

iDEAL 2.0 Getting Started Guide

Status	Published
Author:	Worldline
Document date:	19 February 2024
Classification:	Confidential
Version:	1.4

Version history

Version no.	Version date	Status	Edited by	Most important edit(s)
1.0	18-7-2022	Published	Worldline	Initial Version
1.1	19-12-2022	Published	Worldline	Minor Edits to Section 2.2.3 and Section 2.3.2
1.2	13-03-2023	Published	Worldline	Added Chapter on Production
1.3	28-04-2023	Published	Worldline	Added example under Test Steps Amended Section 2.3.3 and 2.3.4 on Notification Tokens.
1.4	19-2-2024	Published	Worldline	Separate Acquirer Profiles Testing Debtor token callback requests iDEAL FAQ

Copyright © Worldline. All rights reserved.

Worldline is a registered trademark of Worldline SA. © 2024 Worldline.

Confidential information owned by Worldline, to be used by the recipient only. This document, or any part of it, may not be reproduced, copied, circulated and/or distributed nor quoted without prior written approval from Worldline.

Table of contents

Confidential

1	First Steps	5
1.1	What are the Open Banking APIs?	5
1.2	What should you start with?	5
1.3	What functionality is available in the test environment?	5
2	Set up Test Environment	6
2.1	Endpoints for Testing	6
2.2	Test Data	7
2.2.1	Test Certificates and Private Keys	7
2.2.2	Payment Status Simulation	7
2.3	Test Steps for Initiating an iDEAL 2.0 Payment	8
2.3.1	Step 1 - Request an Access Token via Authorization Request.....	8
2.3.2	Step 2 - Initiate an iDEAL 2.0 Payment.....	8
2.3.3	Step 3 : Payment Finalisation	9
3	Moving to Production	11
3.1	Productive Environment Setup.....	11
3.2	Productive Transactions	11
4	iDEAL 2.0 FAQ	12
4.1	General.....	12
4.2	Authorization Requests.....	13
4.3	Payment Initiation	13
4.4	Payment Finalization	14
5	Acquirer Profile	15
5.1	BNG Bank.....	15
5.1.1	Signatures	15
5.1.2	Authorization Requests.....	15
5.1.3	Test Merchants.....	16
5.1.4	Transactions in Production	16

First Steps

Please switch to [First Steps](#) to learn

- what are the Open Banking APIs,
- what you should start with,
- which functionalities are available in the test environment

Set up Test Environment

Please switch to [Set up Test Environment](#) to learn

- how to access the test environment,
- which steps have to be taken for initiating an iDEAL 2.0 payment.

Moving to Production

Please switch to [Moving to Production](#) to learn

- which preconditions have to be fulfilled and
- how the Productive Environment Setup has to be done.

iDEAL FAQ

Please switch to [iDEAL FAQ](#) to answer frequently asked questions

Acquirer Profile

Please switch to [Acquirer Profile](#) to learn :

- what Merchants to use in test environment
- how to fill the headers in the Authorization API

1 First Steps

1.1 What are the Open Banking APIs?

The Open Banking API are fully fledged RESTful APIs enabling the initiation of iDEAL 2.0 payments. Developer tools like [Postman](#) or [SOAP UI](#) can be used to test the Open Banking API in the test environment.

More information on the APIs can be found in the **Open Banking API v3 for iDEAL - Implementation Guide**.

1.2 What should you start with?

We recommend starting with discovering the overall iDEAL 2.0-related functionalities with the Open Banking APIs by reading the **Open Banking API v3 for iDEAL - Implementation Guide** and trying out the different functionalities and APIs in the test environment.

1.3 What functionality is available in the test environment?

Our test environment enables you to test the integration with our Open Banking APIs for iDEAL 2.0. It contains a simulator for the iDEAL Hub and provides simulated API responses. All data in the test environment is mocked and is to be used only for integration testing purposes.

