

# PSD2 API

## Interface specification



# Contents

<b>Contents</b> .....	<b>2</b>
<b>1 Introduction</b> .....	<b>7</b>
1.1 Main focus.....	7
<b>2 Overall description</b> .....	<b>8</b>
<b>3 Interfaces</b> .....	<b>9</b>
3.1 Security requirements .....	9
3.2 Differences between test and production environments .....	13
3.3 Request submission process AI.....	14
3.3.1 IBAN Listing for Account Information (LIAI).....	15
3.3.2 Account Information (AI) .....	17
3.3.3 Account Information Result .....	19
3.4 Request submission process LOT .....	22
3.4.1 IBAN Listing for List Of Transactions (LILOT) .....	22
3.4.2 List Of Transactions (LOT).....	24
3.4.3 Transactions result .....	27
3.5 Request submission process POI.....	29
3.5.1 IBAN Listing for Payment Order Initiation (LIPOI) and for Standing Order .....	31
3.5.2 Payment Order Initiation (POI) .....	36
3.6 Payment Instrument Issuer Service Provider (PIISP) .....	55
3.6.1 /ws/AccountService/1.0/balanceCheck .....	55
<b>4 List of error codes</b> .....	<b>58</b>

## List of figures

Figure 1: TPP Authentication .....	9
Figure 2: Sequence of call steps for AI service.....	14
Figure 3: Sequence of call steps for LOT service .....	22
Figure 4: Sequence of call steps for POI service.....	30
Figure 5 Orchestration PIISP.....	55

## Document information

<b>Name</b>	PSD2 API
<b>Subject</b>	Interface specification
<b>Version</b>	6.00

# List of acronyms

## List of acronyms

#	Acronym	Explanation
	365.bank or bank	365.bank, a. s. with its registered at Dvořákovo nábrežie 4, 811 02 Bratislava, ID No.: 31 340 890
	AI	Account Information
	AISP	Account Information Service Provider
	API GW	API Gateway
	CA	Certificate authority
	IB	Internet banking
	IL	IBAN Listing – list of accounts
	MEP	Message Exchange Pattern
	PISP	Payment Information Service Provider
	PIISP	Payment Instrument Issuer Service Provider
	POI	Payment Order Initiation
	PSD2	Payment Service Directive 2
	SCA	Strong customer authentication
	SSL	Secure Socket Layer
	TLS	Transport Layer Security
	TPP	Third Party Provider

## References

#	Name	Description
	<u>PSD2</u>	DIRECTIVE (EU) 2015/2366 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 25 November 2015 on payment services in the internal market, amending Directives 2002/65/EC, 2009/110/EC and 2013/36/EU and Regulation (EU) No 1093/2010, and repealing Directive 2007/64/EC
	<u>RTS on SCA and CSC</u>	COMMISSION DELEGATED REGULATION (EU) 2018/389 of 27 November 2017 supplementing Directive (EU) 2015/2366 of the European Parliament and of the Council with regard to regulatory technical standards for strong customer authentication and common and secure open standards of communication
	<u>eIDAS</u>	REGULATION (EU) No 910/2014 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 23 July 2014 on electronic identification and trust services for electronic transactions in the internal market and repealing Directive 1999/93/EC

# 1 Introduction

The Document describes PSD2 regulation and RTS on SCA and CSC interfaces which are provided by 365.bank.

## 1.1 Main focus

The document focuses on following topics:

- Describe interfaces which are provided by 365.bank for Third Party Providers (TPP)
- Define security requirements for TPP
- Describe invocation sequence for proper interface handling

## 2 Overall description

Bank allows the connection of third parties to the API without the obligation to register and without granting special access tokens. The interfaces for TPP communication with PABK are described in detail in the following chapters of the documentation. TPP accesses the API interface by direct access by connecting to individual endpoints of the issued services in accordance with defined security requirements based on the client's request.

The API provides the following functionalities from PSD2 control and RTS on SCA and CSC:

PSD2 Services provided in the API		
IBAN Listing	IBAN Listing for AI (LIAI)	Account list for Account Information
IBAN Listing	IBAN Listing for LOT (LILOT)	Account list for Account Transactions
IBAN Listing	IBAN Listing for POI (LIPOI)	List of accounts for the payment service
AISP	Account information (AI)	Account information - The service provides information and balances related to the account Account transactions - the service provides a list of transactions in a defined range of dates related to the account
AISP	List of Transactions (LOT)	Service allows you to authorize an initialized payment. The service is defined according to the type of payments for individual endpoints. The bank does not currently provide instant payments.
PISP	Payment Order Initiation (POI)	Submitting an e-commerce payment
PISP	Payment Order Initiation (POI)- Ecommerce	Submission of a standard SEPA payment
PISP	Payment Order Initiation (POI) – standard payments	Creating a SEPA standing order
PISP	Payment Order Initiation (SOI) - Standing Order	Balance check - the service provides information about a sufficient balance by answering yes / no
PIISP	Payment Instrument Issuer Service Provider	

The sequence of processes is shown in diagrams describing the individual steps for making calls for each service. Failure to comply with the sequence of calls may lead to the rejection of the request by the bank in individual processes. It is important to follow the case-sensitive pattern in both the header and the body in the calls according to the template given in this documentation.

Communication pattern follow 2 scenarios from TPP point of view:

- Multi step pattern with online authorization by client (AISP- AI, AISP- LOT, PISP-POI, SOI)
  - Client puts inquiry via TPP
  - Client authorizes inquiry via 2 factor authentication (Internet banking)
  - Client checks result of inquiry via TPP
- Single step pattern with forward authorization by client (PIISP)
  - Result of the inquiry is immediate, client approved processing of the inquiry via different channels in the past



## 3 Interfaces

### 3.1 Security requirements

Client SW (TPP) willing to use 365.bank API has to follow security requirements described in this section. Communication has to be initiated via TLS (mutual authentication). Brief description is on the following figure.

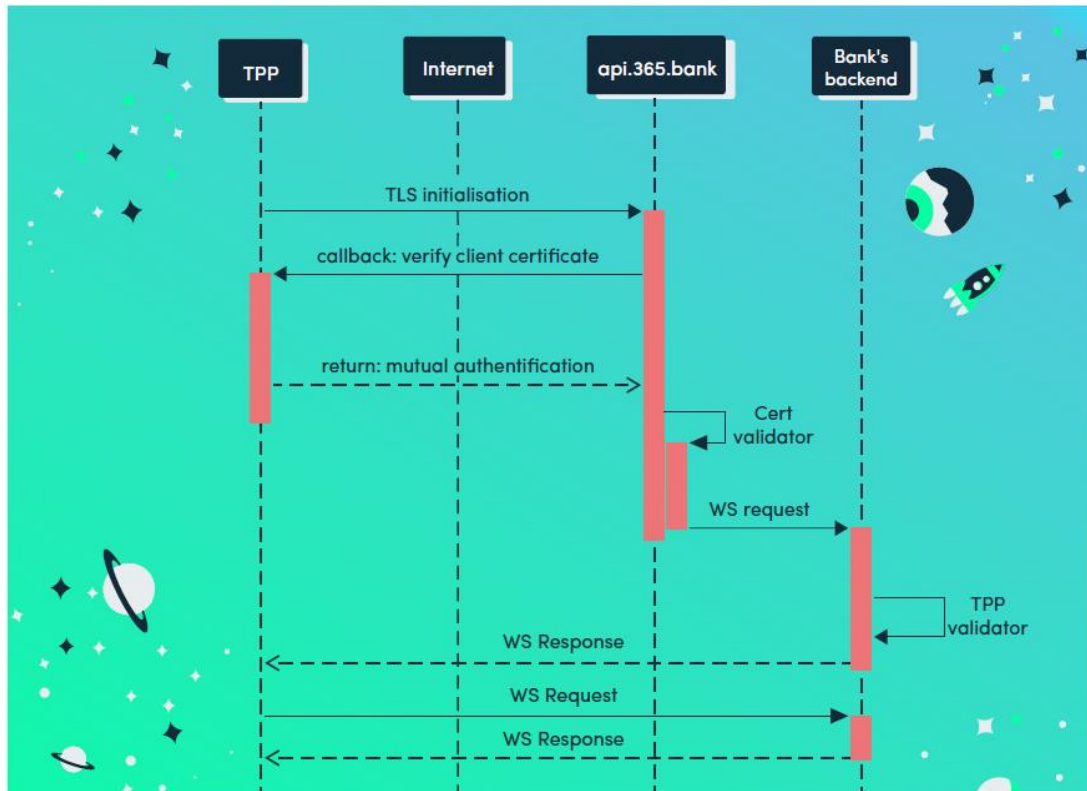


Figure 1: TPP Authentication

Interactions with 365.bank API has to be initiated with TLS handshake. Client has to use valid certificate (X.509 SSL type EV or qualified eIDAS certificate) issued via public certificate authority. Detailed requirements to initiate TLS could be found in RFC 6125.

Version: TLSv1.2

Supported cipher suite: TLS-ECDHE-RSA-WITH-AES-128-GCM-SHA256, TLS-ECDHE-RSA-WITH-AES-256-GCM-SHA384

Client identification is verified and following attributes are validated:

1. Certification authority signature
2. Certificate validity
3. Certificate ownership

Client certificate requirements for a production environment:

- the certificate complies with the X.509 standard
- the certificate is issued by a trusted certification service provider (CA)
- the certificate must contain according to the ETSI TS 119 495 standard:
  - the name of the registrar who assigned the license, including the roles for which the license was issued (qcStatement/ OID 0.4.0.19495.2)
  - license number issued by the registrar (organizationIdentifier/ 2.5.4.97)
- type certificate is accepted:
  - eIDAS qualified certificate for website authentication (eIDAS SSL must meet the requirements of Article 45 of Regulation (EU) No. 910/2014 (the list of accepted trust service providers is available on the website of the Slovak National Security Office (<http://nbu.gov.en/trust-services/index.html>) or the EU Trust Service website (<https://esignature.ec.europa.eu/efda/tl-browser/#/screen/home>))
- The certificate must contain according to eIDAS 2014/910
  - The name of the legal entity to which the certificate is issued (organizationName / OID 2.5.4.10)

- The elements of the address, at least including the city and state, the natural or legal person to whom the certificate is issued and, where applicable, as indicated in the official records (countryName and Locality)
- The domain names operated by the natural or legal person to whom the certificate is issued (commonName, or SAN records in case of multiple domains)
- The certificate may contain additional attributes according to the requirements of the issuer (CA), resp. requirements of individual standards
  - Serial number, Subject Key Identifier
  - Valid From / Valid To
  - Information needed to verify against the OCSP
  - Information needed to verify against the CRL
  - Qualified Certification Policy – Web, Policy identifier=0.4.0.194112.1.4 or 0.4.0.19495.3.1
  - The Enhanced Key Usage attribute must contain a value Client Authentication (OID 1.3.6.1.5.5.7.3.2)
  - Certificate issuer validation values (Authority Key Identifier; values from the attribute Issuer: O, CN, C; values from issuer's certificate (Serial number, Subject Key Identifier)

Immediately after the secure connection is established it is possible to call application interfaces which are described in following sections. Application interfaces are depicted as WS Request/WS Response on figure 5.

Client certificate requirements for test environment:

- TPP has PSD2 license
  - certificate meets requirements for production environment noted above
  - TPP sends confirmation before start testing to support\_tpp@365.bank with certificate
- TPP is in the process of PSD2 license issuing:
  - Accepted certificate is type of Extended Validation SSL certificate
  - TPP confirms his license process, which is provided by the Regulator (confirmation included name of TPP, identification ID of TPP, role)

TPP sends confirmation before start testing to support\_tpp@365.bank with certificate

Password encryption

- Test environment

The login password must be sent in RSAES-OAEP (RSA/ECB/OAEPWithSHA-256AndMGF1Padding) encrypted form and converted to a Base64 string. The encryption key must be used for encryption:

XML format:

<RSAKeyValue>

<Modulus>

```
q7Tv/ks00opyhDelWHYar+azEYIGRHwgUDEPLMeDGsnpEKwMYROI1HD+t4GRgfKXbPsXSkd31VT472Q4kesYbiyELVyU
hwzzjSBo8YqaJ0jACDSs8fNSrBaio/ELCVHpGnFQmLoDACNDhK079pYXxUg1j1EUCGVbLu6sq6GIGeZHe/NRavJCn+zx
v42aGDczQPwFYXz1xfMrDFgQIXTdb+XbpK99FYe/CnEFbABEM7Gf5MsoAXr/vcPwkdFcZng/qVC5F+E3T78sqYgnNi17
VT5d++QYQbir6wM/sMQKRhYiZ/fE/ARZiLsEA9r592hpbZAHlgtcd4vsAPJ00Y1d+Q==
```

</Modulus>

<Exponent>

AQAB

</Exponent>

</RSAKeyValue>

PEM format:

```
-----BEGIN PUBLIC KEY-----  
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAq7Tv/ksO0opyhDeIWHYa  
r+azEYIGRHwgUDePLMeDGSnpEKWMYROI1HD+t4GRgfKXbPsXSkd31VT472Q4kesY  
biyELVyUhwzzjSBo8YqajOjACDSs8fNSrBaio/ELCVHpGnFQmLoDACNDhK079pYX  
xUg1jIEUCGVbLu6sq6GIGeZHe/NRavJCn+zxv42aGDczQPwFYXz1xfMrDFgQIXTd  
b+XbpK99FYe/CnEFbABEM7Gf5MsoAXr/vcPWkdFcZng/qVC5F+E3T78sqYgnNi17  
VT5d++QYQbir6wM/sMQKRhYiZ/fE/ARZiLsEA9r592hpbZAHlgtcd4vsAPJOOY1d  
+QIDAQAB  
-----END PUBLIC KEY-----
```

Example for test environment:

Login and password from Use-cases before encryption:

**X-username:** A1psd2365@gmailabc.com

**X-password:** Test365Test1

Login name and password after encryption:

**X-username:** A1psd2365@gmailabc.com

**X-password:**

EvN6e8sB22iswD5rlnzyRoZq8q5SsWTn/7AuT+4LD3L6jFvskwZ5NVGYbXOqkVDSpO+ZgHHIjaVYvrtH6dKeMfKJkFH4UQ6xL3SPRmAiGDcBTSEXY5U3Os6tFyIbO3CRXiy9GB8TQlqRLqJxxWHHFTsqbK+p3FhqcTdJmCYFkeZWPPpjFoSEVD0/EWUsxUTu9ppq+niD8Mdal836I5oGILzNWc0MXq+Z2TiGcbKu6IKuOu9iJeYWKFeWGfokfClfNxXmyhIINKCvFkh5xiH+RM61uMx6mG+bEJIX1Y74rKmUssPm9dxL3RaFbtrF8SooWOpngHL1t9A49LSBbVRiCftg==

- Production environment

The login password must be sent in encrypted form. To encrypt, you need to download the key from the URL:  
<https://365.bank/cert/passwd/pubkey>

The password is encrypted by an algorithm RSAES-OAEP (RSA/ECB/OAEPWithSHA-256AndMGF1Padding). Finally, the resulting text is encoded in BASE64 format. The resulting value is filled in the X-password parameter in the request header.

