Open Bank Project Guide Documentation

Release 0.0.1

Open Bank Project

Apr 05, 2021

Contents:

Open Bank Project (OBP) is an open source RESTful API platform for banks that supports Open Banking, XS2A and PSD2 through access to accounts, transactions, counterparties, payments, entitlements and metadata - plus a host of internal banking and management APIs.

CHAPTER 1

Understand the basics

Learn about Open Bank Project and its fundamental concepts.

CHAPTER 2

Try Open Bank Project

Follow tutorials to learn how to use Open Bank Project.

- Experiment with Open Bank Project API and API Explorer

CHAPTER 3

Deploy Open Bank Project

Get Open Bank Project running based on your deployment needs.

CHAPTER 4

Contents

4.1 What is the Open Bank Project

Open Bank Project is an open source API for banks that provides a RESTful interface for developers to build customer facing applications without needing to code for each bank or account type differently.

4.1.1 What sort of applications can I build with the Open Bank Project API?

- Customer facing retail banking and fintech applications for consumers, SMEs, associations, charities, governments and NGOs etc
- Personal Finance Management (PFM) Solutions
- Online accounting integrations
- Financial widgets, Savings Apps, Education Apps, Gamification
- Peace of Mind Apps, Transparency Apps, Crowd funding, on boarding, CRM etc..

4.1.2 What data and services can I access?

An Open Bank Project instance can contain simulated customer related data. With real bank integrations, this can be real account data.

4.1.3 APIs include

- Account information, balance and transaction history of multiple bank accounts
- Initiate payments
- Onboard Customers (KYC etc.)

- Enrich bank transactions with metadata (tags, comments, urls and geolocation) for example to link a receipt or video to a transaction
- Create/Access different views on accounts. Each view grants a subset of the data to a restricted group of users. For example, a customer could offer special views on his account to his accountants, auditors or regulators. A charity might open their accounts to the public

4.2 Videos & Talks

Videos and talks about the Open Bank Project

Social Channels:

- Youtube
- Twitter
- LinkedIn

4.2.1 OBP CLI

Contributing to obp cli developerment

- Setting up your development environment
- Developing obp cli- how to contribue. Example adding command

4.3 OBP Components

4.3.1 OBP API

The API

4.3.2 API Explorer

API Explorer

4.3.3 API Manager

API Manager

4.3.4 OBP CLI

API CLI

4.4 API

4.5 Tasks

This section of the Open Bank Project documentation contains pages that show how to do individual tasks. A task page shows how to do a single thing, typically by giving a short sequence of steps.

4.5.1 Install obp cli

Note: The cli requires you use python version 3 or higher. On some systems this means your pip is called *pip3* rather than *pip*

```
pip3 install --user obp-python # Requires at least python 3
```

Verify installation

```
obp --help
```

Expected output:

```
Usage: obp [OPTIONS] COMMAND [ARGS]...

Options:
  --help  Show this message and exit.

Commands:
  addaccount               Add a bank account
  addbank                  Add a bank
  addcustomer              Add a customer
  addfx                    Add exchange rate (FX)
  addrole                  Add a role for current user
  adduser                  Add a user
  addview                  Add a view
  answerconsent            Answer consent
  createconsent            Add a consent
  deletebranches            Delete all branches
  deletecardbyid            Delete card by id
  getaccountbyid           Get account by id (includes balance)
  getaccountsheld          Get list of accounts held
  getaccounttransactions   Get transactions for an account
  getauth                  Get your DirectLogin token
  getbanks                 Get list of banks
  getcardbyid              Get card by id
  getcardbynumber          Get card by card number
  getcards                 Get list of cards at bank
  getconsents              Get consents
  getconsentstatus         Get consent status- with certificate
  getcustomers             Get list of customers
```

(continues on next page)