The test environment for the Open Banking APIs supports the following iDEAL 2.0-related functionalities:

- Initiating payments.
- Retrieving debtor preferences (for payment with user recognition via debtor tokens),
- Retrieving status information about payments
- Receiving payment notifications.

The test environment for the Open Banking APIs does not support the following functionality:

- User/consumer interaction pages for issuer selection and profile registration

The iDEAL Hub pay page is handled by Currence, the scheme owner. There is no test environment provided by Currence for Merchants at the moment. Therefore, the test environment does not send test transactions to an actual functional URL.

A dummy response will be returned. To simulate transactions, Merchants should use the payment amounts mentioned below to simulate different scenarios.

2 Set up Test Environment

2.1 Endpoints for Testing

The test environment can be reached under following **base domain** : <https://digitalrouting.service.awltest.de>¹

You are not required to login and create an account on the weblink. The purpose of this base domain is for it to be used to create the full end points. The full endpoints for all APIs can be derived from the technical documentation (swagger files). The full endpoints are derived in the following manner :

Base Domain + Base URL + Endpoint

Some relevant endpoints in the test environment are as follows. Please note that this is a non exhaustive list. :

- Requesting for an access token via Authorization API

POST /token Generates a token for the Initiating Party

[Base URL: /xs2a/routing/service/services/authorize]

<https://digitalrouting.service.awltest.de/xs2a/routing/service/services/authorize/token>

- Initiating a Payment via Open Banking Payment Initiation Service

POST /payments Payment initiation request

/xs2a/routing/service/services/ob/pis/v3

<https://digitalrouting.service.awltest.de/xs2a/routing/service/services/ob/pis/v3/payments>

- Getting a Payment Status via Open Banking Payment Initiation Service

GET /payments/{paymentId}/status Status of the payment request

/xs2a/routing/service/services/ob/pis/v3

<https://digitalrouting.service.awltest.de/xs2a/routing/service/services/ob/pis/v3/payments/{paymentId}/status>

¹. If you have previously been using an Acquirer specific base domain, those URLs are still functional and will still work. We are transitioning to a generic URL which can be used by all Acquirers.

2.2 Test Data

In order to test the connectivity to the Open Banking APIs in the test environment, the usage of pre-defined test data is necessary:

- Test Certificates for Merchants/PSPs
- Preconfigured Merchants : Merchant A and Merchant B. The IDs for the Merchants are available in the Acquirer Specific Information pages.
- Specific payment amounts to simulate different payment statuses

2.2.1 Test Certificates and Private Keys

We have provided a test certificate that you can use to integrate with the Open Banking APIs in the test environment.

These certificates will give you the possibility to retrieve an access token as a mandatory pre-step to consuming the payment APIs.

The TestCertificatesIDEAL2.0 file contains both the private and public keys needed in order to perform the integration tests. Please find these certificates in the Documents section of the MSP Portal.

The Test Merchants

Test merchants have been set up on the test environment, in order to allow testing the integration with the Open Banking APIs.

The following test Merchants are available:

Merchant Type	Integration	Test Merchant
Merchant with no Sub Merchants	Self onboarding or Service Provider (dPSP)	Merchant A ²
Merchant/cPSP with Sub Merchants	Self onboarding or Service Provider (dPSP)	Merchant B ²

Distributing Service Providers (dPSPs) may also make use of the same preconfigured merchants for their integration tests.

² The IDs are available in the Acquirer Profile pages at the end of the document, along with some examples.

2.2.2 Payment Status Simulation

Different payment statuses can be simulated in the test environment by providing specific payment amounts during the payment initiation (in the POST payments API):

Payment amount in POST /payments	Payment status
1.00	SettlementCompleted
2.00	Cancelled
3.00	Expired
4.00	Open
5.00	Error

It is recommended that all statuses are tested once to ensure your technical integration responds to the statuses as expected.

2.3 Test Steps for Initiating an iDEAL 2.0 Payment

This section describes steps to connect to the Open Banking API for iDEAL in the test environment and execute iDEAL 2.0 payments.

It also describes the behavior of the test environment.