Example:

```
java -Djavax.net.ssl.keyStore=clientkeystore.p12 \  
-Djavax.net.ssl.keyStorePassword=password \  
-Djavax.net.ssl.trustStore=clienttruststore.jks \  
-Djavax.net.ssl.trustStorePassword=password \  
com.fintech.psd2api.SSLSocketClient
```

Python

```
import json  
import requests  
  
serverCert = "server.crt"  
clientCert = "client.crt"  
clientKey = "client.key"  
url = "https://psd2api.example.com"  
payload = {"ID": "123456"}  
headers = {"content-type": "application/json"}  
r = requests.post(url, data=json.dumps(payload), verify=serverCert,  
                  headers=headers, cert=(clientCert, clientKey))  
  
print(r.status_code)  
print(r.json())
```

### 3.2 Differences between test and production environments

A) The following test credentials must be used for the test environment

password	username	Account
Test365Test1	A1psd2365@gmailabc.com	SK936500000000096824759
Test365Test2	B2psd2365@gmailabc.com	SK716500000000096824767
Test365Test3	C3psd2365@gmailabc.com	SK496500000000096824775
Test365Test4	D4psd2365@gmailabc.com	SK276500000000096824783
Test365Test5	E5psd2365@gmailabc.com	SK056500000000096824791
Test365Test6	F6psd2365@gmailabc.com	SK366500000000096985835
Test365Test7	G7psd2365@gmailabc.com	SK146500000000096985843
Test365Test8	H8psd2365@gmailabc.com	SK896500000000096985851
Test365Test9	J9psd2365@gmailabc.com	SK396500000000096985878
Test365Test10	K10psd2365@gmailabc.com	SK176500000000096985886

B) The following creditor test accounts must be used for the test environment:

cdtrAcc	form
SK706500000000096735836	SEPA
SK486500000000096735844	SEPA
SK266500000000096735852	SEPA
SK046500000000096735860	SEPA
SK736500000000096735879	SEPA
SK516500000000096735887	SEPA
SK296500000000096735895	SEPA
SK666500000000096735908	SEPA
SK226500000000096735924	SEPA
SK976500000000096735932	SEPA
SK756500000000096735940	SEPA
SK476500000000096735959	SEPA
SK526500000000020605904	SEPA
SK306500000000020605912	SEPA
SK316500000000035687551	SEPA
SK256500000000096735967	SEPA
SK036500000000096735975	SEPA
SK446500000000096735916	SEPA
SK786500000000035687578	SEPA
SK566500000000035687586	SEPA
SK346500000000035687594	SEPA
SK556500000000020605947	SEPA
SK936500000000096824759	SEPA

SK716500000000096824767	SEPA
SK496500000000096824775	SEPA
SK276500000000096824783	SEPA
SK056500000000096824791	SEPA

If TPP requires testing on its own accounts, then it is necessary to send account numbers in the IBAN format to support\_tpp@365.bank for the purposes of registration by the bank. Following confirmation by the bank, it is possible to continue testing with the submitted accounts.

The Bank responds to all TPP complaints within a reasonable response time, which depends on the complexity and complexity of the TPP party's request.

- C) If SCA is required in the test environment, then it is necessary to compile the OTP code in the format DDMMHH  
DD - the day of sending the request to the service that requested the SCA - numeric value in response  
MM - month of sending the service request, which requested SCA - numeric value in response  
HH - time of sending the service request that requested the SCA in the response - numeric value
- D) Operations realized through the test environment are performed exclusively for the given service and only the functionality of the service is verified. The test environment uses data in a way that allows individual services to be called. The result of one service does not follow the update of mock data in another service. Example: if a transfer is realized in the test environment that returns the status completed, it is not possible to verify the processing of the transaction via the List of transaction service.

### 3.3 Request submission process AI

The figure below describes the sequence of API calls that must be followed in that order. Failure to comply with the call sequence may result in the bank rejecting the request.

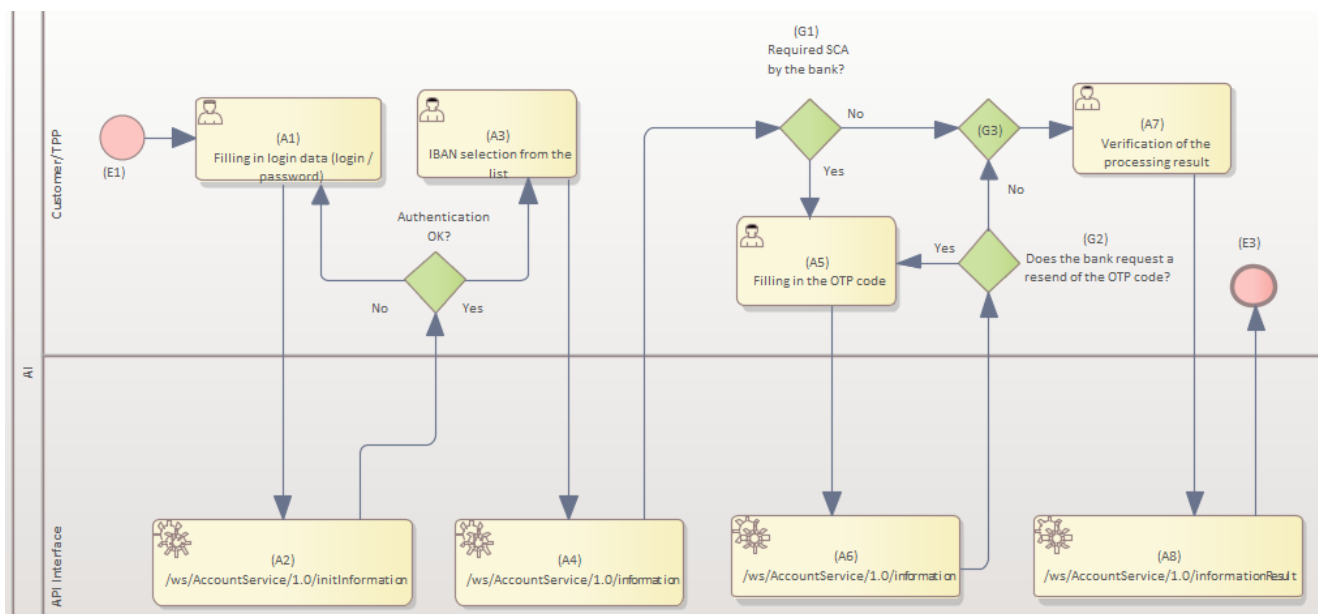


Figure 2: Sequence of call steps for AI service

Description process:

- (A1) The client enters his login details (login and password) with the TPP. After they are obtained, the TPP calls the initInformation service.
  - For the test environment, the login data listed in chapter 3.2 must be used
- (A2) After verifying the accuracy of the data, the bank responds with a list of accounts available to the client. The service response also returns the "orderId" attribute. This is a unique identifier that will be used throughout the process.

- (A3) The client defines an account from the list. After selecting a account, TPP will call the information service with added account information and assigned orderID.
- (A4) The service returns in response information on whether authorization is required.
  - SCA is required
    - A5 (A5) The TPP must request an OTP code from the client to continue. The code sends information by calling the service and the request must contain the same values as in point (A4)
      - DD - day of sending the request – numeric value
      - MM – month of sending the request – numeric value
      - HH – hour of sending the request – numeric value
      - Example: Request (A4) was sent on 27.5.2022 at 15:53. In request (A6) we expect OTP code 270515A6
    - (A6) The service returns authorization information in response.
  - SCA not required - the request entry process is completed. In this case, the request (A6) is skipped and the TPP can proceed to verify the processing result. The Bank applies an exemption from the SCA pursuant to RTS Article 10. The Bank may request an SCA based on a call security assessment.
- (A7) To verify the processing result and display the account detail, it is necessary for TPP to call the informationResult service
- (A8) The service response returns the required data.

### 3.3.1 IBAN Listing for Account Information (LIAI)

The test environment is different from the production environment by a different URL. If you transition from a test environment to a production environment, you need to change the URL. To access the bank's environment, use only the following URL:

URL environment test: <https://apitest.365.bank/ws/AccountService/1.0/initInformation>

Production environment URL: <https://api.365.bank/ws/AccountService/1.0/initInformation>

#### 3.3.1.1 /ws/AccountService/1.0/initInformation

Transport	HTTPS/XML
Method	POST
MIME Type	application/xml
MEP	Synchronous request-response
XSD schema	AccountService_api.xsd

Request message		
	Accept	application/xml
HTTP Header parameters	X-username	Client login to Internet banking
	X-password	Client password to Internet banking
HTTP Content	InitInformationRequest	
Response message		
HTTP Header parameters	error	Error code
	errorDescription	Error description
HTTP Content	InitInformationResponse	

### 3.3.1.1.1 REQUEST MESSAGE - INITINFORMATIONREQUEST

Element name	Required	Type	Description
messageId	Yes	xsd:string	Unique identifier of TPP message

Example of a request from the bank's environment test:

Header:

POST <http://apitest.365.bank/Psd2Mock/ws/AccountService/1.0/initInformation HTTP/1.1> (do not use this URL to access the bank's environment)

Accept-Encoding: gzip,deflate

Content-Type: application/xml

X-password:

p76WQVzQ8y3oy0bApjjh6ln21ojuRkJZAX8LmFWo7FMDsmJHtqLICr6gv0ybk4fK6ISQse50z/ymb5nXNRVuwWQt+0rUUTcvB9nYPv3X1Wr9opOz0JPI0HhBd1R+h23AwCPft8CCXBvW3Vad5Es sPaygRIMr4xgiWmcyenOKZQ70Ki5L1mUfMZ0ZsljDKrelZ2iv+UziCCviyavO1+ufR1bha9pHNKqY HSB/aTOX6nlYImN0KaoD86nl/6x/IDSTZo9SO642eVWePN2NPd9NCQxO+7ObYFES3UZFrDz 612n8ZhkZzlVvSS5gXtUI0UvQ/fdtVxeEyfymZGDpOQRcyw==

Accept: application/xml

X-username: A1psd2365@gmailabc.com

Content-Length: 191

Host: apitest.365.bank

Connection: Keep-Alive

User-Agent: Apache-HttpClient/4.1.1 (java 1.5)

Body:

```
<psd2acc:InitInformationRequest
xmlns:psd2acc="http://pabk.sk/EAI/API/XSD/PSD2/AccountService">
  <messageId>fcaa51f2-7827-4ca7-af2e-9bc8f65c0781</messageId>
</psd2acc:InitInformationRequest
```

### 3.3.1.1.2 RESPONSE MESSAGE - INITINFORMATIONRESPONSE

Element name	Type	Description
orderId	xsd:string	Unique order ID
accountList		IBAN Listing
=id	xsd:string	IBAN

Example of a response from a bank environment test

HTTP/1.1 200 OK

Content-Length: 2653

Content-Type: application/xml; charset=UTF-8

Server:

X-Content-Type-Options: nosniff

error: 0

Date: Fri, 10 Jun 2022 08:24:51 GMT

```
<psd2acc:InitInformationResponse xmlns:psd2acc="http://pabk.sk/EAI/API/XSD/PSD2/AccountService">
  <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
    <ds:SignedInfo>
      <ds:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315" />
      <ds:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256" />
      <ds:Reference URI="">
        <ds:Transforms>
```



```

    <ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature" />
  </ds:Transforms>
  <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
  <ds:DigestValue> Value not specified in the example </ds:DigestValue>
</ds:Reference>
</ds:SignedInfo> Value not specified in the example </ds:SignatureValue>
<ds:KeyInfo>
  <ds:X509Data> Value not specified in the example </ds:X509Certificate>
</ds:X509Data>
</ds:KeyInfo>
</ds:Signature>
<orderId>a423edabac1943109818f6140f437423</orderId>
<accountList>
  <id>SK936500000000096824759</id>
</accountList>
</psd2acc:InitInformationResponse>

```

### 3.3.2 Account Information (AI)

The service described in this chapter is requested in two parts of the process. The first time if the client has chosen an account (this request is always mandatory) and the second time if the bank requests the implementation of SCA (The second request can be repeated several times, it depends on whether the OTP code was sent for the first time or not). All information service requests must have the same completed part of the request, which is labeled "Body" in the example (messageld may vary). Requests differ in the header because the first call does not contain the X-OTP attribute, so the second call already has this attribute.

