De Volksbank PSD2 Sandbox

Version 14 – November 14 2022

Change log

V5	In this version we have made some textual changes and added new endpoints: GET Sandbox AIS GetConsent for AIS, and the endpoints for CAF (Confirmation of the Availability of Funds): POST Sandbox CAF initiateConsent and POST Sandbox CAF FundsConfirmation.
V6	For V6, the one-time agended payments have been added (named *future dated payments* by the Berlin Group) as well as the payment cancellation functionality.
V8	In this version, the initiatePeriodicPayment endpoint has been added.
V9	In this version, the get payment endpoints has been added. Also a quality of life improvement is made for handling access and refresh tokens.
V10	The new getPaymentStatus v1.1 endpoints for one-time direct, one-time agended and deferred payments have been added, as well as the initiateBulkPayment endpoint. In addition, this document has been improved regarding structure, the API description provides a clearer explanation of the flows and Portal APIs we offer, the authorization flow has been described in a more hands-on manner, and the error scenarios have been updated.
V11	The AIS v1.1 endpoints (read account list, read balance, read transactions list) have been added, and the screenshots of the Postman tests have been removed.
V12	In this version, the cancel bulk payment endpoint is added and a paymentId for bulk payment calls is provided. Additionally, the existing cancel payment call now also requires a specific paymentId, and the getRecurringPaymentStatus v1.1 has been added to the Sandbox.
V13	This version removes the AIS v1.0 endpoints (read account list, read balance, read transactions list) and getPaymentStatus v1.0 endpoints, and updates the Postman files to reflect it.
V14	In this version, the CAF endpoint has been updated, and functionality for managing a CAF consent has been added. The consentIds for AIS have been updated (including the consentIds for error scenario testing). They are now UUIDs and as such better represent the new consentId format for the AIS API.

Table of Contents

1. Introduction

This document describes the test environment that de Volksbank offers for testing its PSD2 Open Banking APIs inside a 'sandbox' environment.

The production environment that enables the PSD2 Open Banking functionality is complex. It consists of many different systems in which each system takes care of a portion of the business logic.

The Sandbox looks identical to the production environment, but instead of returning live data (or live error responses) it returns static data. This does not mean that it always returns exactly the same data for each invocation: based on the given input, in particular the consentId or paymentId, a different response (including error responses) may be returned.

The Sandbox environment enables you to develop and test your application:

- It simulates all interactions with the Open Banking APIs of de Volksbank, similar to the production environment;
- It allows you to fully test the OAuth2 process without needing an actual de Volksbank account. This includes the interaction with our production Web Service Gateway (WSG), which requires a valid client certificate (see https://openbanking.devolksbank.nl/apis.html);
- It simulates specific error scenarios.

2. Get started

To give you a kick-start, we provided a zip file that contains three Postman collections and a Postman environment file. These can be used to test all available Authorize and PSD2 AIS, PIS and CAF APIs. The Postman files contain the correct URLs for the various endpoints and provide all the (required) header arguments and bodies (where applicable). The files can be found at https://openbanking.devolksbank.nl/documentation.html.

Inside *sandbox-devolksbank.postman_environment.json* there are a number of environment variable values that start with "*your-*". You should replace these with a valid value. In particular, the following environment variables are critical: *clientId, clientSecret, redirectUrl, accessToken, consentId, paymentId* and *resourceId.* The first three variables are fixed and are part of the initial sign up process. The latter variables are returned in the form of JSON response messages by various requests.

The flows that are available in the Sandbox are explained in more detail in the next section.

3. Application flows

The Sandbox offers three application flows; one for AIS, one for PIS and one for CAF. The Sandbox flows mimic the production flows, which are described in detail in the documentation files 'API AIS', 'API PIS' and 'API CAF' on our Open Banking website: https://openbanking.devolksbank.nl/documentation.html. For more information about the calls, like the required fields, please have a look at the documentation presented there.

3.1 AIS

With the AIS calls provided in the Postman file, you can simulate the full AIS flow. The first call is the initiate call, for initiating a consent. For this, use the Sandbox SNS Bank Consent Services API on the Developer Portal.

Next are the authorize steps, which are described in more detail below. You use this to retrieve an authorization code of the PSU and to exchange this code for access and refresh tokens.

After simulating the authorization, you can test our AIS services. You can simulate calls for obtaining information about or managing the consent, or for retrieving account information:

- Request the consent status using the Sandbox SNS Bank Consent Status Services API;
- Request the details of a consent or delete a consent using the Sandbox SNS Bank Manage Consent Services API;
- Retrieve account information (accounts, balances and transactions) using the Sandbox SNS Bank Account Information Services v1.1 API.

You will need the retrieved access token for most of these calls, except for requesting the consent status, for which you only need a clientId.

To get different status responses for a GetConsentStatus call, you can use different consentIds. These are not extensive, and serve to give an impression of the format of the response.

consentId	status
0ddb9b6c-6355-4aee-a5f1-d5493b70cc2d	valid (standard response)
0ddb9b6c-6355-4aee-a5f1-d5493b70cc2e	revokedByPsu
0ddb9b6c-6355-4aee-a5f1-d5493b70cc2f	expired
0ddb9b6c-6355-4aee-a5f1-d5493b70cc2g	terminatedByTpp

The Sandbox GetTransactions endpoint can return two responses based on the input: a fixed (meaning that query params will not influence the result) number of transactions, or a response indicating that no more transactions are available (indicated by the absence of a next page link). The latter can be obtained using the Postman request ‘Sandbox AIS GetTransactions (Last entry)’ or by using the next page link as provided in the GetTransactions v1.1 response.