Full request-response examples as well as other detailed information on implementation can be found in the Open Banking Implementation Guide for iDEAL.

2.3.1 Step 1 - Request an Access Token via Authorization Request

Authenticate yourself as a Merchant by providing the correct ID and using the correct certificate to make an authorization API request. In the response, receive the access token which is valid for 60 minutes.

Note: In the productive environment, certificates generated by the Merchant/PSP and productive Merchant/PSP/Sub-Merchant Ids are used.

Please check the Acquirer Profile for examples on how to fill the Id, Client and APP headers. More information on creating the request can be found in the Open Banking API Implementation Guide for iDEAL.

2.3.2 Step 2 - Initiate an iDEAL 2.0 Payment

Use the access token retrieved in Step 1 to initiate an iDEAL 2.0 payment.

The necessary fields to use in the POST /payments API depend on the functionality desired. Three flows are possible for iDEAL and all three flows can be tested against the test environment.

- **Standard Flow**
- **Standard Flow with Debtor Token**
- **iDEAL Checkout Flow**

All flows are explained in detail in the Open Banking Implementation Guide for iDEAL.

Testing Best Practice

As the test environment only includes 2 test merchants, Acquirers are unable to know whether a Merchant has started their migration for iDEAL 2.0 or not. As a best practice, you may keep your Acquirer in the loop by adding your Legal name in the CommonPaymentData.RemittanceInformation.

Initiating a payment in the test environment will lead to a dummy Redirect URL of <https://worldline.com>.

```
"Links": {  
  "RedirectUrl": {  
    "Href": "https://worldline.com"  }  
}
```

This is NOT a functional URL. To check the success of a test transaction, testing should be done with the payment notifications based on the payment amounts listed in Section 2.3.3. e.g. sending an amount of 1,00 € to get back "SettlementCompleted".

2.3.2.1 Testing Standard Flow with Debtor Token

All other flows can be tested as is, according to the guidelines in the Open Banking Implementation Guide for iDEAL. To test a standard flow with debtor tokens, the test environment expects a payment amount of **8.00€**.

This triggers the test environment to send the debtor token callback notification request to the URL stated in the DebtorTokenCallbackUrl field in the POST /payments API request. More details are available in the Open Banking Implementation Guide for iDEAL.

2.3.3 Step 3 : Payment Finalisation

2.3.3.1 Step 3a - Receive Notifications

There are dummy values set in the iDEAL Subscription of the MSP Portal in the notification fields for the test Merchant. Most importantly, the Notification Bearer Token

Field	Value
Notification Bearer Token	iDEAL2.0testnotificationtoken

To make use of the Notification Service (Status Notifications and Debtortoken callback notifications), please follow the guidelines below.

Status Notification Requests

The 'InitiatingPartyNotificationURL' header is mandatory in the POST /payments API request in the test environment if you want to receive the Status Notifications. Please ensure this URL ends with /v3 as specified in the Swagger File.

InitiatingPartyNotificationUrl The URL which will be used for notification service request. The Open Banking Service supports two ways in which this field can be filled:

string
(header)

- **Option A** with an URL ending on /v2 or /v3, indicating the version of the Notification API implemented by the Initiating Party. In this case the URL called for notifications will be extended with /notification/status by the Open Banking Service. The 'NotificationVersion' field MUST NOT be filled.
- **Option B** with a full URL. The version information MUST BE provided in the 'NotificationVersion' field. In this case the provided URL will not be extended, and used as-is.

Examples:

Option A

https://www.example.org/myendpoint/v3

For production, the "Status Notification URL" and the "Notification Bearer Token" **must** be filled in the iDEAL Subscription of the MSP Portal.

The URL value can be overridden with the 'InitiatingPartyNotificationURL' header field in the POST /payments API request.

If the Status Notification URL and Notification BearerToken are not filled out in the iDEAL Subscription of the MSP Portal, no notifications will be sent out by the Open Banking Service. No error messages will be provided by the Open Banking Service.

