

The Franx logo consists of the word "franx" in a lowercase, sans-serif font, enclosed within a white, irregular polygonal shape. The background of the entire slide is a blurred photograph of a city skyline at dusk or dawn, with several skyscrapers illuminated and a body of water in the foreground with many dark, vertical posts or pilings.

franx

Trade and pay the smart way

- **PSD2 XS2A documentation Franx :  
Integrated UI solution v 1.0**



## Table of Contents

<b>1. Introduction</b> .....	<b>3</b>
<b>2. Conceptual overview</b> .....	<b>3</b>
2.1 The registration flow .....	3
2.2 The registration flow .....	3
2.3 Franx integrated UI flow .....	4
<b>3. Get started</b> .....	<b>4</b>
3.1 Acquiring access .....	5
3.2 Access Franx .....	6
<b>4. Accessing the data on the Franx Platform</b> .....	<b>7</b>
4.1 Getting Accounting Information .....	5
4.2 Initiating a Payment .....	5
4.2.1 Functional Definition .....	5
4.2.1 Getting the payment out the door .....	5
<b>5. Disclaimer</b> .....	<b>10</b>



# Franx PSD2 – Integrated UI Solution Documentation 1.0 (11<sup>th</sup> of September 2019)

## I Introduction

The Payment Services Directive (PSD2) is an European regulation designed to make managing cash-flows and making payments online more convenient for consumers and businesses. It aims to promote innovation by opening up to third parties – including companies that are not banks, often referred to as TPPs – and increase competition. PSD2 puts people and businesses in control of who can access their bank account information, gives them more options on how they manage their money and make payments online.

We welcome PSD2 and have been working closely with our development partners and TPP's to open up our banking platform. We look forward to offering our customers more enhanced products and better way of managing their money.

## 2. Conceptual Overview

Franx offers account information and payment initiation via an integrated solution in the regular user interface (UI) of Franx. Clients of Franx can request a TPP to acquire a balance overview or perform a payment transaction. The TPP initiates a session to Franx' platform on behalf of the client. It logs in to Franx' platform via an OAuth2 session and can control the platform using screen scraping technology or an headless browser. Franx did not implement any of the exemptions from PSD2 Regulatory Standard. Therefore the access to the data is every time secured under the validation of the strong customer authentication of the client.

### 2.1 Certificate

In order to access the Franx Integrated UI solution, your organization must possess the correct eIDAS certificate. The certificate must have the following specifications :

- Must be QualifiedCertificate: contains QCStatements extension
- Must have a QCType; QWAC only.
- Must have PSD2 statement
- Extract PSP roles from certificate: AS, PI, AI, IC according to ETSI TS 119 495 Section 4.2
- Validate certificate Organization Identifier according to ETSI TS 119 495 Section 5.2.1

### 2.2 The Registration Flow

To access Franx' PSD2 interface you need to register your organization at Franx. The registration is fully automated. It requires a mutual TLS connection using your QWAC eIDAS certificate. We check your certificate and registrations at the authorities and if valid, you will be registered within our platform.

## TPP registration flow



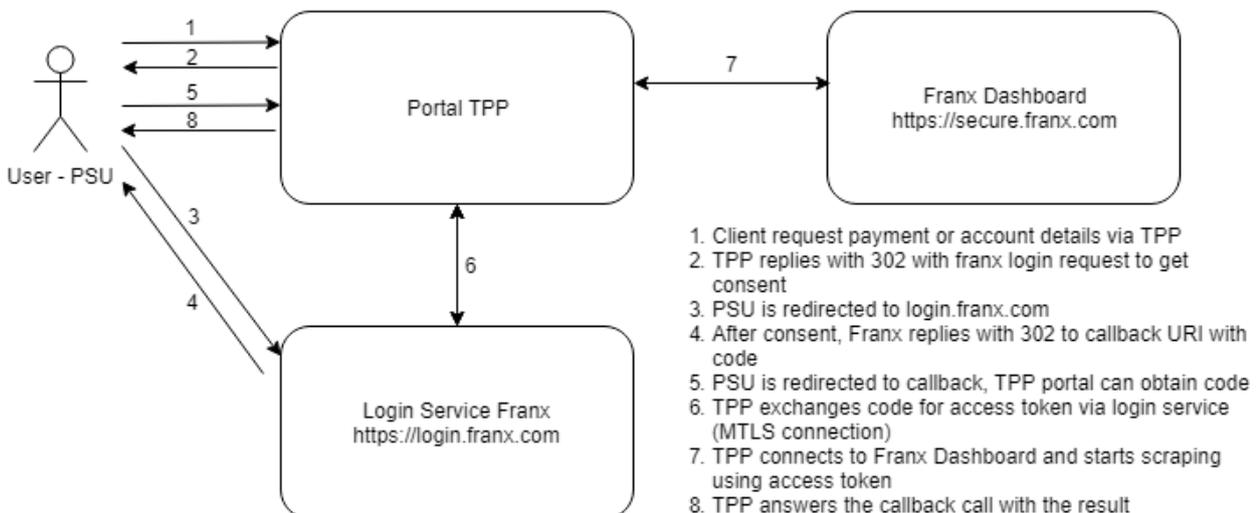
1. TPP request a registration of callback URI's (via MTLS)
2. Franx checks certificate for PSD2 scopes and if it is valid
3. Franx checks TPP registration and scopes in DNB register
4. Franx checks TPP registration and scopes in EBA register
5. Franx responds with the registration including client\_id

### TPP registration flow

#### 2.3 Franx Integrated UI Flow

To use the integrated UI solution, you need to setup an oAuth2 flow. First request a consent from Franx' client (PSU) and exchange the code for an access token. Use the access token to enter Franx' dashboard on behalf of the PSU. Within the dashboard, you can create payments and fetch account and balance information, depending on the consent and scopes of your registration.