(continued from previous page)

```
  getuser                  Get your user info
  getuserid                Get your user id
  getuseridbyusername      Get user id by username
  getviews                 Get views by provider
  importaccounts           Import accounts from spreadsheet template
  importbranches           Import branches from spreadsheet template
  importcardattribues      Import card attributes from spreadsheet template
  importcards              Import cards from spreadsheet template
  importcustomers          Import customers from spreadsheet template
  importtransactions       Import transactions from spreadsheet template
  importusers              Import users from spreadsheet template
  init                     Initalize connection to your Open Bank Project...
  linkusertocustomer       Link user to a customer
  revokeconsent            Revoke consent
  sandboximport            Bulk import sandbox data from json input
```

Congratulations! If you see the help menu output, you have installed the obp cli.

Note: You can also view additional help on individual commands, such as `obp addrole --help` to get more help on the `addrole` command. For example:

```
obp addrole --help
Usage: obp addrole [OPTIONS]

 Add a role for current user

Options:
  --role-name TEXT  Name of the role/entitelment  [required]
  --bank-id TEXT    Some roles need a bank id
  --user-id TEXT    Add role to a differnt user
  --help            Show this message and exit.
```

4.5.2 Initalise obp cli

To use the obp cli against an Open Bank Project instance, you must first initalise it against the url of the instance you wish to access.

Prerequisites:

You must already have:

- *Install obp cli*

How to initialise your connection

Type `obp init`, you will then be asked to enter your Open Bank Project URL, username, password and consumer key (see task_get-consumer-key).

```
obp init
Please enter your API_HOST:  [https://api.example.com]: https://api.example.com
Please enter your username:  [chris]: chris
```

(continues on next page)

(continued from previous page)

```
Please enter your password:
Repeat for confirmation:
... generating direct login token
Please enter your OBP_CONSUMER_KEY: abc123
Init complete
```

Once you see `Init complete`, your obp cli has been successfully initalised! Verify your connection by running a simple command:

```
obp getuser
{"user_id":"uuid-hash","email":"chris@example.com","provider_id":"chris","provider":
↪"https://apisandbox.openbankproject.com","username":"chris","entitlements":{"list
↪":[]},"views":{"list":[]}}
```

The response should be your user object. See *Verify installation* for a selection of possible commands.

4.5.3 Import dummy data via cli

Prerequisites

- *obp cli is installed*
- *obp cli is initalised*
- Permissions

You may not have the correct permissions to perform the imports you want. The API will return an error specifying the permissions you need if you try to import data you don't have permissions for.

Concepts

To import data using the CLI you download the template spreadsheets provided, enter in as many rows as you want records, and then pass the populated spredsheet to the cli.

Using Import Templates

The CLI provides spreadsheet templates which you populate with the data you wish you import. The purpose of this is to allow less technical users create a dataset which may easily be imported into the API.

Note: You must save your templates in the ".ods" file format. This is an open document standard, and the only format that the `obp` cli currently supports. Once you've populated your template with your own data, be sure to save it as `.ods` format.

Locating import templates

The Open Bank Project provides import templates. Use these examples and edit them by adding rows of data for your own desired dataset.

The templates may currently be downloaded from the OBP-CLI repo.

Locate all the files ending in *.ods*. For example, at the time of writing, the users import template is called `users-import-template-v1.ods` from the repo.

> **Warning:** Be sure to check the latest master or release branch for the latest import templates.

Steps:

1. Take a template spreadsheet, for example "users.ods" and populate it with a list of users.
2. Identify the correct CLI import command, for example for users import the command is: `obp importusers users-import-template-v1.ods` press enter
3. If you have the correct permissions, and the data is valid, your import will succeed

If you don't have the correct permissions to create the object, then you may be able to use the cli to add roles for yourself, or request a higher privileged user grants them to you. See `obp addrole --help`.

4.6 Authentication

Consumer Keys

A Consumer Key allows you to register multiple applications under one account.