DebtorToken Callback Notification Requests

The field 'IDEALPayments.DebtorTokenCallbackUrl' is mandatory in the POST /payments API request in the test environment when testing the Standard Flow with debtor token. Please ensure this URL also ends with /v3 as specified in the Swagger File.

```
IDEALPayments
  IDEALPayments {
    UseDebtorToken boolean
      default: false
      Identifies whether the debtor token generated by Access Point should be used

    DebtorTokenCallbackUrl string
      example: https://www.example.org/myendpoint/v3
      The URL which will be used by the notification service post /debtorToken API to send the
      debtor token provided by the iDEAL Hub. The URL should end with the version number of the
      notification service. This way the Initiating Party can influence which version of the
      notification service is used to send the request. At the moment only 'v3' is available. The
      DebtorTokenCallbackUrl can also be set in the Initiating Party service subscription.
```

For production, the "Debtortoken Callback URL" and the "Notification Bearer Token" **must** be filled in the iDEAL Subscription of the MSP Portal

If the Debtortoken Callback URL and Notification BearerToken are not filled out in the iDEAL Subscription of the MSP Portal, no debtor token callback notifications will be sent out by the Open Banking Service. No error messages will be provided by the Open Banking Service.

2.3.3.2 Step 3b (optional) - Retrieve Payment Status

Optionally, a Merchant can also retrieve the information about a payment by using the GET payments/{paymentId}/status API. Refer to the Open Banking Implementation Guide/Swagger files for exact specification).

For the iDEAL Checkout flow, status requests in the test environment can only be done for a period of 1 hour after the payment request was initiated. This is because our simulator of the iDEAL Hub only caches transactions for 1 hour.

3 Moving to Production

3.1 Productive Environment Setup

Once you have successfully completed testing in the test environment, you are ready to move to Production. Before doing so, please check the MSP Portal for the additional information you may need to include in the iDEAL Subscription page. Fields which are mandatory are marked out with a red star.

The following new fields would be visible to you under your iDEAL Subscription. If you do not have access to edit a certain field that applies to you, that means your Acquirer will fill that field out for you.

Here is a short overview on what each of these fields refers to :

Field	Definition
cPSP Flag	This field indicates whether a Merchant is a cPSP provider.
Debtortoken callback URL	URL where the Debtortoken callback is received by the Merchant. This field can also be included in the POST /payments API call. The value in API call will override that in the iDEAL Subscription of the MSP Portal. This functionality is only available if your Acquirer has passed their Release 3 certification with Currence.
Return URL	This refers to the Merchant Return URL. This field can also be included in the POST /payments API call. This is a mandatory field to have in either one of the locations. You may also send it in both. The value provided in the API will override the value from the iDEAL Subscription in the MSP Portal. <div style="display: flex; align-items: center;"> <div style="margin-right: 20px;"> InitiatingPartyReturnUrl string <i>(header)</i> </div> <div style="border: 1px solid #ccc; padding: 5px; border-radius: 5px;"> The callback URL for the redirection back to the initiating party after authorization. <input type="text" value="https://www.example.org"/> </div> </div>
Status Notification URL	Notifications will be sent to the Merchant to this URL. This field is mandatory in the iDEAL Subscription in the MSP Portal if a status notification request is expected by Merchant. The URL can be overridden by including it in the POST /payments API
Notification BearerToken	Static token of choice set by the Merchant to validate the notifications sent by the Open Banking Service. This is mandatory when a Merchant expects to receive a status notification.
CPSP Scheme ID	To be filled if a Merchant is a cPSP.
C2C Provider Scheme ID	To be filled if a (sub)Merchant is a C2C provider.
Country Code	Two-letter country code according to ISO 3166-1 alpha-2 standard. It indicates the origin country of the (sub)Merchant.
Merchant Category code (MCC)	The Merchant Category Code of the (sub)Merchant being paid for this transaction, following the ISO 18245 standard.

