Developer Documentation for Swedbank Open Banking Sandbox (BETA)

The purpose of this documentation is to help and guide developers around what is possible to access in terms of data regarding customers of Swedbank Group.

The contents of this sandbox will grow over time and potentially also change significantly from the early iterations which may result in breaking backward compatibility. The sandbox facilitates the learning and should help developers to become familiar with methods of accessing data within the Swedbank Open Banking Initiative.

As it is defined as a group service data may be queried from Sweden which includes Swedbank all the cooperating Savings Banks, Estonia, Latvia and Lithuania bank implementations. This also means that format and content may differ, one example is that Sweden does not use Euro as currency and other differences may also occur.
The API currently follows the Berlin Group specifications XS2A Interface Interoperability Framework and may be extended to cover further information, beyond what is covered by the PSD2 regulation.

Currently limited first versions of the targeted payment types are available. Everything is in early stages and will be subject to change over time until a stable release is defined and proper versioning applied.

The exposure of data is done through RESTful services and for the most part both requests and responses are in JavaScript Object Notation (JSON) format. In some cases for the Baltic services XML may be used, this is specifically indicated in the description of the applicable service.

We encourage you to provide feedback in order for us to improve our services by sending an mail to openbanking@swedbank.com

What is the difference between Open Banking and PSD2 services?

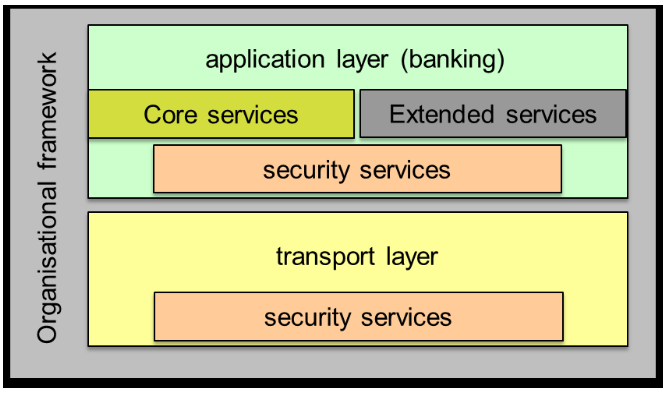
Swedbank Open Banking is an invitation to all developers to build new products and services based on a set of APIs. As an Open Banking Developer you are someone trying to innovate using financial data without being a regulated entity on the market. This will require a contract to access services provided and may also be subject to specific terms and conditions.

APIs offered as part of the PSD2 regulation is a subset of Open Banking. When PSD2 is applicable, as TPP (Third Party Payment Service Provider) within the PSD2 definition, you will be under supervision of the Financial Supervision Authority in your home member state in the European Union and with this comes a set of rights as well as a set of obligations.

API Overview

The API sandbox consists of **static mocked data** for

- Account Information Services (AIS) for Sweden only as defined by article 67 in the PSD2 Directive
- Payment Initiation services (PIS) for Sweden and Baltics as defined by Article 66 in the PSD2 Directive



The implemented services both fall in under core services in the above picture and is not covering all the requirements at this time.

Swedbank is following the Berlin Group API specification and where it is needed extensions are made to cover the full set of requirements needed for the service.

Please note: No security services or consent handling is available in this release, it is simply assumed that you are authorized to perform the operation.

The communication between the Bank and the TPP is secured using TLS 1.2 or higher. Later when the RTS is implemented additional checks against a TPP certificate will be implemented but at the time of this writing the requirements are not fixed.

The rest of the document will only describe the core services in the application layer.

API HTTP Methods

- GET - This method reads a resource and returns it. It returns 200 on success.
- POST - This method creates a new resource. It returns 201 on success.

The following table shows methods supported by the APIs.

API	GET	POST
AIS	X	-
PIS	X	X

API Responses and Response Codes

This API consumes JSON objects and returns responses as JSON objects and in some cases in payment initiation for the Baltic banks an ISO XML in the form of a pain.001, pain.002 and a camt.05x.

