}' \

--header 'customerIP: 10.0.0.0' \

--header 'customerSessionId: 1232545908' \

--header 'loginTimeStamp: {{$timestamp}}' \

--data-raw "{

  "payload": "eyAiZGVidG9yIjp7ICJiYW5rSWQiOiIiLCAiYWNjb3VudElkIjoiMzUxMDEyMzQ1NjcxIiB9LCAiY3JlZGl0b3IiOnsgImJhbmtJZCI6IkNJVElVUzMzIiwgImFjY291bnRJZCI6IjQ4MTkzMjIyMzI0MjMzIiB9LCAidHJhbnNhY3Rpb25BbW91bnQiOnsgImFtb3VudCI6MzAsICJjdXJyZW5jeSI6IkVVUiIgfSwgInBheW1lbnREZXRhaWxzIjoiU1dJRlQgVHJhbnNmZXIiIH0",

  "signatures": [

    {

      "protected": "eyJhbGciOiJSUzI1NiJ9",

      "signature": "s9vy53hGobNDeuQGyQI1J4-Kopo7AsVPMNYuyku9PLV2UXSAzkEfPQQPHYsAHe4ZnArv06XDp2Qsnqti5v88IWIDQe1AlVmNLEiVmkIBwXjsSWcRaNqVPWVas70SuO6ddrqH1Vz_UbvBJD02e49iDhuuCnsKZYBU7jvo4o-JvHyWXneXFElQvXKSCA-iddaivXdKWEuv7R2pkDr3xOJKJ4xS8Ugt5vKUVWMVQhDK6fOfzh50VeCSxC0v-XByMC4wLZcb4HbPtH9YEtP0MqF_AkqFRGD8v5OBBYr6pQdQ7oBRe1N6a9UkAhG0UDrfZFPoD6m1Gbdd9__RspWOU7fMDA"

    }

  ]

}"
```

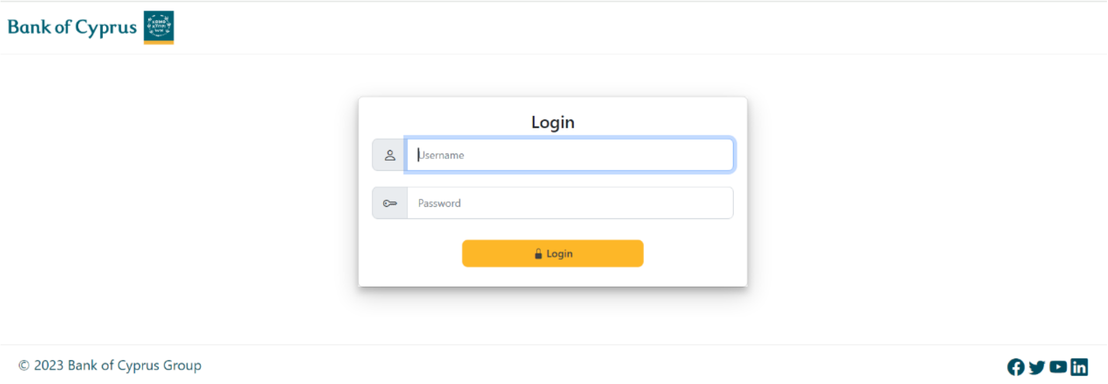
Example Response:

```
{
 "authCodeNeeded": true,
 "payment": {
  "paymentId": "1ccec876-2440-48d7-9751-7ea57935335e",
  "charges": [
   {
    "name": "Transfer_charges",
    "value": "10"
   }
  ]
 }
}
```

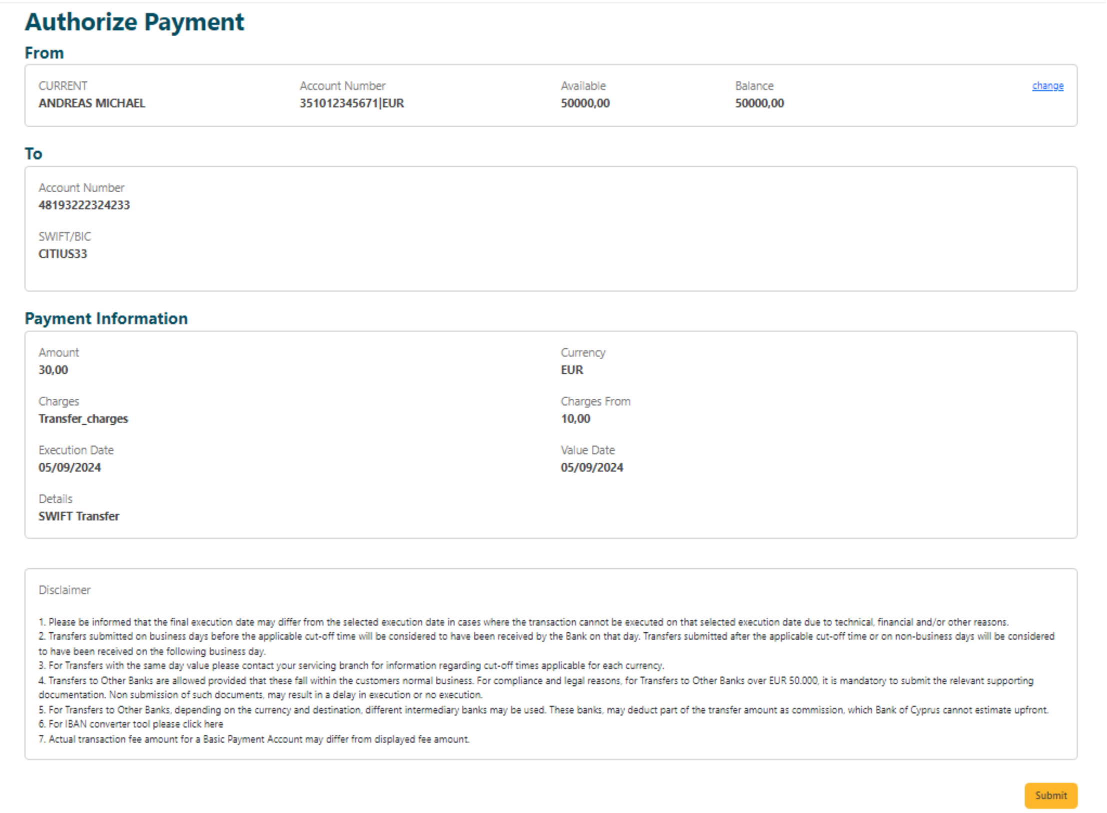
3. Review the payment and Select Account

After initiating the payment, the user needs to review the payment, select the debit account to be used if needed with this subscription and provide a consent. This can be done by redirecting the user to the below URL:

https://sandbox-apis.bankofcyprus.com/df-boc-org-sb/sb/psd2/oauth2/authorize?response_type=code&redirect_uri={{yourAppRedirectionURL}}&scope=UserOAuth2Security&client_id={{yourClientId}}&paymentid={{PaymentId}}



After entering the username=999999 and passcode=112233 we get the following screen.



Authorize Payment

From

CURRENT	Account Number	Available	Balance
ANDREAS MICHAEL	351012345671|EUR	50000,00	50000,00

To

Account Number
48193222324233

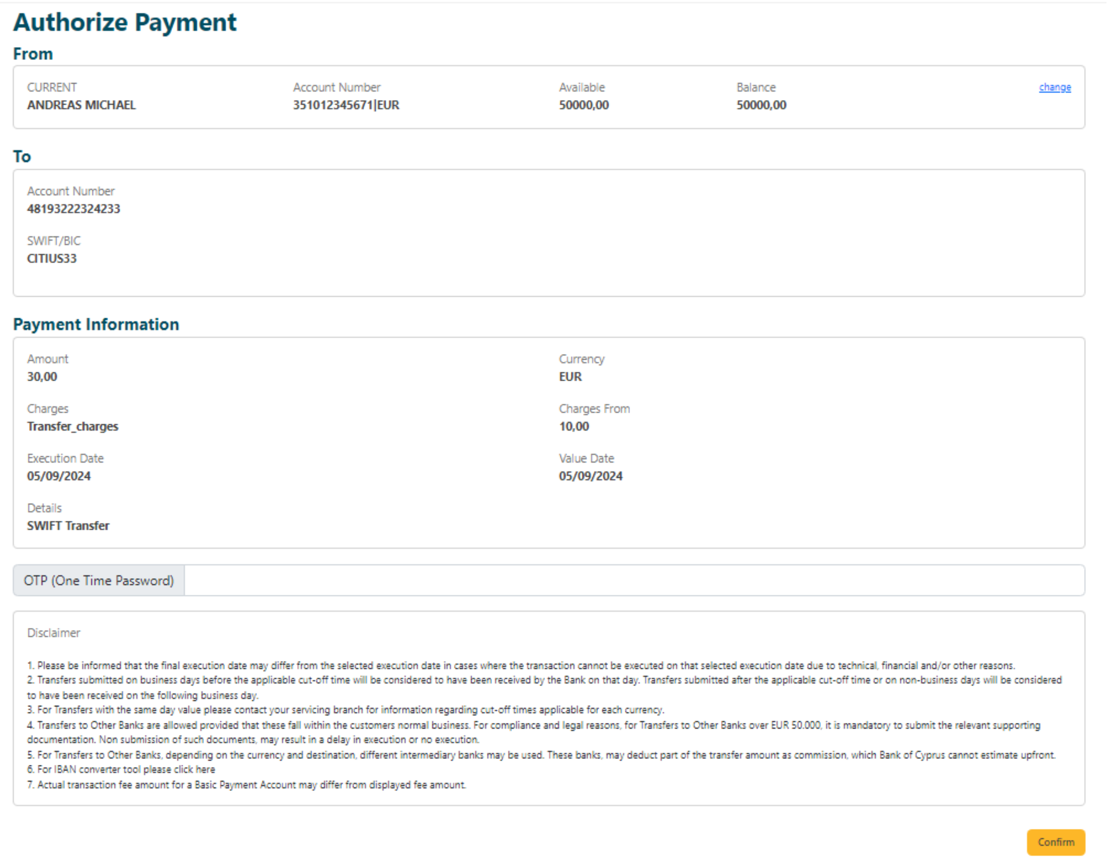
SWIFT/BIC
CITIUS33

Payment Information

Amount 30,00	Currency EUR
Charges Transfer_charges	Charges From 10,00
Execution Date 05/09/2024	Value Date 05/09/2024
Details SWIFT Transfer	

Disclaimer

1. Please be informed that the final execution date may differ from the selected execution date in cases where the transaction cannot be executed on that selected execution date due to technical, financial and/or other reasons.
2. Transfers submitted on business days before the applicable cut-off time will be considered to have been received by the Bank on that day. Transfers submitted after the applicable cut-off time or on non-business days will be considered to have been received on the following business day.
3. For Transfers with the same day value please contact your servicing branch for information regarding cut-off times applicable for each currency.
4. Transfers to Other Banks are allowed provided that these fall within the customers normal business. For compliance and legal reasons, for Transfers to Other Banks over EUR 50.000, it is mandatory to submit the relevant supporting documentation. Non submission of such documents, may result in a delay in execution or no execution.
5. For Transfers to Other Banks, depending on the currency and destination, different intermediary banks may be used. These banks, may deduct part of the transfer amount as commission, which Bank of Cyprus cannot estimate upfront.
6. For IBAN converter tool please click here
7. Actual transaction fee amount for a Basic Payment Account may differ from displayed fee amount.



Authorize Payment

From

CURRENT	Account Number	Available	Balance
ANDREAS MICHAEL	351012345671|EUR	50000,00	50000,00

To

Account Number
48193222324233

SWIFT/BIC
CITIUS33

Payment Information

Amount 30,00	Currency EUR
Charges Transfer_charges	Charges From 10,00
Execution Date 05/09/2024	Value Date 05/09/2024
Details SWIFT Transfer	