- Create an account
- Register a new application to generate your consumer key

`Open Bank Project` offers multiple authentication methods:

- OAuth 1.0a
- Direct Login

4.6.1 Direct Login

First you must create an account on Open Bank Project. Then, register a new application which gives you a *consumer-key*. You use your consumer key when generating a Direct Login token. http

```
POST /my/logins/direct HTTP/1.1
Host: api.openbankproject.com
Accept: application/json
Authorization: DirectLogin username="username", password="password", consumer_key=
↪"yourConsumerKey"
```

curl

```
curl -i -X POST https://api.openbankproject.com/my/logins/direct -H 'Accept:␣
↪application/json' -H 'Authorization: DirectLogin username="username", password=
↪"password", consumer_key="yourConsumerKey"'
```

wget

```
wget -S -O- https://api.openbankproject.com/my/logins/direct --header='Accept:␣
↪application/json' --header='Authorization: DirectLogin username="username",␣
↪password="password", consumer_key="yourConsumerKey"'
```

httpie

```
http POST https://api.openbankproject.com/my/logins/direct Accept:application/json␣
↪Authorization:'DirectLogin username="username", password="password", consumer_key=
↪"yourConsumerKey"'
```

python-requests

```
requests.post('https://api.openbankproject.com/my/logins/direct', headers={
    'Accept': 'application/json',
    'Authorization': 'DirectLogin username="username", password="password", consumer_
↪key="yourConsumerKey"',
})
```

response

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "token": "abc123"
}
```

4.6.2 Verify Authentication

You then use the token recieved from your DirectLogin request.

For example, make an authenticated request using your token.

Get your current user infomation: http

```
POST /obp/v3.1.0/users/current HTTP/1.1
Host: YOUR-HOST
Accept: application/json
Authorization: DirectLogin token="abc123"
```

curl

```
curl -i -X POST https://YOUR-HOST/obp/v3.1.0/users/current -H 'Accept: application/
↪json' -H 'Authorization: DirectLogin token="abc123"'
```

wget

```
wget -S -O- https://YOUR-HOST/obp/v3.1.0/users/current --header='Accept: application/
↪json' --header='Authorization: DirectLogin token="abc123"'
```

httpie

```
http POST https://YOUR-HOST/obp/v3.1.0/users/current Accept:application/json␣
↪Authorization:'DirectLogin token="abc123"'
```

python-requests

```
requests.post('https://YOUR-HOST/obp/v3.1.0/users/current', headers={
    'Accept': 'application/json',
    'Authorization': 'DirectLogin token="abc123"',
})
```

response

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "user_id":"2ef35575-aae9-48fb-ad01-751755b3964f",
  "email":"Fred@example.com",
  "provider_id":"your-provider-id",
  "provider":"your-provider-name",
  "username":"fred",
  "entitlements":{"list":[]}
}
```

4.7 Adapters

Connecting Open Bank Project to a core bank system required an adapter to be written.

Typically, this will involve the development of a Kafka consumer, which reads api requests sent to Open Bank Project, and then the adapter responds by putting a reply message into a corresponding topic.

4.7.1 Akka Adapter

Use Akka as an interface between OBP and your Core Banking System (CBS)

For an introduction to Akka see Akka Docs.

The OBP Akka interface allows integrators to write Java or Scala Adapters (any JVM language with Akka support) respond to requests for data and services from OBP.

For the message definitions see [here](/message-docs?connector=akka_vDec2018)

Installation Prerequisites

- You have OBP-API running.
- Ideally you have API Explorer running (the application serving this page) but its not necessary - you could use any other REST client.
- You might want to also run API Manager as it makes it easier to grant yourself roles, but its not necessary - you could use the API Explorer / any REST client instead.

Akka Messages

obp.getAdapterInfo

Gets information about the active general (non bank specific) Adapter that is responding to messages sent by OBP.