Depending on your Acquirer, certain fields may be mandatory/optional. Please check your iDEAL Subscription in the MSP Portal for which fields are marked out as mandatory.

3.2 Productive Transactions

Once the iDEAL subscription is enriched with the required information for iDEAL 2.0, you may proceed to make payments on the productive environment using the same steps as taken in Test Environment.

Please refer to the Open Banking v3 for iDEAL 2.0 Implementation guide for more details. For the URL to send requests in Production, please refer to the [Acquirer Profile](#) page.

4 iDEAL 2.0 FAQ

This section covers several frequently asked questions. Though it is hard to determine the root cause of a problem or question without looking into it deeper, we hope these questions and answers serve as a guide to steer the implementation and migration in the correct direction.

4.1 General

- **I am trying to login to the Base Domain provided in the iDEAL 2.0 Getting Started Guide but my ID and password do not work. Can you help me?**

The base domain provided in the Getting Started Guide is not meant for clicking on and logging in. It is used to obtain the full URL where requests should be sent. Merchants should use the Test Merchants provided in the Getting Started Guide.

- **I am subscribed to the R16 Reports for iDEAL 1.0. How can I also get this information for iDEAL 2.0 transactions?**

The reports for iDEAL 2.0 will need to be subscribed to separately. These are available for subscription via the MSP Portal, via the "Reports" tab.

- **I am using iDEAL QR. How will iDEAL QR work with iDEAL 2.0 transactions ?**

Since October 2017, "iDEAL QR" offers merchants the option of initiating iDEAL payments via QR codes. The entire iDEAL QR system is described in Currence's Implementation Guidelines for merchants, acquirers and issuers.

If an iDEAL acquirer offers iDEAL QR, the Merchant can register to the iDEAL QR backend via the MSP Portal in the iDEAL subscription.

The Merchant can create QR codes for iDEAL Payments via the iDEAL QR API. If such a QR code is scanned by a User with their smartphone, the payment data related to this QR code is displayed to the User on their device and they are asked whether they want to pay, i.e. whether they want to "confirm" the payment.

As soon as a User has confirmed a QR code, the Merchant is prompted by the iDEAL QR backend to initiate an iDEAL payment and return the Issuer redirect URL to the backend.

With the introduction of iDEAL 2.0, the merchant now has the choice of either starting an iDEAL 1.0 payment (as before) or starting an iDEAL 2.0 payment.

If the merchant already uses iDEAL QR with iDEAL 1.0, he has already implemented the necessary endpoints on his side and can process the "iDEAL QR transaction API call" and the "iDEAL QR status API call" from the iDEAL QR back.

With the transaction call, the merchant starts a new iDEAL 1.0 payment through an AcquirerTrxReq to the iDEAL 1.0 Acquiring System and with the status call, he asks the iDEAL 1.0 Acquiring System for the status of a specific iDEAL transaction by means of an AcquirerStatusReq. If the Merchant now wants to use iDEAL QR with iDEAL 2.0, he must replace the iDEAL 1.0 messages AcquirerTrxReq and AcquirerStatusReq with the corresponding API call from the Openbanking API v3 for iDEAL 2.0, i.e. with "POST /payments" or "GET /payments/{paymentId}/status". If using iDEAL QR with iDEAL 2.0, the Merchant MUST send in the CommonPaymentData.DebtorInformation.Agent which is the BIC of the issuer.

Since iDEAL 2.0 now also has active Status Notification requests, it is possible that the Merchant already knows the final status of an iDEAL 2.0 payment when the iDEAL QR backend asks for it via the "iDEAL QR Status API call". In this case, the Merchant does not have to ask the Open Banking Service for the status again, but can already return the known status directly to the QR backend.

4.2 Authorization Requests

- **I noticed when I use my own merchant id in the token request headers that I get the message "InitiatingParty status is APPLIED". So it seems that something is wrong, could you clarify this?**

Please check what environment is being used.

In the test environment, please use the test Merchants provided in the Acquirer Profile section. These should be used along with the test certificates.