Whenever XML is used this is specifically pointed out in the examples section to see how the response objects look like. The responses are returned are encoded in UTF-8 format and will be in *snake_case*.

The objects which the API returns as a response are specified in the API reference and they all contain some attributes that are common to all objects. For instance, every object has return code embedded in them which makes it easy for application developers to check whether the request was successful or not.

Definitions

This section offers explanations to the terminology used throughout the document.

Third Party Provider

Third Party Provider(TPP) is the provider of an application which the user uses and is not offered by the bank. TPP is the client/consumer of the API and acts on behalf of the user through ***consent***.

Before the PSD2 deadline, for a bank customer to access their data via a TPP, following conditions must be met:

- TPP must have a valid agreement with Swedbank.
- The TPP must be enrolled on Swedbank's Open Banking platform and register the third party application before they can use the API.

This documentation is published and managed within the API-Explorer, make sure you use the latest version.

Sandbox

Sandbox gives access to a small set of static data is used as an example to illustrate what would be returned when using the live API. The sandbox can be reached within the developer portal at developer.swedbank.com.

API Call

API call is a request towards the API which receives a response. The API is by design stateless, it does not "remember" anything about previous requests, i.e. there is no session. Therefore every request made towards the API must contain certain headers so that the API can authenticate and authorize the use.

Message parameters can be passed at different levels;

- message parameters as part of the https level (https header)
- message parameters by defining a resource path (URL path information)
- message parameters as part of the https body

Authentication

Authentication is the process of verifying that an individual, entity or website is who it claims to be. This authentication is later used to grant authorization to specific data and functions within a system. SCA or Strong Customer Authentication is the process of using a strong (2-factor) identification method to identify the customer.

Authorization

Authorization is to validate an authenticated user against a defined access policy and will lead to granting or denying access to the data and functions defined in the policy.

Permission

Permissions are stored in an access policy and are part of the consent given by the user for the TPP. Permissions dictate what the TPP is allowed to do with the user data.

Consent

Consent is the agreement given by the user (PSU) to the bank to share data with a third party provider (TPP). A consent is stored by the bank and may have a duration or just be used for a single API call. The given consent will be available for the user to list and revoke within the bank services.

OAuth2

OAuth 2 is an authorization framework that enables applications to obtain limited access to user accounts on an HTTP service. Oauth2 is not implemented in the first iteration of the sandbox.

User

The user refers to the bank customer who uses the TPP application. (PSU)

Client

The client refers to the client of the API which is commonly the TPP application.

Swagger

An API design and documentation platform to aid in the development lifecycle. It is used to publish the documentation within the API-Explorer.

Account Information Services API

This is used to get information about accounts connected to a customer. A prerequisite for querying is that consent is collected already and consent information is passed in the request. In all our queries a BIC code is mandatory to specify in order to route the request to the correct backend. *Please note that Swedbank Sweden and cooperating Savings Banks share the same BIC.*

Browse API-Explorer for more information and refer to examples further down in this document.

The information can be categorized in following categories:

- Account list
- Balances
- Transaction history

Response Format

The AIS API response format is in JSON and looks like the following.

```
Response:     HTTP/1.1 200 OK
              Server: Apache-Coyote/1.1
              Content-Encoding: gzip
              Content-Type: application/json;charset=UTF-8
              Content-Length: 203
              Date: Fri, 13 Oct 2017 12:56:52 GMT

              {"account_list" : [
                {"id": "AsdF01234EfgH4567",
              "name":"privatkonto",
              "currency":"SEK",
              “product”:”privatkonto”,
              "account_type":"CACC ",
              "iban":"xxxxxxxx",
              “bic”:”SWEDSESS”,
              “clearingnumber”:”xxxx”,
              “accountnumber”:”xxxx”,
              },
              {"id": "AbcD1234eFgH568",
              "name":"Privatkonto",
              "currency":"SEK",
              “product”:”Privatkonto”
              "account_type":"CACC ",
              "iban":"xxxxxxx",
              “bic”:”SWEDSESS”,
              “clearingnumber”:”xxxx”,
              “accountnumber”:”xxxx”,
              }
              ]}
```

The Response header will show the http response code, encoding, content type as well as the creation time. The response contains the actual payload of the response, and the endpoint in question defines its structure.