## TPP AIS/PIS access flow



### TPP integrated UI flow



### 3. Get Started

Before you can request a scraping consent, you need to register at our oAuth2 service. The registration is simple, fast and fully automated:

- Send a POST request to our registration endpoint using your eIDAS QWAC certificate for setting up a mutual TLS connection

When your registration is successful you can start using Franx' platform directly.

*Sample request:*

```
curl -X POST \  
  https://login.franx.com:4443/register \  
  --cert TPPCertificate.crt \  
  --key TPPPrivateKey.key \  
  -H 'Content-Type: application/json' \  
  -d '{  
    "client_name": "Acme Bank",  
    "client_uri": "https://example.com",  
    "redirect_uris": [  
      "https://example.com/callback"  
    ]  
  }'
```

This request will register your organization, or you can update your settings.

*Request Fields*

<b>Field</b>	<b>Description</b>
<b>client_name</b>	<b>Name used for displaying in the consent screen</b>
<b>client_uri</b>	<b>Link to your site in consent screen</b>
<b>redirect_uris</b>	<b>List of allowed callback URI's for the oAuth sign-in request</b>

*Sample response*

```
{  
  "client_id": "PSDNL-DNB-B0123",  
  "client_uri": "http://example.com",  
  "redirect_uris": [  
    "http://example.com/callback"  
  ],  
  "scope": "portal:account_information portal:payments"  
}
```

*Response Fields*

<b>Field</b>	<b>Description</b>
<b>client_id</b>	<b>Your registration ID fetched from the eIDAS certificate</b>
<b>client_uri</b>	<b>Link to your site in consent screen</b>
<b>redirect_uris</b>	<b>List of allowed callback URI's for the oAuth sign-in request</b>
<b>scope</b>	<b>Space separated list of scopes you are allowed to use (fetched from eIDAS certificate, and validated with the authority)</b>



### 3.1 Acquiring access

Getting the consent

To gain access to the scraping pages, you need to request a consent from the PSU. You perform this consent by redirecting the PSU to

```
https://login.franx.com/identity/connect/authorize?client_id={{client_id}}&redirect_uri={{redirect_uri}}&response_type=code&scope={{scope}}&nonce={{nonce}}
```

You need to replace the values between the double-curly brackets with the following information:

<b>Attribute</b>	<b>Description</b>
<b>client_id</b>	<b>Your registration ID (see your eIDAS certificate)</b>
<b>redirect_uri</b>	<b>One of your redirect URI's</b>
<b>scope</b>	<b>Space separated list of scopes for the consent request</b>
<b>nonce</b>	<b>A random value</b>

The PSU get Franx' login page, including two-factor authentication, and need to give an explicit consent for the requested scopes. When the consent is given, the browser of the PSU is redirected to your `redirect_uri` with the URI argument code. You need the code to obtain an access token for Franx' dashboard.

*NOTE: This code is valid for 5 minutes.*

Obtaining the Access Token

You can exchange the code for an access token via a request to our login service.

*Sample Request*

```
curl -X POST \  
  https://login.franx.com/identity/connect/token \  
  --cert TPPCertificate.crt \  
  --key TPPprivateKey.key \  
  -H 'Content-Type: application/x-www-form-urlencoded' \  
  -d 'code={{code}}&grant_type=authorization_code&client_id={{client_id}}&redirect_uri={{redirect_uri}}'
```

*Request Form Attributes*

<b>Attribute</b>	<b>Description</b>
<b>code</b>	<b>The code retrieved via the redirect</b>
<b>client_id</b>	<b>Your registration ID (see your eIDAS certificate or registration response)</b>
<b>redirect_uri</b>	<b>The used redirect_uri</b>

*Sample response*

The response contains the access token you need to enter the scraping interface. Keep this token in a safe place, and let it never leave your infrastructure in an insecure way.



```
{
  "access_token": "eyJ0eXAiOiJKV1QiL...",
  "expires_in": 45,
  "token_type": "Bearer"
}
```

*NOTE: The access token is valid for 45 seconds.*

### 3.2 Access Franx

After you have obtained the access token, you can access Franx' platform to fetch account information or perform (international) payments. The easiest way to use the scraping interface is a headless browser, like chromium. Make sure your viewport is large enough (1900 x 1200), otherwise some elements are not visible for the scraping tool. It is important to use the same browser screen during the complete transaction to keep the session.