In Production, the correct ID from the MSP Portal should be used. Please refer to the Acquirer Profile section for more information.

- **We are receiving an answer : Status 401, Not Authorized, initiating party is unknown. What are we doing wrong?**

If this is in the test environment, this might be the case of using an incorrect ID. Please use the test Merchants provided in the Acquirer specific profiles in the Getting Started Guide.

For the IDs to be used in production, please refer also to the Acquirer specific pages in the Getting Started Guide which outline where the correct ID can be found.

- **What are the IP addresses from which we will receive notifications from you?**

160.92.100.224 and 160.92.19.160

- **What are the parameter values to be used for the Authorize token endpoint?- Client- Id**

All information for doing first transactions is available in the iDEAL 2.0 Getting Started Guide. We strongly suggest referring to it for more information.

This is covered in detail in the Acquirer specific pages in the Getting Started Guide.

- **How long is the token valid for once it has been requested?**

When a token is first requested, it is valid for a period of 60 minutes. It can be used all the way until the expiration time. It is recommended to cache and reuse the access token during its validity period to prevent unnecessary token requests. If the request for the token is made again during the 60 minute period, the same token is returned for the first 50 minutes. In the last 10 minutes, a new token is provided. During this 10 minute overlap, both tokens will be valid, the old and also the new.

4.3 Payment Initiation

- **Once I make a payment, I land at page <https://worldline.com> . I am unable to continue from here. How do I ensure my testing went through successfully?**

In the productive environment, when the payment request is sent through to the Acquirer, Worldline processes the request and then it is sent to the iDEAL Hub. The redirect URL returned to the Merchant is the iDEAL Hub Pay page.

When testing, requests go through a simulation of the iDEAL Hub. At the moment, Currence has not provided any test environment for Merchant testing purposes. In order to test the success of payments, Merchants should refer to the Getting Started Guide under Payment Status Simulation. Here, it is outlined

what payment amounts should be sent in order to simulate different Payment statuses. For example, 1.00€ for SettlementCompleted.

- **In iDEAL 1.0 there were the fields Purchase ID and description. What fields do these correspond to in the Open Banking API ?**

→ RemittanceInformation field is the equivalent of Description in iDEAL 1.0

→ RemittanceInformationStructured > Reference field should be used in place of purchaseID in iDEAL 1.0.

4.4 Payment Finalization

- **How is the status of the payment received in iDEAL2.0 ? Must I request for the status? How do I do so?**

Merchants are given the possibility to receive push notifications for iDEAL 2.0 transactions. The payment Notification URL can be provided in the iDEAL Subscription of the MSP Portal as well as via the request header in the POST/payments request.

In order to receive the notification, the URL must be provided in the MSP Dashboard, in conjunction with the Notification BearerToken. This value can be overridden by the InitiatingPartyNotificationUrl header in the POST/payments request.

This is considered "Pushed based notification".

In the event something goes wrong, and they do not receive a notification from us, they may opt to make a GET/payments/{paymentId}/status request as explained in the Open Banking Implementation Guide for iDEAL.

- **I haven't received a notification on the payment request. Can you tell me why?**

For Merchants to receive a notification, they **must** fill in both the Status Notification URL and Notification Bearertoken in the iDEAL subscription of the MSP Portal. The InitiatingPartyNotificationUrl header in the API is present and this value will override the value which is entered in the subscription. The bearertoken can only be entered in the subscription.

- **What is the character limit for the Notification Bearertoken in the iDEAL MSP Subscription?**

A maximum of 255 characters may be provided in the Notification Bearertoken field in the iDEAL MSP Subscription.

- **What are the statuses I can expect to receive for iDEAL transactions via the Open Banking Service?**

The following statuses are relevant for iDEAL transactions.

SettlementCompleted → Equivalent to Success in iDEAL 1.0

Open

Cancelled

Expired

Error → Equivalent to Failure in iDEAL 1.0

5 Acquirer Profile

5.1 BNG Bank

Here you will find all specific information for BNG Bank as an Acquirer.