Connecting to API

To be able to use and connect to the API there are few requirements. Since no security services are available in this release, the only things that is needed to be passed in the request header is Authorization with a value of Bearer followed by a string of characters (A-Z, a-z,0-9 and minus is allowed). This value will **NOT** be validated in the sandbox. In future releases this should be your Oauth2 token. Also process-id and request-id are mandatory headers throughout the API. Process-id must be a unique identifier for the transaction that is initiated and request-id should be unique per request. When connecting through Swagger and testing examples there Date header will automatically be added in the request but when using any other tool directly against the API you must supply the Date header in your request.

Applications

The TPP provides the application(s), and they are the clients of the API. The application can refer to a website, mobile application, etc. which uses the API. The resource owner (application user) grants the permission for the TPP application (consumer of the API) to use API resources. This permission is given by giving the consent after authentication.
In this version of the API it is simply assumed that this has happened previously and you are passing in a Bearer token which currently is not validated.

Error Codes and Responses

Every response returned by this API has a response code. Response codes can be used to check the result of the requests e.g. was the request successful or did it fail.
The following table shows the return codes used by AIS API.

HTTP response	Text	Description
200	OK	Request was fulfilled.
201	Created	The request has been fulfilled, resulting in the creation of a new resource.
302	Found	Redirect.
401	Unauthorized	Similar to 403 Forbidden, but specifically for use when authentication is required and has failed or has not yet been provided.
403	Forbidden	The request was valid, but the server is refusing action. The user might not have the necessary permissions for a resource.
404	Not Found	The requested resource could not be found but may be available in the future.
405	Invalid input	The input provided is invalid.

Please note that the underlying data for the account is same and static in all endpoints of the AIS API. However, some endpoints return more data from this model, e.g. account listing endpoint returns more account data compared to account details endpoint even though the underlying data model is the same.

API Examples AIS

Below you can find examples how to use the API endpoints.

Example: List Accounts
This example lists all payment accounts on the user.

This endpoint URL has the following form:

https://psd2.api.swedbank.com//sandbox/v1/accounts

HTTP Headers

- Authorization (mandatory) planned later releases for the Oauth2 token
- Process-ID
- Request-ID
- Date

HTTP Query parameters:

- bic (mandatory): one of SWEDSESS, HABAEE2x, HABALT22, HABALV22
  specifies if it's a Swedish (bic=SWEDSESS) or Baltic customer (bic=HABAEE2x, bic=HABALT22, HABALV22).
- with-balance (optional - boolean): [true/false]

This endpoint supports only GET requests.

The following cURL can be used to fetch account list from this endpoint.

```
curl "https://psd2.api.swedbank.com//sandbox/v1/accounts/?bic=SWEDSESS"
-H "Authorization Bearer ThisWillBeYourOauthToken123"
-H "Process-ID: AZXS3456"
-H "Request-ID: 12345SGHDF"
-H "Date: Thu, 01 Dec 1994 16:00:00 GMT"
```

Response:

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Encoding: gzip
Content-Type: application/json;charset=UTF-8
Content-Length: 203
Date: Fri, 13 Oct 2017 12:56:52 GMT
```