The test environment is different from the production environment by a different URL. If you transition from a test environment to a production environment, you need to change the URL. To access the bank's environment, use only the following URL:

URL test environment: <https://apitest.365.bank/ws/AccountService/1.0/information>

Production environment URL: <https://api.365.bank/ws/AccountService/1.0/information>

#### 3.3.2.1 /ws/AccountService/1.0/information

Transport	HTTPS/XML
Method	POST
MIME Type	application/xml
MEP	Synchronous request-response
XSD schema	AccountService_api.xsd

Request message		
	Accept	application/xml
HTTP Header parameters	X-orderId	orderId of operation, which has to be authorized
	X-OTPkod	OTP authorization code
HTTP Content	InformationRequest	

Response message		
HTTP Header parameters	error	Error code
	errorDescription	Error description

## 3.3.2.1.1 REQUEST MESSAGE - INFORMATIONREQUEST

Element name	Required	Type	Description
IBAN	yes	xsd:string	client IBAN- (international bank account number)
messageId	yes	xsd:string	Unique identifier of TPP message

Example of a request from the bank's environment test:

Header of the first:

POST [http://apitest.365.bank/Psd2Mock/ws/AccountService/1.0/information\\_HTTP/1.1](http://apitest.365.bank/Psd2Mock/ws/AccountService/1.0/information_HTTP/1.1) (do not use this URL to access the bank's environment)

Accept-Encoding: gzip,deflate  
 Content-Type: application/xml  
 Accept: application/xml  
 X-orderId: a423edabac1943109818f6140f437423  
 Content-Length: 211  
 Host: apitest.365.bank  
 Connection: Keep-Alive  
 User-Agent: Apache-HttpClient/4.1.1 (java 1.5)

Header of the second request:

POST [http://apitest.365.bank/Psd2Mock/mockup\\_direct/ws/AccountService/1.0/information\\_HTTP/1.1](http://apitest.365.bank/Psd2Mock/mockup_direct/ws/AccountService/1.0/information_HTTP/1.1) (do not use this URL to access the bank's environment)

Accept-Encoding: gzip,deflate  
 Content-Type: application/xml  
 Accept: application/xml  
 X-orderId: a423edabac1943109818f6140f437423  
 X-OTPKod: 030223  
 Content-Length: 211  
 Host: apitest.365.bank  
 Connection: Keep-Alive  
 User-Agent: Apache-HttpClient/4.1.1 (java 1.5)

Body:

```
<acc:InformationRequest xmlns:acc="http://pabk.sk/EAI/API/XSD/PSD2/AccountService">
  <IBAN>SK936500000000096824759</IBAN>
  <messageId>63295640-a63c-4907-8189-a09633da2749</messageId>
</acc:InformationRequest>
```

## 3.3.2.1.2 RESPONSE MESSAGE - INFORMATIONRESPONSE

Element name	Required	Type	Description
orderId	yes	xsd:string	Unique ID of Order corresponding to client inquiry
SCA_Type	yes	xsd:boolean	Attribute for requirement of additional authorisation 1 – is required 0 – is not required

If Er27 is returned in the response header, it is a final error message and the order is rejected (eg multiple incorrectly entered verification codes). In this case, you need to start request from the beginning.

Example of rejection from the bank's test environment  
 HTTP/1.1 200 OK  
 Transfer-Encoding: chunked  
 Content-Type: application/xml; charset=UTF-8  
 Server:  
 Content-Type-Options: nosniff  
 error: Er27  
 errorDescription: Prikaz nebolo mozne spracovat. Vyprsal cas pre autorizaciu.  
 Date: Tue, 07 Jun 2022 07:40:31 GMT

Example of a positive response from a bank environment test  
 HTTP/1.1 200 OK  
 Content-Length: 2597  
 Content-Type: application/xml; charset=UTF-8  
 Server:  
 X-Content-Type-Options: nosniff  
 error: 0  
 Date: Fri, 10 Jun 2022 08:59:35 GMT

```
<psd2acc:InformationResponse xmlns:psd2acc="http://pabk.sk/EAI/API/XSD/PSD2/AccountService">
  <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
    <ds:SignedInfo>
      <ds:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315" />
      <ds:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256" />
      <ds:Reference URI="">
        <ds:Transforms>
          <ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature" />
        </ds:Transforms>
        <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
        <ds:DigestValue> Value not specified in the example </ds:DigestValue>
      </ds:Reference>
    </ds:SignedInfo> Value not specified in the example </ds:SignatureValue>
    <ds:KeyInfo>
      <ds:X509Data> Value not specified in the example </ds:X509Certificate>
    </ds:X509Data>
    </ds:KeyInfo>
  </ds:Signature>
  <orderId>a423edabac1943109818f6140f437423</orderId>
  <SCA_Type>1</SCA_Type>
</psd2acc:InformationResponse>
```

### 3.3.3 Account Information Result

The test environment is different from the production environment by a different URL. If you transition from a test environment to a production environment, you need to change the URL. To access the bank's environment, use only the following URL:

URL environment test: <https://apitest.365.bank/ws/AccountService/1.0/informationResult>

Production environment URL: <https://api.365.bank/ws/AccountService/1.0/informationResult>

#### 3.3.3.1 /ws/AccountService/1.0/informationResult

Transport	HTTPS/XML
Method	POST

MIME Type	application/xml
MEP	Synchronous request-response
XSD schema	AccountService_api.xsd

Request message	
HTTP Header parameters	Accept application/xml
HTTP Content	InformationResultRequest

Response message	
HTTP Header parameters	error Error code errorDescription Error description
HTTP Content	InformationResultResponse

### 3.3.3.1.1 REQUEST MESSAGE - INFORMATIONRESULTREQUEST

Element name	Required	Type	Description
orderId	yes	xsd:string	Unique order id
messageId	yes	xsd:string	Unique identifier of TPP message

Example of a request from the bank's environment test:

Header:

POST <http://apitest.365.bank/Psd2Mock/ws/AccountService/1.0/informationResult HTTP/1.1> (do not use this URL to access the bank's environment)  
 Accept-Encoding: gzip,deflate  
 Content-Type: application/xml  
 Accept: application/xml  
 Content-Length: 237  
 Host: apitest.365.bank  
 Connection: Keep-Alive  
 User-Agent: Apache-HttpClient/4.1.1 (java 1.5)

Body:

```
<acc:InformationResultRequest xmlns:acc="http://pabk.sk/EAI/API/XSD/PSD2/AccountService">
  <orderId> a423edabac1943109818f6140f437423</orderId>
  <messageId>94e9284a-2b1b-4197-a6b8-f430d45ba96b</messageId>
</acc:InformationResultRequest
```

### 3.3.3.1.2 RESPONSE MESSAGE - INFORMATIONRESULTRESPONSE

Element name	Required	Type	Description
accountName	yes	xsd:string	Account name
accountType	yes	xsd:string	Account type 0 – current account 1 – saving account
accountCurrency	yes	xsd:string	Account currency

accountingBalance	yes	xsd:decimal	Accounting balance
availableBalance	yes	xsd:decimal	Available balance
ownFunds	yes	xsd:decimal	Own funds
datetime	yes	xsd:date	Time of the balances snapshot

Example of a positive response from a bank environment test

HTTP/1.1 200 OK

Content-Length: 2810

Content-Type: application/xml; charset=UTF-8

Server:

X-Content-Type-Options: nosniff

error: 0

Date: Tue, 07 Jun 2022 07:45:24 GMT

```
<psd2acc:InformationResultResponsexmlns:psd2acc="http://pabk.sk/EAI/API/XSD/PSD2/AccountService">
  <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
    <ds:SignedInfo>
      <ds:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315" />
      <ds:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256" />
      <ds:Reference URI="">
        <ds:Transforms>
          <ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature" />
        </ds:Transforms>
        <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
        <ds:DigestValue> Value not specified in the example </ds:DigestValue>
      </ds:Reference>
    </ds:SignedInfo> Value not specified in the example </ds:SignatureValue>
    <ds:KeyInfo>
      <ds:X509Data>
        <ds:X509Certificate> Value not specified in the example </ds:X509Certificate>
      </ds:X509Data>
    </ds:KeyInfo>
  </ds:Signature>
  <accountName>AB Tester</accountName>
  <accountType>0</accountType>
  <accountCurrency>EUR</accountCurrency>
  <accountingBalance>33.00</accountingBalance>
  <availableBalance>26.00</availableBalance>
  <ownFunds>33.00</ownFunds>
  <datetime>2022-01-10T00:00:00</datetime>
</psd2acc:InformationResultResponse>
```

### 3.4 Request submission process LOT

The figure below describes the sequence of API calls that must be followed in that order. Failure to comply with the call sequence may result in the bank rejecting the request.

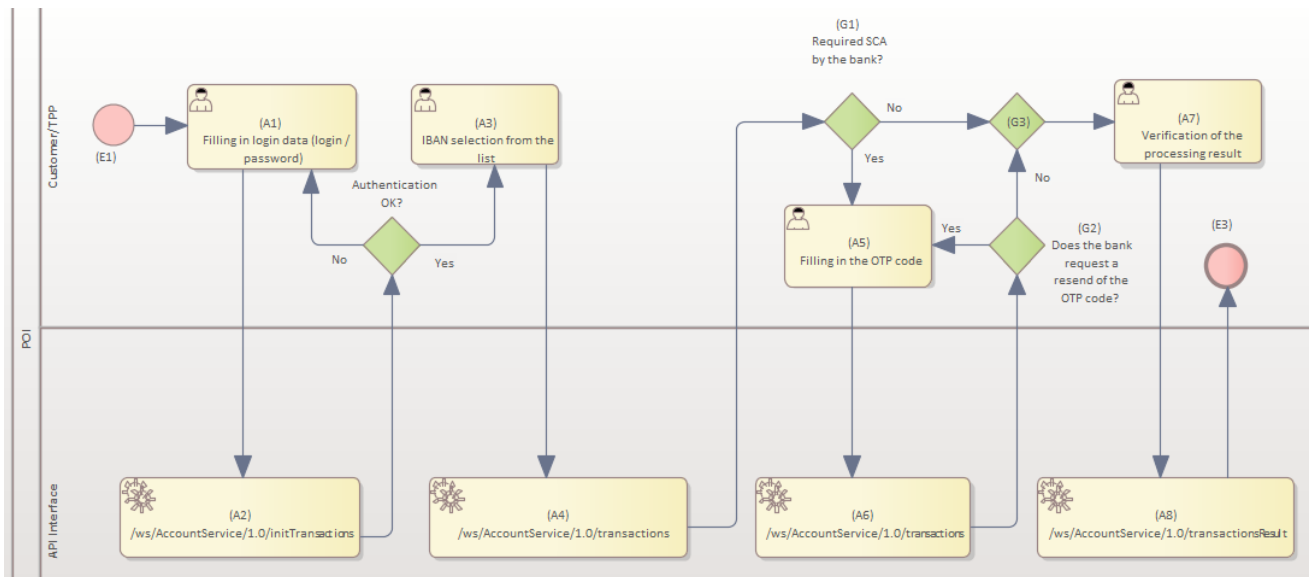


Figure 3: Sequence of call steps for LOT service

Process:

- (A1) The client enters his login details (login and password) with the TPP. After they are retrieved, the TPP calls the initTransactions service.
  - For the test environment, the login data listed in chapter 3.2 must be used
- (A2) After verifying the accuracy of the data, the bank responds with a list of accounts available to the client. The service response also returns the "orderId" attribute. This is a unique identifier that will be used throughout the process.
- (A3) The client defines an account from the list. After selecting a TPP account, it calls the transactions service with added account information and assigned orderID.
- (A4) The service returns in response information on whether authorization is required.
  - SCA is required
    - o A5 (A5) The TPP must request an OTP code from the client to continue. The code is sent by calling the transaction service, where the request must contain the same values as in point (A4)
      - DD - day of sending the request – numeric value
      - MM – month of sending the request – numeric value
      - HH – hour of sending the request – numeric value
      - Example: Request (A4) was sent on 27.5.2022 at 15:53. In request (A6) we expect OTP code 270515
    - o A6 (A6) The service returns authorization information in response.
    - o SCA not required - the request entry process is completed. In this case, the request (A6) is skipped and the TPP can proceed to verify the processing result. The Bank applies an exemption from the SCA pursuant to RTS Article 10. The Bank may request an SCA based on a request security assessment.
- (A7) To verify the processing result and display the account detail, it is necessary for TPP to call the transactionsResult service
- (A8) The service response returns the required data.