Disclaimer

1. Please be informed that the final execution date may differ from the selected execution date in cases where the transaction cannot be executed on that selected execution date due to technical, financial and/or other reasons.
2. Transfers submitted on business days before the applicable cut-off time will be considered to have been received by the Bank on that day. Transfers submitted after the applicable cut-off time or on non-business days will be considered to have been received on the following business day.
3. For Transfers with the same day value please contact your servicing branch for information regarding cut-off times applicable for each currency.
4. Transfers to Other Banks are allowed provided that these fall within the customers normal business. For compliance and legal reasons, for Transfers to Other Banks over EUR 50.000, it is mandatory to submit the relevant supporting documentation. Non submission of such documents, may result in a delay in execution or no execution.
5. For Transfers to Other Banks, depending on the currency and destination, different intermediary banks may be used. These banks, may deduct part of the transfer amount as commission, which Bank of Cyprus cannot estimate upfront.
6. For IBAN converter tool please click here
7. Actual transaction fee amount for a Basic Payment Account may differ from displayed fee amount.

After you review the payment and provide your consent the user will be redirected back to your app’s redirect URL with the authorization code as a query parameter as seen below.

Example Response

https://localhost/?code= AAIB7CW9gOFNa6c3B5l9GsfSHT

4. Get second Access token for the PATCH subscription

Example Request

```
curl —-location --request POST https://sandbox-apis.bankofcyprus.com/df-boc-org-sb/sb/psd2/oauth2/token \

--header accept: application/json \

--header content-type: application/x-www-form-urlencoded \

--data
grant_type=authorization_code&client_id={{client_id}}&client_secret={(client_secret}}&code={{code_from_redirect}}&scope=UserOAuth2Security
```

Example Response

```
{
"token_type": "bearer",
"access_token": "6334262361656275",
"expires_in": 3600,
"consented_on": 1542264419,
"scope": "UserOAuth2Security",
}
```

5. Execute the payment

The access token will be used to execute the payment and complete the flow.

Example Request

```
curl --location --request POST "https://sandbox-apis.bankofcyprus.com/df-boc-org-sb/sb/psd2/v1/payments/{{paymentId}}/execute’ \
--header "Content-Type: application/json" \
--header "Authorization: Bearer {{oauth_code}} \
--header 'journeyId: {{uuid}}' \
--header ‘timestamp: {{$timestamp}}’
```

Example Response

```
{
  "code": "CPLT",
  "description": [
    "The transaction has passed all validations and was successfully posted in bank systems"
  ],
  "refNumber": "12345678"
}
```

The transaction has status CPLT.