3.2 PIS

With the PIS calls provided in the Postman file, you can simulate the full PIS flow. The first call is the initiate call, for initiating a payment. For this, use the Sandbox SNS Bank Payment Initiation Services API on the Developer Portal. This API supports initiating calls for one-time direct, one-time agended, deferred, recurring, periodic and bulk payments.

Next are the authorize steps, which are described in more detail in the next chapter. You use this to retrieve an authorization code of the PSU and to exchange this code for access and refresh tokens.

After simulating the authorization, you can test our PIS services. You can simulate calls for obtaining information about or managing the payment, or for executing payments:

- Execute a deferred or recurring payment using the Sandbox SNS Bank Manage Payment Services API;
- Request the details of a one-time direct, one-time agended, deferred, recurring or periodic payment using the Sandbox SNS Bank Manage Payment Services API;
- Cancel a one-time agended payment or bulk payment using the Sandbox SNS Bank Payment Cancellation Services API;
- Request the transaction status using the Sandbox SNS Bank Payment Status Services v1.1 API (for one-time direct, one-time agended, deferred, recurring and bulk payments).

You will need the retrieved access token for most of these calls, except for cancelling a payment and requesting the transaction status using the v1.1 version, for which you only need a clientId.

To get the correct response for a GetPayment, GetPaymentStatus and CancelPayment call, you need to use the payment-type specific paymentIds which are also included in the Postman environment file. These are:

Payment type	paymentId
One-time direct	SNS1375722387857
One-time agended	SNS1313131313021
Deferred	SNS1370294778604
Recurring	SNS7020012812341
Periodic	SNS7020012812928
Bulk	SNS1370294778604

3.3 CAF

With the CAF calls provided in the Postman file, you can simulate the full CAF flow. The first call is the initiate call, for initiating a CAF consent. For this, use the Sandbox SNS Bank Funds Confirmation Consent Services API on the Developer Portal.

Next are the authorize steps, which are described in more detail below. You use this to retrieve an authorization code of the PSU and to exchange this code for access and refresh tokens.

After simulating the authorization, you can test our CAF services. You can simulate the call for requesting a confirmation of funds using the Sandbox SNS Bank Funds Confirmation Services, and you can manage your CAF consent:

- Request the consent status using the Sandbox SNS Bank Consent Status Services API;
- Request the details of a consent or delete a consent using the Sandbox SNS Bank Manage Consent Services API.

You will need the retrieved access token for most of these calls, except for requesting the consent status, for which you only need a clientId.

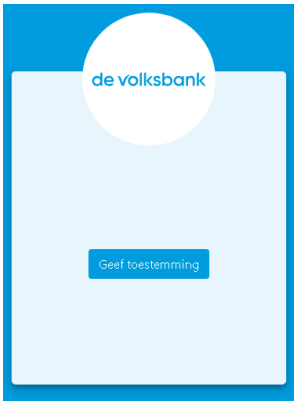
4. Authorization flow

Within the Sandbox, the flow as described below is used to simulate the process for authorizing a TPP to access data of a customer of de Volksbank. You will need to follow this flow in order to obtain the access (and refresh) token needed as authorization for most calls.

Note that the authorization flow described below is not part of the Berlin Group API, and cannot be found in a Swagger file on the Developer Portal. This is a de Volksbank-specific API.

The best way to simulate the authorize process is to use the Postman files we offer. The flow is as follows:

1. After initiating an AIS consent, PIS payment or CAF consent, send in the call as provided in the Postman request 'Sandbox Authorize'.
2. This call will return status '302 Found' and several header fields. Copy the value of the header field 'Location' and paste this in your browser.
3. You will be taken to the page which simulates the production login and consent flows. Note that it is only a very simple page meant to obtain the code you need for requesting the access token. In production, de Volksbank customer is redirected to the login page, where they need to log in, select an account and finally grant access. In the Sandbox the following screen is presented:



4. On this page, click the button 'Geef toestemming'.
5. You will be redirected to a new page. You need the information that is present in its URL, specifically the value of 'code'. Copy this value.
6. Return to Postman and in the 'Sandbox Exchange token' request, paste the value you just copied in the URL as the value for 'code'.
7. Send the request. In the response you will find your access and refresh token. You will need to use the clientId and clientSecret that you received from the Volksbank for this call.
8. The access token is valid for 10 minutes. After 10 minutes this access token cannot be used anymore. With the refresh token that was returned, a new access token can be obtained with the Postman request 'Sandbox Refresh token'.

5. Error scenarios

The Sandbox features validations on input fields (including header and request parameters) which are also present in the production environment.

In addition, some specific error scenarios can be simulated using specific consentIds in the AIS flow. These are not extensive, and serve to give an impression of the format of the error responses.

consentId	Error scenario
0ddb9b6c-6355-4aee-a5f1-a0000b11cc2a	The mandate could not be found.
0ddb9b6c-6355-4aee-a5f1-a0000b11cc2b	The expiration date of the mandate has been expired.
0ddb9b6c-6355-4aee-a5f1-a0000b11cc2c	The mandate is revoked.