#### 3.4.1 IBAN Listing for List Of Transactions (LILOT)

The test environment is different from the production environment by a different URL. If you transition from a test environment to a production environment, you need to change the URL. To access the bank's environment, use only the following URL:

URL test environment: <https://apitest.365.bank/ws/AccountService/1.0/initTransactions>

URL production environment: <https://api.365.bank/ws/AccountService/1.0/initTransactions>

### 3.4.1.1 /ws/AccountService/1.0/initTransactions

Transport	HTTPS/XML
Method	POST
MIME Type	application/xml
MEP	Synchronous request-response
XSD schema	AccountService_api.xsd

Request message		
Accept	application/xml	
HTTP Header parameters	X-username	Client login to Internet banking
	X-password	Client password to Internet banking
HTTP Content	InitTransactionsRequest	

Response message		
HTTP Header parameters	error	Error code
	errorDescription	Error description
HTTP Content	InitTransactionsResponse	

#### 3.4.1.1.1 REQUEST MESSAGE - INITTRANSACTIONSREQUEST

Element type	Mandatory	Type	Description
messageld	yes	xsd:string	Unique identifier of TPP message

Example of a request from the bank's environment test:

Header:

POST <http://apitest.365.bank/Psd2Mock/ws/AccountService/1.0/initTransactions> HTTP/1.1 (do not use this URL to access the bank's environment)

Accept-Encoding: gzip,deflate

Content-Type: application/xml

X-password:

p76WQVzQ8y3oy0bApjjh6ln21ojuRkJZAX8LmFWo7FMDsmJHtqLICr6gv0ybk4fK6ISQse50z/ym  
b5nXNRVuwWQt+0rUUTcvB9nYPv3X1Wr9opOz0JPI0HhBd1R+h23AwCPft8CCXBvW3Vad5Es  
sPaygRIMr4xgiWmcyenOKZQ70Ki5L1mUfMZ0ZsljDKrelZ2iv+UziCCviyavO1+ufR1bha9pHNKqY  
HSB/aTOX6nlYImN0KaoD86nl/6x/IDSTZo9SO642eVWePN2NPd9NCQxO+7ObYFES3UZFrDz  
612n8ZhkZzIVvSS5gXtUI0UvQ/fdtVxeEyfymZGDpOQRcyw==

Accept: application/xml

X-username: A1psd2365@gmailabc.com

Content-Length: 181

Host: apitest.365.bank

Connection: Keep-Alive

User-Agent: Apache-HttpClient/4.1.1 (java 1.5)

Body:

<acc:InitTransactionsRequest xmlns:acc="http://pabk.sk/EAI/API/XSD/PSD2/AccountService">

```
<messageId>5638102a-3218-47a2-b4ba-9780c3063ad7</messageId>
</acc:InitTransactionsRequest>
```

### 3.4.1.1.2 RESPONSE MESSAGE - INITTRANSACTIONSRESPONSE

Element name	Type	Description
orderId	xsd:string	Unique order id
accountList		List of accounts
=id	xsd:string	IBAN (international bank account number)

Example of a positive response from a bank environment test

HTTP/1.1 200 OK

Content-Type: application/xml; charset=UTF-8

Server:

X-Content-Type-Options: nosniff

error: 0

Date: Fri, 10 Jun 2022 09:29:20 GMT

Content-Length: 2655

```
<psd2acc:InitTransactionsResponse xmlns:psd2acc="http://pabk.sk/EAI/API/XSD/PSD2/AccountService">
  <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
    <ds:SignedInfo>
      <ds:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315" />
      <ds:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256" />
      <ds:Reference URI="">
        <ds:Transforms>
          <ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature" />
        </ds:Transforms>
        <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
        <ds:DigestValue> Value not specified in the example </ds:DigestValue>
      </ds:Reference>
    </ds:SignedInfo> Value not specified in the example </ds:SignatureValue>
    <ds:KeyInfo>
      <ds:X509Data> Value not specified in the example </ds:X509Certificate>
    </ds:KeyInfo>
  </ds:Signature>
  <orderId>0a048510407443abaa629f9b6541a679</orderId>
  <accountList>
    <id>SK936500000000096824759</id>
  </accountList>
</psd2acc:InitTransactionsResponse>
```

### 3.4.2 List Of Transactions (LOT)

The service described in this chapter is request in two parts of the process. The first time if the client has chosen an account (this request is always mandatory) and the second time if the bank requests the implementation of SCA (The second request can be repeated several times, it depends on whether the OTP code was sent for the first time or not). All service requests must have the same completed part of the request, which is marked as "Body" in the example (messageId may vary). Requests differ in the header because the first request does not contain the X-OTP attribute. The second request also contains this attribute.

The test environment is different from the production environment by a different URL. If you transition from a test environment to a production environment, you need to change the URL. To access the bank's environment, use only the following URL:



URL test environment: <https://apitest.365.bank/ws/AccountService/1.0/transactions>

Production environment URL: <https://api.365.bank/ws/AccountService/1.0/transactions>

### 3.4.2.1 /ws/AccountService/1.0/transactions

Transport	HTTPS/XML
Method	POST
MIME Type	application/xml
MEP	Synchronous request-response
XSD schema	AccountService_api.xsd

Request message		
HTTP Header parameters	Accept	application/xml
	X-orderId	orderId of operation, which has to be authorized
	X-OTPKod	OTP authorization code
HTTP Content	TransactionsRequest	

Response message		
HTTP Header parameters	error	Error code
	errorDescription	Error description
HTTP Content	TransactionResponse	

#### 3.4.2.1.1 REQUEST MESSAGE - TRANSACTIONSREQUEST

Element name	Required	Type	Description
IBAN	yes	xsd:string	Client's IBAN (international bank account number)
dateFrom	yes	xsd:date	Date of first transaction
dateTo	yes	xsd:date	Date of last transaction
transactionType	yes	xsd:string	Transaction type: pending realized
messageId	yes	xsd:string	Unique identifier of TPP message

Example of a request from the bank's environment test:

Header of the first request:

```
POST http://apitest.365.bank/Psd2Mock/ws/AccountService/1.0/transactions HTTP/1.1 (do not use this URL to access the bank's environment)
Accept-Encoding: gzip,deflate
Content-Type: application/xml
Accept: application/xml
X-orderId: 0a048510407443abaa629f9b6541a679
Content-Length: 373
```

Host: apitest.365.bank  
Connection: Keep-Alive  
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)

Header of the second request:

POST [http://apitest.365.bank/Psd2Mock/mockup\\_direct/ws/AccountService/1.0/transactions HTTP/1.1](http://apitest.365.bank/Psd2Mock/mockup_direct/ws/AccountService/1.0/transactions HTTP/1.1) (do not use this URL to access the bank's environment)  
Accept-Encoding: gzip,deflate  
Content-Type: application/xml  
Accept: application/xml  
X-orderId: 0a048510407443abaa629f9b6541a679  
X-OTPKod: 090208  
Content-Length: 373  
Host: apitest.365.bank  
Connection: Keep-Alive  
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)

Body:

```
<acc:TransactionsRequest xmlns:acc="http://pabk.sk/EAI/API/XSD/PSD2/AccountService">  
<IBAN>SK936500000000096824759</IBAN>  
<dateFrom>2022-01-01</dateFrom>  
<dateTo>2022-04-20</dateTo>  
<transactionType>realized</transactionType>  
<messageId>7927be29-48cf-4b90-ac6c-2b0ba2a321c1</messageId>  
</acc:TransactionsRequest>
```

#### 3.4.2.1.2 RESPONSE MESSAGE - TRANSACTIONRESPONSE

Element name	Required	Type	Description
orderId	yes	xsd:string	Unique order id
SCA_Type	yes	xsd:boolean	Attribute for requirement of additional authorisation . 1 – is required 0 – is not required

If Er27 is returned in the response header, it is a final error message and the order is rejected (eg multiple incorrectly entered verification codes). In this case, you need to start calling from the beginning.

Example of a positive response from a bank environment test

HTTP/1.1 200 OK  
Content-Length: 2599  
Content-Type: application/xml; charset=UTF-8  
Server:  
X-Content-Type-Options: nosniff  
error: 0  
Date: Fri, 10 Jun 2022 09:34:57 GMT

```
<psd2acc:TransactionsResponse xmlns:psd2acc="http://pabk.sk/EAI/API/XSD/PSD2/AccountService">  
<ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">  
<ds:SignedInfo>  
<ds:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315" />  
<ds:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256" />  
<ds:Reference URI="">  
<ds:Transforms>  
<ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature" />
```

```

</ds:Transforms>
<ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
<ds:DigestValue> Value not specified in the example </ds:DigestValue>
</ds:Reference>
</ds:SignedInfo> Value not specified in the example </ds:SignatureValue>
<ds:KeyInfo>
<ds:X509Data> Value not specified in the example </ds:X509Certificate>
</ds:X509Data>
</ds:KeyInfo>
</ds:Signature>
<orderId>0a048510407443abaa629f9b6541a679</orderId>
<SCA_Type>1</SCA_Type>
</psd2acc:TransactionsResponse>

```

### 3.4.3 Transactions result

The test environment is different from the production environment by a different URL. If you transition from a test environment to a production environment, you need to change the URL. To access the bank's environment, use only the following URL:

URL test environment: <https://apitest.365.bank/ws/AccountService/1.0/transactionsResult>

Production environment URL: <https://api.365.bank/ws/AccountService/1.0/transactionsResult>

#### 3.4.3.1 /ws/AccountService/1.0/transactionsResult

Transport	HTTPS/XML
Method	POST
MIME Type	application/xml
MEP	Synchronous request-response
XSD schema	AccountService_api.xsd

Request message		
HTTP Header parameters	Accept	application/xml
HTTP Content	TransactionsResultRequest	
Response message		
HTTP Header parameters	error	Error code
	errorDescription	Error description
HTTP Content	TransactionsResultResponse	

#### 3.4.3.1.1 REQUEST MESSAGE - TRANSACTIONSRESULTREQUEST

Element name	Required	Type	Description
orderId	yes	xsd:string	Unique order id
pageNr	yes	xsd:string	Page number
messageld	yes	xsd:string	Unique identifier of TPP message

Example of a request from the bank's environment test:

Header:

```
POST http://apitest.365.bank/Psd2Mock/ws/AccountService/1.0/transactionsResult_HTTP/1.1 (do not use
this URL to access the bank's environment)
Accept-Encoding: gzip,deflate
Content-Type: application/xml
Accept: application/xml
Content-Length: 259
Host: apitest.365.bank
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)
```

Body:

```
<acc:TransactionsResultRequest xmlns:acc="http://pabk.sk/EAI/API/XSD/PSD2/AccountService">
<orderId>0a048510407443abaa629f9b6541a679</orderId>
<pageNr>1</pageNr>
<messageId>d5f76cf5-0cd7-49a5-b334-33030222f8d8</messageId>
</acc:TransactionsResultRequest>
```

### 3.4.3.1.2 RESPONSE MESSAGE – TRANSACTIONSRESULTRESPONSE

Character = in column Element name represents one level in XML structure

Element name	Required	Type	Description
transactionList			
=pageNr	yes	xsd:integer	Actual page number
=totalPages	yes	xsd:integer	Total number of pages
=transaction	no		One transaction in page
==id	yes	xsd:string	Id of transaction
==type	yes	xsd:string	Operation type
==date	yes	xsd:date	Date of transaction
== amount	yes	xsd:string	Transaction amount
== reference	yes	xsd:decimal	Reference
== note	no	xsd:string	Note
==rowIDInPage	no	xsd:integer	Id of page number
==accountNumber	yes	xsd:string	Account number
==accountBankCode	no	xsd:string	Account bank code
==accountName	no	xsd:string	Account name
==correspondingAccountNumber	no	xsd:string	Counterparty account number
==correspondingAccountBankCode	no	xsd:string	Counterparty bank code

Example of a response from a bank environment test

```
HTTP/1.1 200 OK
Content-Length: 3244
Content-Type: application/xml; charset=UTF-8
```

Server:  
X-Content-Type-Options: nosniff  
error: 0  
Date: Tue, 07 Jun 2022 09:16:52 GMT

```
<psd2acc:TransactionResultResponse
xmlns:psd2acc="http://pabk.sk/EAI/API/XSD/PSD2/AccountService">
  <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
    <ds:SignedInfo>
      <ds:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315" />
      <ds:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256" />
      <ds:Reference URI="">
        <ds:Transforms>
          <ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature" />
        </ds:Transforms>
        <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmldsig#sha256" />
        <ds:DigestValue> Value not specified in the example </ds:DigestValue>
      </ds:Reference>
    </ds:SignedInfo> Value not specified in the example </ds:SignatureValue>
  <ds:KeyInfo>
    <ds:X509Data> Value not specified in the example </ds:X509Certificate>
  </ds:X509Data>
</ds:KeyInfo>
</ds:Signature>
<transactionList>
  <pageNr>1</pageNr>
  <totalPages>1</totalPages>
  <transaction>
    <id>367</id>
    <type>debit</type>
    <date>2022-01-04T00:00:00</date>
    <amount>-0.14</amount>
    <reference>new testID</reference>
    <note>detail test remittance information</note>
    <rowIDInPage>1</rowIDInPage>
    <accountNumber>SK936500000000096824759</accountNumber>
    <accountBankCode>POBNSKBAXXX</accountBankCode>
    <accountName>JK Tester</accountName>
    <correspondingAccountNumber>SK486500000000096735844</correspondingAccountNumber>
    <correspondingAccountBankCode>POBNSKBAXXX</correspondingAccountBankCode>
  </transaction>
</transactionList>
</psd2acc:TransactionResultResponse>
```

### 3.5 Request submission process POI

The figure below describes the sequence of API calls that must be followed in that order. Failure to comply with the call sequence may result in the bank rejecting the request.