## 4. Accessing the data on the Franx platform

Direct your headless browser to <https://secure.franx.com/scraping/> and use the access token to log in. When the login is successful you are redirected to the dashboard of Franx with the account of the PSU. Depending on the scope, account information is visible, and/or you are able to perform payments. The session will be closed after 5 minutes of inactivity.

### 4.1 Get account information

Fetching the account information is possible from the home screen of the dashboard. You can obtain the current funds, or the account movements.:

- <https://secure.franx.com/> -- **Fetch general account information.**
- <https://secure.franx.com/reporting/account-movements/{{CURRENCY}}> -- **Fetch account movements, CURRENCY should be one of the currencies: EUR, USD, GBP, CHF, JPY, AED, CAD, CNH, CZK, DKK, HKD, HUF, ILS, KES, MAD, MXN, NOK, NZD, OMR, PLN, RUB, SAR, SEK, SGD, and THB.**

For example, to fetch the current balances you can implement the following flow:

1. **Go to <https://secure.franx.com/>**
2. **The IBAN can be grabbed by selecting the element with the attribute [data-accountnumber] and obtain the text value**
3. **The balances can be grabbed by looping through the elements with the attribute [data-balance] and obtain the text value**

### 4.2 Initiate a payment

#### 4.2.1 Functional definitions

**Franx offers SEPA and international payments. With our International Payment service, you can make payments in currencies other than the Euro. The Franx business IBAN provides one**

single current account for 33 currencies. Apart from the Euro, the Franx payment functionality accepts payments in the following currencies:

**USD - US Dollar**

**GBP - British Pound**

**CNY - Chinese Yuan**

**AED - UAE Dirham**

**AUD - Australian Dollar**

**CAD - Canadian Dollar**

**CHF - Swiss Frank**

**CZK - Czech Koruna**

**DKK - Danish Krone**

**HKD - Hong Kong Dollar**

**HUF - Hungarian Forint**

**ILS - Israeli New Shekel**

**JPY - Japanese Yen**

**MXN - Mexican Peso**

**NOK - Norwegian Krone**

**NZD - New Zealand Dollar**

**PLN - Polish Zloty**

**RON - Romanian Leu**

**RUB - Russian Ruble**

**SAR - Saudi Arabian Riyal**

**SEK - Swedish Krona**

**SGD - Singapore Dollar**

**THB - Thai Baht**

**TRY - Turkish Lira**

**ZAR - South African Rand**

**BGN - Bulgarian Lev**

**BHD - Bahraini dinar**

**KES - Kenyan Shilling**



**KWD - Kuwaiti Dinar**

**MAD - Moroccan dirham**

**OMR - Omani Rial**

**QAR - Qatari riyā**

#### 4.2.2 Getting the payment out the door

Performing a payment is more complex. You need to create a payment and request an OTP (one time password) code. The OTP will be send to the phone of the PSU. It is necessary to keep the browser session alive when asking the PSU to enter the OTP code to approve the payment. When you know the OTP, you can approve the payment and mark it as successfully. If you don't keep the session alive, the OTP becomes invalid

For example, to create a payment implement the following steps:

1. **Go to** <https://secure.franx.com/payment?franx-feature=paymentsnewscreen>
2. **Wait till the page is loaded, and set the attribute** `[data-payment-input = "transaction-currency"]` **to the required currency, for example EUR**
3. **Set the amount via the attribute** `[data-payment-input="transaction-amount"]`
4. **Set the name of the counter party via the attribute** `[data-payment-input="searchBeneficiaryByName"]`
5. **Set the destination country of the payment via the attribute** `[data-payment-input="beneficiary-country"]`
6. **Set the country of the destination bank via the attribute** `[data-payment-input="beneficiary-bank-country"]`
7. **You need to wait till the form is ready to accept the payment destination account details**
8. **Set the destination bank account via the attribute** `[data-payment-input="searchBeneficiaryByAccountNumber"]`
9. **Put a payment description via the attribute** `[data-payment-input="payment-description"]`
10. **Submit the payment by clicking on the button** `[data-payment-action="payment-submit"]`
11. **Wait for the text Payment initiated successfully and** `[data-info="loading"]` **element disappeared**
12. **Optionally fetch the payment id via the attribute** `[data-payment-info="paymentId"]` **to show to the PSU**



13. **Click on the approve button with the attribute** [data-payment-action="request-approval-code"]
14. **Wait for the element with attribute** [data-payment-info="approval-code-sent"]
15. **Request an approval code by click on the button with attribute** [data-payment-action="approve-payment"]
16. **Wait for the text with attribute** [data-payment-info="approval-code-sent"].
17. **Keep the scraping browser session open, and ask the PSU to submit its received code. The PSU will see the amount and destination account in the sent SMS message**
18. **Enter the code received by the PSU in the field with the attribute** [data-payment-input="approval-code"]
19. **Click on the button** [data-payment-action="approve-payment"]
20. **The payment was successful when the text** [data-payment-info="approved"] **appears**

## 5. Disclaimer

- **The HTML classes and identifiers used in the scraping examples might be subject of change.**
- **Any changes to the above mentioned will be communicated with the market 3 months in advance.**
- **All TPP activity is logged**