To query the most up to date message for your instance: http

```
POST /obp/v3.1.0/users/current HTTP/1.1
Host: YOUR-HOST
Accept: application/json
Authorization: DirectLogin token="abc123"
```

curl

```
curl -i -X POST https://YOUR-HOST/obp/v3.1.0/users/current -H 'Accept: application/
↪json' -H 'Authorization: DirectLogin token="abc123"'
```

wget

```
wget -S -O- https://YOUR-HOST/obp/v3.1.0/users/current --header='Accept: application/
↪json' --header='Authorization: DirectLogin token="abc123"'
```

httpie

```
http POST https://YOUR-HOST/obp/v3.1.0/users/current Accept:application/json␣
↪Authorization:'DirectLogin token="abc123"'
```

python-requests

```
requests.post('https://YOUR-HOST/obp/v3.1.0/users/current', headers={
    'Accept': 'application/json',
    'Authorization': 'DirectLogin token="abc123"',
})
```

response

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "user_id":"2ef35575-aae9-48fb-ad01-751755b3964f",
  "email":"Fred@example.com",
  "provider_id":"your-provider-id",
  "provider":"your-provider-name",
  "username":"fred",
  "entitlements":{"list":[]}
}
```

Table 1: obp.getAdapterInfo

Kafka/Akka	Outbound	Inbound
Topic	OutBoundGetAdapterInfo	InBoundGetAdapterInfo

Message	{
	"outboundAdapterCallContext":{
	"correlationId":"1flssoftxq0cr1nssr68u0mioj",
	"sessionId":"b4e0352a-9a0f-4bfa-b30b-9003aa467f50",
	"consumerId":"7uy8a7e4-6d02-40e3-a129-0b2bf89de8uh",
	"generalContext":[{
	"key":"5987953",
	"value":"FYIUYF6SUYFSD"
	}],
	"outboundAdapterAuthInfo":{
	"userId":"9ca9a7e4-6d02-40e3-a129-0b2bf89de9b1",
	"username":"felixsmith",
	"linkedCustomers":[{
	"customerId":"7uy8a7e4-6d02-40e3-a129-0b2bf89de8uh",
	"customerNumber":"5987953",
	"legalName":"Eveline Tripman"
	}],
	"userAuthContext":[{
	"key":"5987953",
	"value":"FYIUYF6SUYFSD"
	}],
	"authViews":[
	{
	"view":{
	"id":"owner",
	"name":
		{
		"inboundAdapterCallContext":{
		"correlationId":"1flssoftxq0cr1nssr68u0mioj",
		"sessionId":"b4e0352a-9a0f-4bfa-b30b-9003aa467f50",
		"generalContext":[{
		"key":"5987953",
		"value":"FYIUYF6SUYFSD"
		}]
		},
		"status":{
		"errorCode":"Status errorCode",
		"backendMessages":[{
		"source":"String",
		"status":"String",
		"errorCode":"String",
		"text":"String"
		}]
		},
		"data":{
		"errorCode":"",
		"backendMessages":[{
		"source":"String",
		"status":"String",
		"errorCode":"String",
		"text":"String"
		}],
		"name":"felixsmith",
		"version":"",
		"git_commit":"String",
		"date":"2017-09-19T02:31:05.000Z"
		}
		}

4.7.2 Kafka Adapter

The Kafka connector provides the following advantages when connecting OBP to a core banking system.

- It provides a logging layer for non repudiation. i.e. OBP sent this request to the core banking system at this time. The messages can be consumed into another (offsite) storge system.
- It provides a layer for real time analytics / fraud monitoring. i.e. Apache Spark or other tools could monitor the queue and look for irregularities.
- It provides a separation from the core OBP API. i.e. it's easier to use the OBP API develop branch when using the Kafka connector.
- Connector Code (south of the Kafka queue) is not restricted to a JVM language (Scala, Java, Clojure etc.) You can use any language that speaks Kafka e.g. Python, Go, C# etc.

```
   OBP
    |
North side connector (Scala or any JVM language)
  Kafka
South side connector (Any language)
    |
Core Banking
```

- genindex