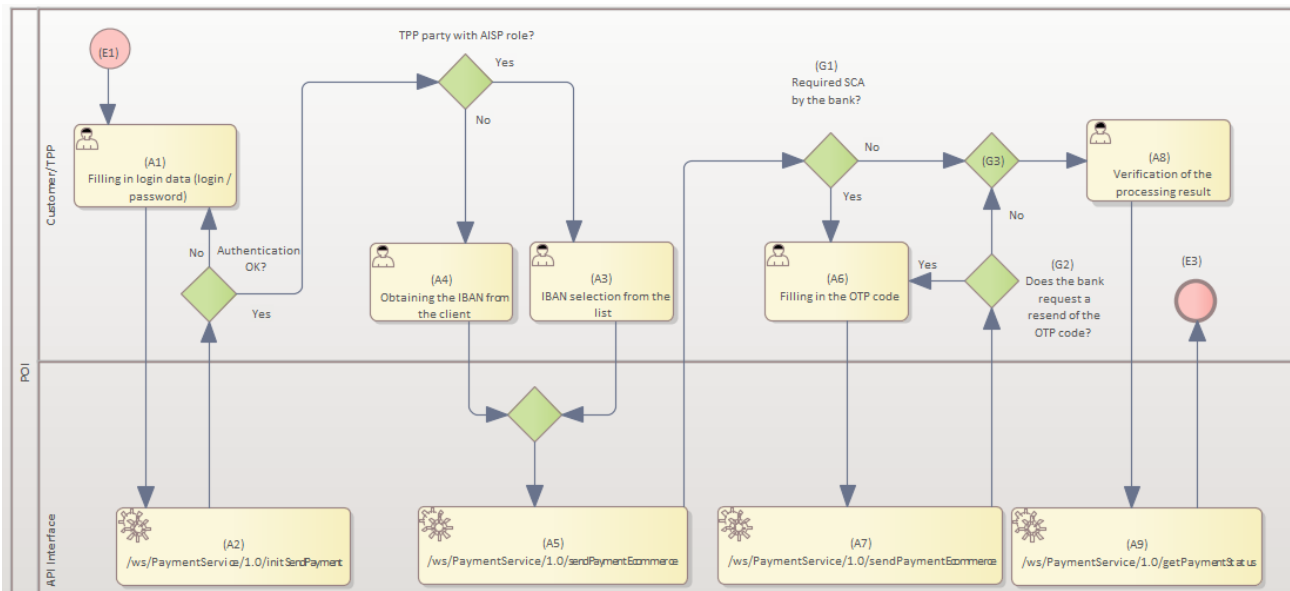


Figure 4: Sequence of call steps for POI service

#### Process:

- (A1) The client enters his login details (login and password) with the TPP. After they are obtained, TPP requests the service
  - initPayment: initialization service for scenarios:
    - sendPaymentEcommerce: instant payment (SEPA)
    - sendPaymentStandard: payment with forward due date (SEPA)
  - initStandingOrder initialization service for the script:
    - createStandingOrder: standing order creation service (SEPA)
  - For the test environment, the login data listed in chapter 3.2 must be used
- (A2) After validating the data, the service response returns the "orderId" attribute. It is a unique identifier that will be used throughout the process.
  - If the TPP party also has an AISP role, we return the list of accounts available to the client in the service response
  - In the event that the TPP party does not have the AISP role (it only has the PISP/PIISP role), the service does not return a list of client accounts
- (A3) The client defines the account from the list (the TPP party also has the role of AISP). After choosing an account, TPP calls the service with information supplemented by the account, assigned orderId.
  - The testing environment has a defined list of payee accounts that TPP can use for testing. If the account from the list is not used, the request will be rejected. The list of accounts is also returned in the Init response of the service.
- (A4 TPP party without AISP role) The client enters the IBAN of the third party, then the TPP party calls the service with information supplemented by the account and assigned orderId.
  - The testing environment has a defined list of payee accounts that TPP can use for testing. If the account from the list is not used, the request will be rejected. The list of accounts is not returned in the Init response of the service, so it is necessary to use the account from chapter 3.2. The account corresponding to the used login must be selected.
- (A5) In the response, the service returns information about whether authorization needs to be carried out. The information is given in the TxSts attribute.
  - The bank requests the sending of SCA (TxSts = SCA)
    - (A6) TPP must request an OTP code from the client to continue. The code is sent by calling the service, while the request must contain the same values as in point (A5). The OTP code is sent in a separate attribute in the call header. The OTP for the test environment must be sent in the format DDMMHH
      - DD – day of sending the request to the service that requested SCA in the response – numerical value
      - MM – month of sending the service request, which requested SCA in the response – numerical value
      - HH – hour of sending the service request, which requested SCA in the response – numeric value
 Example: Request (A5) was sent on 5/27/2022 at 3:53 p.m. In the call (A7), we expect the OTP code 270515
    - (A7) The service returns information about the execution of authorization in the response. If an error is returned in the header but TxSts = SCA, it means that the OTP code could not be verified. The order is still valid, the bank is waiting for the correct OTP code to be sent.

- SCA not required - the request entry process is completed. In this case, the call (A7) is skipped and the TPP can proceed to verify the processing result.
  - TxSts returned one of the following values:
    - Request processing states:
      - COMP = payment is made
      - WAIT = payment has been received for processing. This status returns mainly with the sendPaymentStandard service if the due date is in the future.
      - RJCT = payment is declined. This is the final state and if it occurs, it is necessary to enter the request from the beginning.
  - Technical conditions:
    - RCVD = the request is being processed, it is the technical status of the request processing on the bank's side, if the following value is returned, it is necessary to verify the status change via the getPaymentStatus service.
    - ERR = this is a technical error, it is necessary to contact the bank to verify the error
- (A8) To verify the processing result and display the account detail, it is necessary for the TPP to call the getPaymentStatus service
- (A9) The service response returns the required data.

### 3.5.1 IBAN Listing for Payment Order Initiation (LIPOI) and for Standing Order

The test environment is different from the production environment by a different URL. If you transition from a test environment to a production environment, you need to change the URL. To access the bank's environment, use only the following URL:

URL test environment:

initPayment: <https://apitest.365.bank/ws/PaymentService/1.0/initPayment>

initStandingOrder: <https://apitest.365.bank/ws/PaymentService/1.0/initStandingOrder>

Production environment URL:

initPayment: <https://api.365.bank/ws/PaymentService/1.0/initPayment>

initStandingOrder: <https://api.365.bank/ws/PaymentService/1.0/initStandingOrder>

#### 3.5.1.1 /ws/AccountService/1.0/initPayment

Transport	HTTPS/XML
Method	POST
MIME Type	application/xml
MEP	Synchronous request-response
XSD schema	PaymentService_api.xsd

Request message		
	Accept	application/xml
HTTP Header parameters	X-username	Client login to Internet banking
	X-password	Client password to Internet banking
HTTP Content	initPaymentRequest	

Response message		
HTTP Header parameters	error	Error code
	errorDescription	Error description
HTTP Content	initPaymentResponse	

### 3.5.1.1.1 REQUEST MESSAGE - INITTRANSACTIONSREQUEST

Element name	Required	Type	Description
messageld	Yes	xsd:string	Unique identifier of TPP message

Example of a request from the bank's environment test:

Header:

POST <http://apitest.365.bank/Psd2Mock/ws/PaymentService/1.0/initPayment HTTP/1.1> (do not use this URL to access the bank's environment)

Accept-Encoding: gzip,deflate

Content-Type: application/xml

X-password:

p76WQVzQ8y3oy0bApjjh6ln21ojuRkJZAX8LmFWo7FMDsmJHtqLICr6gv0ybk4fK6ISQse50z/ymb5nXNRVu  
wWQt+0rUUTcvB9nYPv3X1Wr9opOz0JPI0HhBd1R+h23AwCPft8CCXBvW3Vad5EssPaygRIMr4xgiWmcy  
nOKZQ70Ki5L1mUfMZ0ZsljDKrelZ2iv+UziCCviyavO1+ufR1bha9pHNKqYHSB/aTOX6nYImN0KaoD86nI/6  
x/IDSTZo9SO642eVWePN2NPd9NCQxO+7ObYFES3UZFrDz612n8ZhkZzIVvSS5gXtUI0UvQ/fdtVxeEyfym  
ZGDpOQRcyw==

Accept: application/xml

X-username: A1psd2365@gmailabc.com

Content-Length: 183

Host: apitest.365.bank

Connection: Keep-Alive

User-Agent: Apache-HttpClient/4.1.1 (java 1.5)

Body:

```
<psd2pay:InitPaymentRequest xmlns:psd2pay="http://pabk.sk/EAI/API/XSD/PSD2/PaymentService">
  <messageld>913d5900-48b0-4a66-9c1b-4da92a77a2bb</messageld>
</psd2pay:InitPaymentRequest>
```

### 3.5.1.1.2 RESPONSE MESSAGE - INITTRANSACTIONSRESPONSE

Element name	Type	Description
orderId	xsd:string	Unique order id
accountList		List of accounts (The collection is returned only if the TPP party also has an AISP role)
=id	xsd:string	IBAN (international bank account number)

An example of a positive answer from the bank's test environment in the case where the TPP party also has an AISP role.

HTTP/1.1 200 OK

Content-Length: 2645

Content-Type: application/xml; charset=UTF-8

Server:

X-Content-Type-Options: nosniff

error: 0

Date: Fri, 10 Jun 2022 11:03:22 GMT

```
<psd2pay:InitPaymentResponse xmlns:psd2pay="http://pabk.sk/EAI/API/XSD/PSD2/PaymentService">
  <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
    <ds:SignedInfo>
      <ds:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315" />
      <ds:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256" />
      <ds:Reference URI="">
```



```

<ds:Transforms>
  <ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature" />
</ds:Transforms>
<ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
<ds:DigestValue> Value not specified in the example </ds:DigestValue>
</ds:Reference>
</ds:SignedInfo> Value not specified in the example </ds:SignatureValue>
<ds:KeyInfo>
  <ds:X509Data> Value not specified in the example </ds:X509Certificate>
</ds:X509Data>
</ds:KeyInfo>
</ds:Signature>
<orderId>fc8004b48aef4d9daa08ae490cef8871</orderId>
<accountList>
  <id>SK936500000000096824759</id>
</accountList>
</psd2pay:InitPaymentResponse>

```

An example of a positive response from the bank's test environment in case the TPP party does not have an AISP role.

```

HTTP/1.1 200 OK
Content-Length: 2645
Content-Type: application/xml; charset=UTF-8
Server:
X-Content-Type-Options: nosniff
error: 0
Date: Fri, 10 Jun 2022 11:03:22 GMT

```

```

<psd2pay:InitPaymentResponse xmlns:psd2pay="http://pabk.sk/EAI/API/XSD/PSD2/PaymentService">
  <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
    <ds:SignedInfo>
      <ds:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315" />
      <ds:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256" />
      <ds:Reference URI="">
        <ds:Transforms>
          <ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature" />
        </ds:Transforms>
        <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
        <ds:DigestValue> Value not shown in the example </ds:DigestValue>
        </ds:Reference>
      </ds:SignedInfo> Value not shown in the example </ds:SignatureValue>
    <ds:KeyInfo>
      <ds:X509Data> Value not shown in the example </ds:X509Certificate>
    </ds:X509Data>
    </ds:KeyInfo>
  </ds:Signature>
  <orderId>fc8004b48aef4d9daa08ae490cef8871</orderId>
</psd2pay:InitPaymentResponse>

```

### 3.5.1.2 /ws/AccountService/1.0/initStandingOrder

Transport	HTTPS/XML
Method	POST
MIME Type	application/xml
MEP	Synchronous request-response
XSD schema	PaymentService_api.xsd

Request message		
	Accept	application/xml
HTTP Header parameters	X-username	Client login to Internet banking
	X-password	Client password to Internet banking
HTTP Content	initStandingOrderRequest	
Response message		
HTTP Header parameters	error	Error code
	errorDescription	Error description
HTTP Content	initStandingOrderResponse	

### 3.5.1.2.1 REQUEST MESSAGE - INITSTANDINGORDERREQUEST

Element	Mandatory	Type	Description
messageId	yes	xsd:string	Unique identifier of TPP message

Example of a request from the bank's environment test:

Header:

POST [http://apitest.365.bank/Psd2Mock/ws/PaymentService/1.0/initStandingOrder\\_HTTP/1.1](http://apitest.365.bank/Psd2Mock/ws/PaymentService/1.0/initStandingOrder_HTTP/1.1) (do not use this URL to access the bank's environment)

Accept-Encoding: gzip, deflate

Content-Type: application/xml

X-password:

p76WQVzQ8y3oy0bApjjh6ln21ojuRkJZAX8LmFWo7FMDsmJHtqLICr6gv0ybk4fK6ISQse50z/ymb5nXNRVu  
wWQt+0rUUTcvB9nYPv3X1Wr9opOz0JPI0HhBd1R+h23AwCPft8CCXBvW3Vad5EssPaygRIMr4xgiWmcyc  
nOKZQ70Ki5L1mUfMZ0ZsljDKrelZ2iv+UziCCviyavO1+ufR1bha9pHNKqYHSB/aTOX6nlYImN0KaoD86nl/6  
x/IDSTZo9SO642eVWePN2NPd9NCQxO+7ObYFES3UZFrDz612n8ZhkZzIVvSS5gXtUI0UvQ/fdtVxeEyfym  
ZGDpOQRcyw==

Accept: application/xml

X-username: A1psd2365@gmailabc.com

Content-Length: 195

Host: apitest.365.bank

Connection: Keep-Alive

User-Agent: Apache-HttpClient/4.1.1 (java 1.5)

Body:

```
<psd2pay:InitStandingOrderRequest xmlns:psd2pay="http://pabk.sk/EAI/API/XSD/PSD2/PaymentService">
  <messageId>83b99667-68f1-424f-963f-0464a5596651</messageId>
</psd2pay:InitStandingOrderRequest>
```

### 3.5.1.2.2 RESPONSE MESSAGE - INITSTANDINGORDERRESPONSE

Element	Type	Description
orderId	xsd:string	Unique order id
accountList		List of accounts (The collection is returned only if the TPP party also has an AISP role)
=id	xsd:string	IBAN (international bank account number)

An example of a positive answer from the bank's test environment in the case where the TPP party also has an AISP role.