```
{"account_list": [
    {
```

```
    "id": "AsdF01234EfgH4567",
    "currency": "SEK",
    "product": "privatkonto",
    "account_type": "CACC",
    "iban": "SE4880000123459876543219",
    "bic": "SWEDSESS",
    "bban": "1234-5,987 654 321-9",
    "clearingnumber": "1234-5",
    "account_number": "987 654 321-9",
    "balances": [{"booked":      {
      "amount":        {
        "currency": "SEK",
        "content": 100
      },
      "date": "2017-11-02"
    }}]
  },
    {
    "id": "AbcD1234eFgH568",
    "currency": "SEK",
    "product": "ungdomskonto",
    "account_type": "CACC",
    "iban": "SE4880000123451234567890",
    "bic": "SWEDSESS",
    "bban": "1234-5,123 456 789-0",
    "clearingnumber": "1234-5",
    "account_number": "123 456 789-0",
    "balances": [{"booked":      {
      "amount":        {
        "currency": "SEK",
        "content": 35000
      },
      "date": "2017-11-02"
    }}]
  }
]}
```

Example: Fetch Account Information on specific account
In this example, we fetch account information by ACCOUNT-ID which can be found by listing the accounts.

The endpoint URL has the following form:
https://psd2.api.swedbank.com//sandbox/v1/accounts/{ACCOUNT-ID}

HTTP Headers

- Authorization (mandatory) planned later releases for the Oauth2 token
- Process-ID
- Request-ID

- Date

HTTP Query parameters:

- bic (mandatory): one of SWEDSESS, HABAEE2x, HABALT22, HABALV22
  specifies if it’s a Swedish (bic=SWEDSESS) or Baltic customer (bic=HABAEE2x, bic=HABALT22, HABALV22).
- with-balance (optional - boolean): [true/false]

This endpoint supports only GET HTTP method.

Response:

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Encoding: gzip
Content-Type: application/json;charset=UTF-8
Content-Length: 203
Date: Fri, 13 Oct 2017 13:06:16 GMT


{
  "id": "AbcD1234eFgH568",
  "currency": "SEK",
  "product": "ungdomskonto",
  "account_type": "CACC",
  "iban": "SE4880000123451234567890",
  "bic": "SWEDSESS",
  "bban": "1234-5,123 456 789-0",
  "clearingnumber": "1234-5",
  "account_number": "123 456 789-0",
  "balances": [{"booked":   {
    "amount":     {
      "currency": "SEK",
      "content": 35000
    },
    "date": "2017-11-02"
  }}]
}
```

Example: Fetch Account Information on specific account

This is how it looks if you query an unsupported BIC.

```
GET https://psd2.api.swedbank.com//sandbox/v1/accounts/AsdF01234EfgH4567/?bic=HABALT22
Authorization: Bearer YourOauthTokenGoesHere1234
Response:      HTTP/1.1 404 Not Found
               Server: Apache-Coyote/1.1
               Content-Type: text/plain;charset=UTF-8
               Content-Length: 70
               Date: Fri, 13 Oct 2017 13:15:13 GMT
               Connection: close
```

```
{"tpp_messages" : [{"category" : "ERROR",
"code" : "NOT_IMPLEMENT"}]}
```

Example: Read Transaction list

The following request will fetch transactions from the account by ACCOUNT-ID. Note that this query also requires parameters DateFrom and DateTo.

These parameters mentioned above control the time range from between which to fetch the transactions, and they are mandatory. They are sent as query parameters.

The endpoint URL has the following form:

```
https://psd2.api.swedbank.com//sandbox/v1/accounts/{ACCOUNT-ID}/transactions
```

This endpoint supports only GET method.

The **{ACCOUNT-ID}** URL parameter(s) can be found from account listing request above.

HTTP Headers

- Authorization (mandatory) planned later releases for the Oauth2 token
- Process-ID
- Request-ID
- Date

HTTP Query parameters:

- bic (mandatory): one of SWEDSESS, HABAEE2x, HABALT22, HABALV22
  specifies if it's a Swedish (bic=SWEDSESS) or Baltic customer (bic=HABAEE2x, bic=HABALT22, HABALV22).
- with-balance (optional - boolean): [true/false]
- date_from
- date_to

Response:

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Encoding: gzip
Content-Type: application/json;charset=UTF-8
Content-Length: 1105
Date: Fri, 13 Oct 2017 13:06:16 GMT