5.1.1 Signatures

Messages for BNG are **not signed**.

The authorization requests via POST /token API endpoint will still be signed, but other requests and responses such as POST /payments, GET /payments/{paymentId}/status are **not signed**.

5.1.2 Authorization Requests

Authorization requests are authenticated by checking the following fields to identify the Merchant's Acquirer and Initiating Party ID correctly :

App * required string (header)	The application name. For example PIS or AIS <input type="text" value="App"/>
Client * required string (header)	The name of the client. This name is provided to the Initiating Party during onboarding. The name of the client is created by the Open Banking Service. <input type="text" value="Client"/>
Id * required string (header)	The combination of Initiating Party ID and sub Id. For example if Initiating Party ID is 433 and the sub ID is 5 the ID will be 433:5 IP=433, subId=5 -> 433:5 IP=434, no subId -> 434 <input type="text" value="Id"/>

Header	Value	Description
App	IDEAL	App value is always IDEAL for iDEAL transactions, also in Production.
Client	BngNL	Client value is always BngNL for Merchants with BNG. This value is also the same in Production.
Id	Initiating Party ID*	This is the identifier used by the Open Banking Service to identify the Merchant. In the test environment, only the test Merchants should be used.

*The Initiating Party ID for a Merchant can be found in the MSP Portal via the Merchant Page or the Subscription Page. This is also known as **"Platform Merchant ID" or "Parent-Merchant ID"** for BNG Merchants. Please note this is to be taken from the iDEAL MSP Portal for transactions in **Production**.

General Details					
Service Merchant ID:	067002540	Parent-Merchant ID:	001741	Status:	ACTIVE
Service:	IDEAL	Legal Name:	BNG TEST QA2 Merchant A	Activation Date:	06/07/2022
				Contract Start:	01/07/2022
				Contract End:	

For transactions in the test environment, please refer below to the section on Test Merchants.

5.1.3 Test Merchants

To perform tests in the test environment, the Merchant must use the following test Merchants. Attempting to use other Merchant IDs will lead to error messages in the test environment.

Two Merchant IDs are available :

- **Test Merchant A** - Merchant with no Sub-Merchants
- **Test Merchant B** - Merchant or cPSP with Sub-Merchants

Distributing Service Providers (dPSPs) may also make use of the same preconfigured merchants for their integration tests.

Merchant	Initiating Party ID	Sub ID (use any one)
Test Merchant A	001741	N/A
Test Merchant B	001761	1,2,3

5.1.3.1 Examples

For Merchant without Sub-Merchants

Header	Value for test environment
Client	BngNL
Id	001741
App	IDEAL

For Merchant with Sub-Merchants (option A)

Header	Value for test environment
Client	BngNL
Id	001761:1
App	IDEAL

The generated token here will only be valid for Sub ID : 1.

Sub ID 2 and 3 will need a separate token to be generated.

For Merchant with Sub-Merchants (option B)

Header	Value for test environment
Client	BngNL
Id	001761
App	IDEAL

The generated token here will be valid for all sub Merchants under the Merchant/PSP 001761.

If using option B, the Merchant/PSP **must** send in the **InitiatingPartySubID** field under **Creditor Information** in the `POST /payments` API when making payment requests for a specific sub Merchant.

5.1.4 Transactions in Production

The base domain for Productive transactions is <https://openbanking1.worldline-solutions.com/>

Some useful full URLs are as follows :

- Authorization requests :
<https://openbanking1.worldline-solutions.com/xs2a/routing/services/authorize/token>
- Payment Initiation Requests :
<https://openbanking1.worldline-solutions.com/xs2a/routingservice/services/ob/pis/v3/payments>
- Get Payment Status Request :
<https://openbanking1.worldline-solutions.com/xs2a/routingservice/services/ob/pis/v3/payments/{paymentId}/status>

For the authorization requests, the App and Client values remain the same. The Id value is the Initiating Party ID from the MSP Portal as explained above in the section on Authorization Requests.