HTTP/1.1 200 OK  
Content-Length: 2657  
Content-Type: application/xml; charset=UTF-8  
Server:  
X-Content-Type-Options: nosniff  
error: 0  
Date: Fri, 10 Jun 2022 11:10:37 GMT

```
<psd2pay:InitStandingOrderResponse
xmlns:psd2pay="http://pabk.sk/EAI/API/XSD/PSD2/PaymentService">
  <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
    <ds:SignedInfo>
      <ds:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315" />
      <ds:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256" />
      <ds:Reference URI="">
        <ds:Transforms>
          <ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature" />
        </ds:Transforms>
        <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#sha256" />
        <ds:DigestValue> Value not specified in the example </ds:DigestValue>
      </ds:Reference>
    </ds:SignedInfo> Value not specified in the example </ds:SignatureValue>
  </ds:Signature>
  <ds:KeyInfo>
    <ds:X509Data> Value not specified in the example </ds:X509Certificate>
  </ds:KeyInfo>
</orderId>a74bd53a140a42d199be5548268a66fd</orderId>
<accountList>
  <id>SK936500000000096824759</id>
</accountList>
</psd2pay:InitStandingOrderResponse>
```

An example of a positive response from the bank's test environment in case the TPP party does not have an AISP role.

HTTP/1.1 200 OK  
Content-Length: 2657  
Content-Type: application/xml; charset=UTF-8  
Server:  
X-Content-Type-Options: nosniff  
error: 0  
Date: Fri, 10 Jun 2022 11:10:37 GMT

```
<psd2pay:InitStandingOrderResponse
xmlns:psd2pay="http://pabk.sk/EAI/API/XSD/PSD2/PaymentService">
  <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
    <ds:SignedInfo>
      <ds:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315" />
      <ds:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256" />
      <ds:Reference URI="">
        <ds:Transforms>
          <ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature" />
        </ds:Transforms>
        <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#sha256" />
        <ds:DigestValue> Value not specified in the example </ds:DigestValue>
      </ds:Reference>
    </ds:SignedInfo> Value not specified in the example </ds:SignatureValue>
  </ds:Signature>
  <ds:KeyInfo>
```

```

    <ds:X509Data> Value not specified in the example </ds:X509Certificate>
  </ds:X509Data>
</ds:KeyInfo>
</ds:Signature>
<orderId>a74bd53a140a42d199be5548268a66fd</orderId>
</psd2pay:InitStandingOrderResponse>

```

### 3.5.2 Payment Order Initiation (POI)

Interface to be used to payment initiation. Based on the payment type there are 3 endpoints provided:

- sendPaymentEcommerce - Ecommerce payment (SEPA)
- sendPaymentStandard – payment with future payment date (SEPA)
- createStandingOrder – create standing order (SEPA)

The services described in this chapter are called in two parts of the process. The first time if the client has chosen an account (this call is always mandatory) and the second time if the bank requests the implementation of SCA (The second call can be repeated several times, it depends on whether the OTP code was sent for the first time or not). All service calls must have the same part of the request, which is marked as "Points" in the example (the value of messageId may vary). Calls differ in the header because the first call does not contain the X-OTP attribute, so the second call already has this attribute.

The test environment is different from the production environment by a different URL. If you transition from a test environment to a production environment, you need to change the URL. To access the bank's environment, use only the following URL:

Test environment URL:

sendPaymentEcommerce: <https://apitest.365.bank/ws/PaymentService/1.0/sendPaymentEcommerce>

sendPaymentStandard: <https://apitest.365.bank/ws/PaymentService/1.0/sendPaymentStandard>

createStandingOrder: <https://apitest.365.bank/ws/PaymentService/1.0/createStandingOrder>

Production environment URL:

sendPaymentEcommerce: <https://api.365.bank/ws/PaymentService/1.0/sendPaymentEcommerce>

sendPaymentStandard: <https://api.365.bank/ws/PaymentService/1.0/sendPaymentStandard>

createStandingOrder: <https://api.365.bank/ws/PaymentService/1.0/createStandingOrder>

#### 3.5.2.1 /ws/PaymentService/1.0/sendPaymentEcommerce

Transport	HTTPS/XML
Method	POST
MIME Type	application/xml
MEP	Synchronous request-response
Request XSD schema	<p>pain.001.001.07.PaBkCustomPOlecomm.xsd</p> <p>Customised PAIN001 scheme is used as payload. See PAIN001 for element type specification.</p>
Response XSD schema	<p>pain.002.001.08.xsd</p> <p>Customised PAIN002 scheme is used as payload. See PAIN002 for element type specification.</p>

Request message	
HTTP Header parameters	Accept application/xml

	X-orderId	orderId of operation, which has to be authorized
	X-OTPkod	OTP authorization code
HTTP Content	SendPaymentEcommerceRequest	
<b>Response message</b>		
HTTP Header parameters	error	Error code
	errorDescription	Error description
HTTP Content	SendPaymentEcommerceResponse	

### 3.5.2.1.1 REQUEST MESSAGE - SENDPAYMENTECOMMERCEREQUEST

Character = in column Element name represents one level in XML structure

Element name	Required	Type	Description
CstmrCdtTrfInItN	yes		
=GrpHdr	yes		
==MsgId	yes	Max35Text	Unique identifier of TPP message No diacritics or space is allowed. Only characters are allowed: A-Za-z0-9+?/-(:),.'
==CreDtTm	yes	ISODatetime	Date and time that the request was created
=PmtInf	yes		
==Dbtr	no		
===Nm	no	Max140Text	Name of the payer/originator No diacritic is allowed. Only characters are allowed: A-Za-z0-9+?/-(:),.' and space
==DbtrAcct	yes		
===Id	yes		
====IBAN	yes	IBAN2007Identifier	IBAN of the Payer/Originator that will be debited
==CdtTrfTxInf	yes		
===PmtId	yes		
====InstrId	yes	Max35Text	Transaction ID No diacritics or space is allowed. Only characters are allowed: A-Za-z0-9+?/-(:),.'
====EndToEndId	yes	Max35Text	E2E reference – max 35 characters No diacritics is allowed. Only characters are

			allowed: A-Za-z0-9+?/-(:),.' and space If the client does not fill the value, the TPP sends NOTPROVIDED
=== Amt	yes		
==== InstdAmt	yes	ActiveOrHistoricCurrencyAndAmount	Amount of transaction and currency. Only EUR currency possible. The amount must be >= 0.01 and <= 999999999.99. The amount must be rounded to only 2 decimal places. Decimal places separator is "." (dot)
===Cdtr	yes		
====Nm	yes	Max140Text	Creditor name No diacritics is allowed. Only characters are allowed: A-Za-z0-9+?/-(:),.' and space
===CdtrAcct	yes		
====Id	yes		
====IBAN	yes	IBAN2007Identifier	Creditor IBAN
===RmtInf	no		
====Ustrd	no	Max140Text	Remittance Information - Free text provided for information purposes No diacritics is allowed. Only characters are allowed: A-Za-z0-9+?/-(:),.' and space

Example of a request from the bank's environment test:

Header of the first request:

```
POST http://apitest.365.bank/Psd2Mock/ws/PaymentService/1.0/sendPaymentEcommerce
HTTP/1.1 (do not use this URL to access the bank's environment)
Accept-Encoding: gzip,deflate
Content-Type: application/xml
Accept: application/xml
X-orderId: 64078544a5384160ab56ca4b67d5476f
Content-Length: 1256
Host: apitest.365.bank
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)
```

Header of the second request:

POST <http://apitest.365.bank/Psd2Mock/ws/PaymentService/1.0/sendPaymentEcommerceHTTP/1.1> (do not use this URL to access the bank's environment)  
Accept-Encoding: gzip,deflate  
Content-Type: application/xml  
Accept: application/xml  
X-orderId: 64078544a5384160ab56ca4b67d5476f  
X-OTPKod: 080613  
Content-Length: 1256  
Host: apitest.365.bank  
Connection: Keep-Alive  
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)

Body:

```
<pay:SendPaymentEcommerceRequest
xmlns:pay="http://pabk.sk/EAI/API/XSD/PSD2/PaymentService"
xmlns:urn="urn:iso:std:iso:2002:tech:xsd:pain.001.001.07.PaBkCustomPOlecomm">
  <urn:CstmrCdtTrfIntrn>
    <urn:GrpHdr>
      <urn:MsgId>MsgId</urn:MsgId>
      <urn:CreDtTm>2022-06-08T13:43:06.309+02:00</urn:CreDtTm>
    </urn:GrpHdr>
    <urn:PmtInf>
      <urn:DbtrAcct>
        <urn:Id>
          <urn:IBAN>SK9365000000000096824759</urn:IBAN>
        </urn:Id>
      </urn:DbtrAcct>
      <urn:CdtTrfTxInf>
        <urn:PmtId>
          <urn:InstrId>thx</urn:InstrId>
          <urn:EndToEndId>e2e</urn:EndToEndId>
        </urn:PmtId>
        <urn:Amt>
          <urn:InstdAmt Ccy="EUR">0.02</urn:InstdAmt>
        </urn:Amt>
        <urn:CdtrAgt>
          <urn:FinInstnId>
            </urn:FinInstnId>
          </urn:CdtrAgt>
          <urn:Cdtr>
            <urn:Nm>crdtname</urn:Nm>
          </urn:Cdtr>
          <urn:CdtrAcct>
            <urn:Id>
              <urn:IBAN>SK7565000000000096735940</urn:IBAN>
            </urn:Id>
          </urn:CdtrAcct>
          <urn:RmtInf>
            <urn:Ustrd>remittance</urn:Ustrd>
          </urn:RmtInf>
        </urn:CdtTrfTxInf>
      </urn:PmtInf>
    </urn:CstmrCdtTrfIntrn>
  </pay:SendPaymentEcommerceRequest>
```

### 3.5.2.1.2 RESPONSE MESSAGE - SENDPAYMENTECOMMERCERESPONSE

Element name	Required	Type	Description
CstmrPmtStsRpt	yes		
=GrpHdr	yes		
==MsgId	yes	Max35Text	Response message identification
==CreDtTm	yes	ISODateTime	Date and time when the response was created
=OrgnlGrpInfAndSts	yes		
==OrgnlMsgId	yes	Max35Text	Copy of field GrpHdr/MsgId from Request
==OrgnlMsgNmId	yes	Max35Text	pain.001.001.07 – constant value
=OrgnlPmtInfAndSts	no		
==OrgnlPmtInfId	yes	Max35Text	Copy of field PmtInf/CdtTrfTxInf/PmtId/InstrId from Request
==TxInfAndSts	no		
===OrgnlInstrId	no	Max35Text	Order Id - a unique reference assigned by the bank to uniquely identify the order.
===TxSts	no	ExternalPaymentTransactionStatus1Code	RCVD = payment is processing ERR = error occurred COMP = payment processing completed WAIT = payment is waiting for processing SCA = additional authorization is required
=== StsRsnInf	no		
====Rsn	no		
====Cd	yes	ExternalStatusReason1Code	Reason code

Example of a response from a bank environment test

HTTP/1.1 200 OK

Content-Length: 3515

Content-Type: application/xml; charset=UTF-8

Server:

X-Content-Type-Options: nosniff

error: 0

Date: Wed, 08 Jun 2022 11:43:06 GMT

```
<?xml version="1.0" encoding="utf-16"?>
<psd2pay:SendPaymentEcommerceResponse
xmlns:pain02="urn:iso:std:iso:20022:tech:xsd:pain.002.001.08"
xmlns:psd2pay="http://pabk.sk/EAI/API/XSD/PSD2/PaymentService">
  <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
    <ds:SignedInfo>
```



```

    <ds:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-
20010315" />
    <ds:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256" />
    <ds:Reference URI="">
    <ds:Transforms>
    <ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature" />
    </ds:Transforms>
    <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
    <ds:DigestValue> Value not specified in the example </ds:DigestValue>
    </ds:Reference>
</ds:SignedInfo> Value not specified in the example </ds:SignatureValue>
<ds:KeyInfo>
    <ds:X509Data> Value not specified in the example </ds:X509Certificate>
    </ds:X509Data>
</ds:KeyInfo>
</ds:Signature>
<pain02:CstmrPmtStsRpt>
<pain02:GrpHdr>
    <pain02:MsgId>b5c1c788-df8c-446d-8dce-59899da8251d</pain02:MsgId>
    <pain02:CreDtTm>2022-06-08T13:43:06.3146091+02:00</pain02:CreDtTm>
</pain02:GrpHdr>
<pain02:OrgnlGrplnfAndSts>
    <pain02:OrgnlMsgId>MsgId</pain02:OrgnlMsgId>
    <pain02:OrgnlMsgNmId>pain.001.001.07</pain02:OrgnlMsgNmId>
</pain02:OrgnlGrplnfAndSts>
<pain02:OrgnlPmtInfAndSts>
    <pain02:OrgnlPmtInflId>thx</pain02:OrgnlPmtInflId>
<pain02:TxInfAndSts>
    <pain02:OrgnlInstrId>64078544a5384160ab56ca4b67d5476f</pain02:OrgnlInstrId>
    <pain02:TxSts>SCA</pain02:TxSts>
<pain02:StsRsnInf>
    <pain02:Rsn>
    <pain02:Cd>00</pain02:Cd>
    </pain02:Rsn>
</pain02:StsRsnInf>
</pain02:TxInfAndSts>
</pain02:OrgnlPmtInfAndSts>
</pain02:CstmrPmtStsRpt>
</psd2pay:SendPaymentEcommerceResponse>

```

### 3.5.2.2 /ws/PaymentService/1.0/sendPaymentStandard

Transport	HTTPS/XML
Method	POST
MIME Type	application/xml
MEP	Synchronous request-response
Request XSD schema	pain.001.001.07.PaBkCustomPOlecomm.xsd Customised PAIN001 scheme is used as payload. See PAIN001 for element type specification.
Response XSD schema	pain.002.001.08.xsd Customised PAIN002 scheme is used as payload. See PAIN002 for element type specification.