{
  "transactions": {
    "booked": [
      {
        "credit_debit": "Debited",
        "amount" : {"currency" : “SEK”, content: 343 } ,
        "booking_date": "2017-10-30",
        "transaction_date": "2017-10-28",
        "value_date": "2017-10-30",
        "remittance_information": "Fackavgift",
        "balances": {
          "booked": {
            "amount": {
              "currency": "SEK",
              "content": 10000
            },
            "date": "2017-11-02"
          }
        }
      },
      {
        "credit_debit": "Credited",
        "amount" : {"currency" : “SEK”, content: 2365 } ,
        "booking_date": "2017-10-25",
        "transaction_date": "2017-10-24",
        "value_date": "2017-10-25",
        "remittance_information": "Löneinkomst",
        "balances": {
          "booked": {
            "amount": {
              "currency": "SEK",
              "content": 35000
            },
            "date": "2017-11-02"
          }
        }
      }
    ]
  }
}
```

Payment Initiation Services API

This is used to submit a payment initiation request on behalf of a customer. It is assumed that the account information has been obtained from the customer in some way prior to the request most commonly through a query of the list accounts request. In all our queries a BIC code is mandatory to specify in order to route the request to the correct backend. See examples for more information.

API Examples PIS

Below you can find examples how to use the API endpoints.

Example: Initiate SEPA payment with pain.001 XML
This example intiates a SEPA payment using a pain.001 XML. This is only valid for the Baltic BIC's. This endpoint supports only POST requests.

This endpoint URL has the following form:

```
https://psd2.api.swedbank.com//sandbox/v1/payments/pain.001-sepa-credit-transfers
```

HTTP Headers

- Authorization (mandatory) planned later releases for the Oauth2 token
- Process-ID
- Request-ID
- Date

HTTP parameters:

- bic (mandatory): one of HABAEE2x, HABALT22, HABALV22
  specifies Baltic customer (bic=HABAEE2x, bic=HABALT22, HABALV22).

Currently we do not validate that the pain.001 conforms to the schema.

Example: Initiate Swedish domestic payment

This example intiates a Swedish payment in SEK and allows the use of BBAN, IBAN or Swedish Bank or PostGiro payments. Only SWEDSESS is allowed as BIC. This endpoint supports only POST requests. The only supported Content-Type is application/json.

This endpoint URL has the following form:

https://psd2.api.swedbank.com//sandbox/v1/payments/se-domestic-ct

HTTP Headers

- Authorization (mandatory) planned later releases for the Oauth2 token
- Process-ID
- Request-ID
- Date

HTTP parameters:

- bic (mandatory): one of SWEDSESS, HABAEE2x, HABALT22, HABALV22
  specifies if it’s a Swedish (bic=SWEDSESS) or Baltic customer (bic=HABAEE2x, bic=HABALT22, HABALV22).

Example with IBAN

```
{
"instructed_amount" : {"currency" : "SEK" , "content" : 123},
"debtor_account" : { "iban":"SE4880000123459876543219"},
"creditor_account": {"iban":"SE4880000123459876543219"},
"remittance_information_unstructured" : "Ref Number Mottagare-1234567890"
}
```

Example with BBAN

```
{
"instructed_amount" : {"currency" : "SEK" , "content" : 123},
"debtor_account" : { "bban":"1234-5,987 654 321-9"},
"creditor_account": {"bban":"1234-5,987 654 321-9"},
"remittance_information_unstructured" : "Ref Number Mottagare-1234567890"
}
```

Example BG account BBAN

```
{
"instructed_amount" : {"currency" : "SEK" , "content" : 123},
"debtor_account" : { "bban":"1234-5,987 654 321-9"},
"creditor_account": {"bban":"BG 2345-6789"},
"remittance_information_unstructured" : "OCR Number 1234567890"
}
```

Common Issues

In this section, we will collect the common issues that the users of the API face. This section will be updated over time.

Date header

When connecting through Swagger and testing examples there Date header will automatically be added in the request but when using any other tool directly against the API you must supply the Date header in your request.