Request message		
HTTP Header parameters	Accept	application/xml
	X-orderId	orderId of operation, which has to be authorized
	X-OTPKod	OTP authorization code
HTTP Content	SendPaymentStandardRequest	

Response message		
HTTP Header parameters	error	Error code
	errorDescription	Error description
HTTP Content	SendPaymentStandardResponse	

#### 3.5.2.2.1 REQUEST MESSAGE - SENDPAYMENTSTANDARDREQUEST

Character = in column Element name represents one level in XML structure

Element name	Required	Type	Description
CstmrCdtTrfInItN	yes		
=GrpHdr	yes		
==MsgId	yes	Max35Text	Unique identifier of TPP message No diacritics or space is allowed. Only characters are allowed: A-Za-z0-9+?/-(:),.'
==CreDtTm	yes	ISODateTime	Date and time that the request was created
=PmtInf	yes		
==ReqdExctnDt	yes	ISODate	Requested execution date
==Dbtr	no		
===Nm	no	Max140Text	Name of the payer/originator

			No diacritics is allowed. Only characters are allowed: A-Za-z0-9+?/-(:),', and space
==DbtrAcct	yes		
===Id	yes		
====IBAN	yes	IBAN2007Identifier	IBAN of the Payer/Originator that will be debited
==CdtTrfTxInf	yes		
===PmtId	yes		
====InstrId	yes	Max35Text	Transaction ID No diacritics or space is allowed. Only characters are allowed: A-Za-z0-9+?/-(:),',.
====EndToEndId	yes	Max35Text	E2E reference – maximum 35 characters are allowed No diacritics is allowed. Only characters are allowed: A-Za-z0-9+?/-(:),', and space If the client does not fill the value, the TPP sends NOTPROVIDED
=== Amt	yes		
==== InstdAmt	yes	ActiveOrHistoricCurrencyAndAmount	Amount of transaction and currency. Only EUR currency possible. The amount must be > = 0.01 and <= 999999999.99. The amount must be rounded to only 2 decimal places. Decimal places separator is "." (dot)
===Cdtr	yes		
====Nm	yes	Max140Text	Creditor name No diacritic is allowed. Only characters are allowed: A-Za-z0-9+?/-(:),', and space

===CdrAcct	yes		
====Id	yes		
=====IBAN	yes	IBAN2007Identifier	Creditor IBAN
===RmtInf	no		
====Ustrd	no	Max140Text	Remittance Information - free text provided for information purposes No diacritics is allowed. Only characters are allowed: A-Za-z0-9+?/-(:),.' and space

Example of a request from the bank's environment test:

Header of the first request:

```
POST http://apitest.365.bank/Psd2Mock/ws/PaymentService/1.0/sendPaymentStandardHTTP/1.1 (do not use this URL to access the bank's environment)
Accept-Encoding: gzip,deflate
Content-Type: application/xml
Accept: application/xml
X-orderId: 463940df8b664b6390f7607528d8f188
Content-Length: 1681
Host: apitest.365.bank
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)
```

Header of the second request:

```
POST http://apitest.365.bank/Psd2Mock/ws/PaymentService/1.0/sendPaymentStandardHTTP/1.1 (do not use this URL to access the bank's environment)
Accept-Encoding: gzip,deflate
Content-Type: application/xml
Accept: application/xml
X-orderId: 463940df8b664b6390f7607528d8f188
X-OTPKod: 200414
Content-Length: 1742
Host: apitest.365.bank
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)
```

Body:

```
<pay:SendPaymentStandardRequest
xmlns:pay="http://pabk.sk/EAI/API/XSD/PSD2/PaymentService"
xmlns:urn="urn:iso:std:iso:20022:tech:xsd:pain.001.001.07.PaBkCustomPOIstd">
  <urn:CstmrCdtTrfInIt>
    <urn:GrpHdr>
      <urn:MsgId>MsgId</urn:MsgId>
      <urn:CreDtTm>2022-06-08T13:55:37.615+02:00</urn:CreDtTm>
    </urn:GrpHdr>
    <urn:PmtInf>
      <urn:ReqdExctnDt>2022-06-10</urn:ReqdExctnDt>
    </urn:Dbtr>
```

```

    <urn:Nm>dbtrname</urn:Nm>
  </urn:Dbtr>
  <urn:DbtrAcct>
    <urn:Id>
      <urn:IBAN>SK9365000000000096824759</urn:IBAN>
    </urn:Id>
  </urn:DbtrAcct>
  <urn:CdtTrfTxInf>
    <urn:PmtId>
      <urn:InstrId>thx</urn:InstrId>
      <urn:EndToEndId>e2e</urn:EndToEndId>
    </urn:PmtId>
    <urn:Amt>
      <urn:InstdAmt Ccy="EUR">2.00</urn:InstdAmt>
    </urn:Amt>
    <urn:ChrgBr>SHA</urn:ChrgBr>
    <urn:CdtrAgt>
      <urn:FinInstnId>
        <urn:BICFI>COBADEFFXXX</urn:BICFI>
      </urn:FinInstnId>
    </urn:CdtrAgt>
    <urn:Cdtr>
      <urn:Nm>crdtr</urn:Nm>
      <urn:PstlAdr>
        <urn:StrtNm>Rosse Strasse 10</urn:StrtNm>
        <urn:TwnNm>Berlin</urn:TwnNm>
        <urn:CtrySubDvsn>Nemecko</urn:CtrySubDvsn>
        <urn:Ctry>DE</urn:Ctry>
      </urn:PstlAdr>
    </urn:Cdtr>
    <urn:CdtrAcct>
      <urn:Id>
        <urn:IBAN>SK4765000000000096735959</urn:IBAN>
      </urn:Id>
    </urn:CdtrAcct>
    <urn:RmtInf>
      <urn:Ustrd>remittance</urn:Ustrd>
    </urn:RmtInf>
  </urn:CdtTrfTxInf>
</urn:PmtInf>
</urn:CstmrCdtTrfInIt>
</pay:SendPaymentStandardRequest>

```

### 3.5.2.2.2 RESPONSE MESSAGE - SENDPAYMENTSTANDARDRESPONSE

Element name	Required	Type	Description
CstmrPmtStsRpt	yes		
=GrpHdr	yes		
==MsgId	yes	Max35Text	Response message identification
==CreDtTm	yes	ISODateTime	Date and time when the response was created
=OrgnlGrpInfAndSts	yes		

==OrgnlMsgld	yes	Max35Text	Copy of field GrpHdr/Msgld from Request
==OrgnlMsgNmld	yes	Max35Text	pain.001.001.07 – constant value
=OrgnlPmtInfAndSts	no		
==OrgnlPmtInfld	yes	Max35Text	Copy of field PmtInf/CdtTrfTxInf/PmtId/InstrId from Request
==TxInfAndSts	no		
===OrgnlInstrld	no	Max35Text	Order Id
===TxSts	no	ExternalPaymentTransactionStatus1Code	RCVD = prebieha spracovanie platby ERR = chyba COMP = platba bola spracovana WAIT = platba bola prijata na spracovanie SCA = potrebné dodatočná autorizácia platby
=== StsRsnInf	no		
====Rsn	no		
=====Cd	yes	ExternalStatusReason1Code	reason code

Example of a response from a bank environment test  
 HTTP/1.1 200 OK  
 Content-Length: 3513  
 Content-Type: application/xml; charset=UTF-8  
 Server:  
 X-Content-Type-Options: nosniff  
 error: 0  
 Date: Wed, 08 Jun 2022 11:55:37 GMT

```
<?xml version="1.0" encoding="utf-16"?>
<psd2pay:SendPaymentStandardResponse
xmlns:pain02="urn:iso:std:iso:20022:tech:xsd:pain.002.001.08"
xmlns:psd2pay="http://pabk.sk/EAI/API/XSD/PSD2/PaymentService">
  <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
    <ds:SignedInfo>
      <ds:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315" />
      <ds:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256" />
      <ds:Reference URI="">
        <ds:Transforms>
          <ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature" />
        </ds:Transforms>
        <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
        <ds:DigestValue> Value not specified in the example </ds:DigestValue>
      </ds:Reference>
    </ds:SignedInfo> Value not specified in the example </ds:SignatureValue>
    <ds:KeyInfo>
      <ds:X509Data> Value not specified in the example </ds:X509Certificate>
    </ds:X509Data>
  </ds:KeyInfo>
```

```

</ds:Signature>
<pain02:CstmrPmtStsRpt>
  <pain02:GrpHdr>
    <pain02:MsgId>38e43231-a03e-42e5-9fe2-0328940de268</pain02:MsgId>
    <pain02:CreDtTm>2022-06-08T13:55:37.7002613+02:00</pain02:CreDtTm>
  </pain02:GrpHdr>
  <pain02:OrgnlGrpInfAndSts>
    <pain02:OrgnlMsgId>MsgId</pain02:OrgnlMsgId>
    <pain02:OrgnlMsgNmId>pain.001.001.07</pain02:OrgnlMsgNmId>
  </pain02:OrgnlGrpInfAndSts>
  <pain02:OrgnlPmtInfAndSts>
    <pain02:OrgnlPmtInfId>thx</pain02:OrgnlPmtInfId>
    <pain02:TxInfAndSts>
      <pain02:OrgnlInstrId>463940df8b664b6390f7607528d8f188</pain02:OrgnlInstrId>
      <pain02:TxSts>SCA</pain02:TxSts>
      <pain02:StsRsnInf>
        <pain02:Rsn>
          <pain02:Cd>00</pain02:Cd>
        </pain02:Rsn>
      </pain02:StsRsnInf>
    </pain02:TxInfAndSts>
  </pain02:OrgnlPmtInfAndSts>
</pain02:CstmrPmtStsRpt>
</psd2pay:SendPaymentStandardResponse>

```

### 3.5.2.3 /ws/PaymentService/1.0/getPaymentStatus

Transport	HTTPS/XML
Method	POST
MIME Type	application/xml
MEP	Synchronous request-response
Request XSD schema	PaymentService_api.xsd
Response XSD schema	PaymentService_api.xsd

Request message	
HTTP Header parameters	Accept application/xml
HTTP Content	GetPaymentStatusRequest

Response message	
HTTP Header parameters	error Error code
	errorDescription Error description
HTTP Content	GetPaymentStatusResponse

#### 3.5.2.3.1 REQUEST MESSAGE - GETPAYMENTSTATUSREQUEST

Character = in column Element name represents one level in XML structure

Element name	Required	Type	Description
PaymentStatus			

=GrpHdr	yes		
==MsgId	yes	xsd:string	Unique identifier of TPP message
==CreDtTm	yes	xsd:dateT ime	Date and time that the request was created
=Order	yes		
==OrderId	yes	xsd:string	Order id

Example of a request from the bank's environment test:

Header:

POST <http://apitest.365.bank/Psd2Mock/ws/PaymentService/1.0/getPaymentStatus> HTTP/1.1  
(do not use this URL to access the bank's environment)

Accept-Encoding: gzip,deflate

Content-Type: application/xml

Accept: application/xml

Content-Length: 505

Host: apitest.365.bank

Connection: Keep-Alive

User-Agent: Apache-HttpClient/4.1.1 (java 1.5)

Body:

```
<psd2pay:GetPaymentStatusRequest
xmlns:psd2pay="http://pabk.sk/EAI/API/XSD/PSD2/PaymentService"
xmlns:req="http://pabk.sk/EAI/API/XSD/PSD2/POI/RequestStatus">
  <req:PaymentStatus>
    <req:GrpHdr>
      <req:MsgId>5d7d49be-74ee-4851-8d78-2e8267869494</req:MsgId>
      <req:CreDtTm>2022-06-08T14:09:52.090+02:00</req:CreDtTm>
    </req:GrpHdr>
    <req:Order>
      <req:OrderId>a25fd9d70e214dfa9550a5da89815f03</req:OrderId>
    </req:Order>
  </req:PaymentStatus>
</psd2pay:GetPaymentStatusRequest>
```

### 3.5.2.3.2 RESPONSE MESSAGE - GETPAYMENTSTATUSRESPONSE

Element name	Required	Type	Description
CstmrPmtStsRpt	yes		
=GrpHdr	yes		
==MsgId	yes	Max35Text	Response message identification
==CreDtTm	yes	ISODateTime	Date and time when the response was created
=OrgnlGrpInfAndSts	yes		
==OrgnlMsgId	yes	Max35Text	Copy of field GrpHdr/MsgId from Request
==OrgnlMsgNmId	yes	Max35Text	pain.001.001.07 – constant value



=OrgnPmtInfAndSts	no		
==OrgnPmtInflId	yes	Max35Text	Copy of field PmtInf/CdtTrfTxInf/PmtId/InstrId from Request
==TxInfAndSts	no		
===OrgnInstrId	no	Max35Text	Order Id - a unique reference assigned by the bank to uniquely identify the order.
===TxSts	no	ExternalPaymentTransactionStatus1Code	RCVD = payment is processing ERR = error occurred COMP = payment processing completed WAIT = payment is waiting for processing SCA = additional authorization is required
===StsRsnInf	no		
====Rsn	no		
====Cd	yes	ExternalStatusReason1Code	reason code

Example of a response from a bank environment test

HTTP/1.1 200 OK

Content-Length: 3516

Content-Type: application/xml; charset=UTF-8

Server:

X-Content-Type-Options: nosniff

error: 0

Date: Wed, 08 Jun 2022 12:09:52 GMT

```
<?xml version="1.0" encoding="utf-16"?>
<psd2pay:GetPaymentStatusResponse
xmlns:pain02="urn:iso:std:iso:20022:tech:xsd:pain.002.001.08"
xmlns:psd2pay="http://pabk.sk/EAI/API/XSD/PSD2/PaymentService">
  <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
    <ds:SignedInfo>
      <ds:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315" />
      <ds:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256" />
      <ds:Reference URI="">
        <ds:Transforms>
          <ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature" />
        </ds:Transforms>
        <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmldsig#sha256" />
        <ds:DigestValue> Value not specified in the example </ds:DigestValue>
      </ds:Reference>
    </ds:SignedInfo> Value not specified in the example </ds:SignatureValue>
    <ds:KeyInfo>
      <ds:X509Data> Value not specified in the example </ds:X509Certificate>
    </ds:KeyInfo>
  </ds:Signature>
  <pain02:CstmrPmtStsRpt>
    <pain02:GrpHdr>
      <pain02:MsgId>440c78c5-2712-4a27-92fa-7987af9fbade</pain02:MsgId>
```

```

    <pain02:CreDtTm>2022-06-08T14:09:52.1789077+02:00</pain02:CreDtTm>
  </pain02:GrpHdr>
  <pain02:OrgnlGrpInfAndSts>
    <pain02:OrgnlMsgId>5d7d49be-74ee-4851-8d78-2e8267869494</pain02:OrgnlMsgId>
    <pain02:OrgnlMsgNmId>pain.001.001.07</pain02:OrgnlMsgNmId>
  </pain02:OrgnlGrpInfAndSts>
  <pain02:OrgnlPmtInfAndSts>
    <pain02:OrgnlPmtInfId />
    <pain02:TxInfAndSts>
      <pain02:OrgnlInstrId>a25fd9d70e214dfa9550a5da89815f03</pain02:OrgnlInstrId>
      <pain02:TxSts>SCA</pain02:TxSts>
      <pain02:StsRsnInf>
        <pain02:Rsn>
          <pain02:Cd>Er15</pain02:Cd>
        </pain02:Rsn>
      </pain02:StsRsnInf>
    </pain02:TxInfAndSts>
  </pain02:OrgnlPmtInfAndSts>
</pain02:CstmrPmtStsRpt>
</psd2pay:GetPaymentStatusResponse>

```

### 3.5.2.4 /ws/PaymentService/1.0/createStandingOrder

Transport	HTTPS/XML
Method	POST
MIME Type	application/xml
MEP	Synchronous request-response
Request XSD schema	PaymentService_api.xsd
Response XSD schema	PaymentService_api.xsd

Request message		
HTTP Header parameters	Accept	application/xml
	X-orderId	orderId of the specific operation that TPP wants to authorize
	X-OTPKod	OTP code for verification of authorization
HTTP Content	CreateStandingOrderRequest	

Response message		
HTTP Header parameters	error	Error code
	errorDescription	Description of error
HTTP Content	CreateStandingOrderResponse	

#### 3.5.2.4.1 REQUEST MESSAGE - CREATESTANDINGORDERREQUEST

Character = in column Element name represents one level in XML structure

Element name	Required	Type	Description
StdOrderPmt			

= GrpHdr	yes		
== MsgId	yes	Max35Text	Unique TPP message identifier No accents or spaces are allowed. Only characters are allowed: A-Za-z0-9+?/-(:),'
== CreDtTm	yes	ISODateTime	Date and time the message was sent from the TPP
= PmtInf	yes		
== DbtrAcct	yes		
=== Id			
==== IBAN		IBAN2007Identifier	IBAN (international bank account number)
== PmtId	no		
=== EndToEndId	yes	Max35Text	E2E reference - validation syntactically to max 35 characters Accents are not allowed. Only characters are allowed: A-Za-z0-9+?/-(:),' and space
== Amt	yes		
=== InstdAmt	yes	ActiveOrHistoricCurrencyAndAmount	Transfer order amount and currency. Currency only EUR The amount must be >=0.01 and <=999999999.99. The amount must be rounded to 2 decimal places only. The decimal separator is "." (dot)
== Cdtr	yes		
=== Nm	yes	Max140Text	The name of the creditor Accents are not allowed. Only characters are allowed: A-Za-z0-9+?/-(:),' and space
== CdtrAcct	yes		
=== Id	yes		
==== IBAN	yes	IBAN2007Identifier	IBAN (international bank account number)
== RmtInf	no		
=== Ustrd	yes	Max140Text	Payer's reference Accents are not allowed. Only characters are allowed: A-Za-z0-9+?/-(:),' and space
== Detail	yes		
=== Periodicity	yes	integer	Only one type of periodicity allowed. Allowed values: 2 (daily) 4 (monthly) 5 (quarterly) 6 (semi-annually) 7 (annually) 8 (the end of the month)
=== FirstPaymentDt	yes	date	Date must be at least D+1 and may not be greater than the 28th day of the month
=== LastPaymentDt	no	date	future date maximum 31.12.2099, which must be greater than the date of the first payment

Example of a request from the bank's environment test:

Header of the first request:

```
POST http://apitest.365.bank/Psd2Mock/ws/PaymentService/1.0/createStandingOrder HTTP/1.1
(do not use this URL to access the bank's environment)
Accept-Encoding: gzip,deflate
Content-Type: application/xml
Accept: application/xml
X-orderId: a09576e5151b409a9d0e81b79faef806
Content-Length: 1180
Host: apitest.365.bank
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)
```

Header of the second request:

```
POST http://apitest.365.bank/Psd2Mock/ws/PaymentService/1.0/createStandingOrder HTTP/1.1
(do not use this URL to access the bank's environment)
Accept-Encoding: gzip,deflate
Content-Type: application/xml
Accept: application/xml
X-orderId: a09576e5151b409a9d0e81b79faef806
X-OTPKod: 200414
Content-Length: 1180
Host: apitest.365.bank
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)
```

Body:

```
<psd2pay:CreateStandingOrderRequest
  xmlns:psd2pay="http://pabk.sk/EAI/API/XSD/PSD2/PaymentService">
  <StdOrderPmt>
    <GrpHdr>
      <MsgId>MsgId</MsgId>
      <CreDtTm>2022-06-08T16:18:37.109+02:00</CreDtTm>
    </GrpHdr>
    <PmtInf>
      <DbtrAcct>
        <Id>
          <IBAN>SK9365000000000096824759</IBAN>
        </Id>
      </DbtrAcct>
      <PmtId>
        <EndToEndId>e2e</EndToEndId>
      </PmtId>
      <Amt>
        <InstdAmt Ccy="EUR">1000000.00</InstdAmt>
      </Amt>
      <Cdtr>
        <Nm>cdtr</Nm>
      </Cdtr>
      <CdtrAcct>
        <Id>
          <IBAN>SK2265000000000096735924</IBAN>
        </Id>
      </CdtrAcct>
      <RmtInf>
```

```

    <Ustrd>remittance</Ustrd>
  </RmtInf>
  <Detail>
    <Periodicity>4</Periodicity>
    <FirstPaymentDt>2022-06-10</FirstPaymentDt>
    <LastPaymentDt>2022-08-09</LastPaymentDt>
  </Detail>
</PmtInf>
</StdOrderPmt>
</psd2pay:CreateStandingOrderRequest>

```

### 3.5.2.4.2 RESPONSE MESSAGE - CREATESTANDINGORDERRESPONSE

Element name	Required	Type	Description
OrgnlInstrId	Yes		
TxSts	yes		RCVD = payment is processing ERR = error occurred COMP = payment processing completed WAIT = payment is waiting for processing SCA = additional authorization is required

Example of a response from a bank environment test

```

HTTP/1.1 200 OK
Content-Length: 2754
Content-Type: application/xml; charset=UTF-8
Server:
X-Content-Type-Options: nosniff
error: 0
Date: Wed, 08 Jun 2022 14:18:37 GMT

```

```

<?xml version="1.0" encoding="utf-16"?>
<psd2pay:CreateStandingOrderResponse
xmlns:pain02="urn:iso:std:iso:20022:tech:xsd:pain.002.001.08"
xmlns:psd2pay="http://pabk.sk/EAI/API/XSD/PSD2/PaymentService">
  <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
    <ds:SignedInfo>
      <ds:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315" />
      <ds:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256" />
      <ds:Reference URI="">
        <ds:Transforms>
          <ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature" />
        </ds:Transforms>
        <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
        <ds:DigestValue> Value not specified in the example </ds:DigestValue>
      </ds:Reference>
    </ds:SignedInfo> Value not specified in the example </ds:SignatureValue>
    <ds:KeyInfo>
      <ds:X509Data> Value not specified in the example </ds:X509Certificate>
    </ds:X509Data>
  </ds:KeyInfo>
</ds:Signature>
<psd2pay:OrgnlInstrId>a09576e5151b409a9d0e81b79faef806</psd2pay:OrgnlInstrId>
<psd2pay:TxSts>SCA</psd2pay:TxSts>
</psd2pay:CreateStandingOrderResponse>

```



### 3.6 Payment Instrument Issuer Service Provider (PIISP)

To successfully proceed of this scenario, it is necessary that client of 365.bank provides forward authorization for selected TPP to validate his account balances.

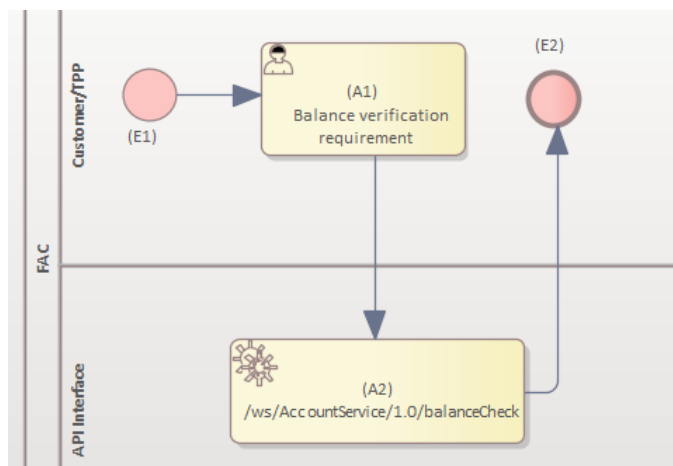


Figure 5 Orchestration PIISP

#### Sequence diagram description:

- Client puts inquiry (PIISP) via TPP
- TPP invokes PIISP API on API GW
- 365.bank verifies TPP and process inquiry online
- Result is presented to client via TPP

The test environment is different from the production environment by a different URL. If you transition from a test environment to a production environment, you'll need to change the URL. To access the bank's environment, use only the following URL:

Environment URL testing: <https://apitest.365.bank/ws/AccountService/1.0/balanceCheck>

Production environment URL: <https://api.365.bank/ws/AccountService/1.0/balanceCheck>

#### 3.6.1 /ws/AccountService/1.0/balanceCheck

Transport	HTTPS/XML
Method	POST
MIME Type	application/xml
MEP	Synchronous request-response
XSD schema	AccountService_api.xsd

<b>Request message</b>	
HTTP Header parameters	Accept application/xml
HTTP Content	BalanceCheckRequest

<b>Response message</b>	
HTTP Header parameters	error Error code
	errorDescription Error description

## 3.6.1.1.1 REQUEST MESSAGE - BALANCECHECKREQUEST

Element name	Required	Type	Description
IBAN	yes	xsd:string	Client IBAN (international bank account number)
amount	yes	xsd:decimal	Amount balance to be checked
currency	yes	xsd:string	Currency
messageld	yes	xsd:string	Unique identifier of TPP message

Example of a request from the bank's environment test

Header:

POST [http://apitest.365.bank/Psd2Mock/ws/AccountService/1.0/balanceCheck\\_HTTP/1.1](http://apitest.365.bank/Psd2Mock/ws/AccountService/1.0/balanceCheck_HTTP/1.1) (do not use this URL to access the bank's environment)  
 Accept-Encoding: gzip,deflate  
 Content-Type: application/xml  
 X-SubjectDN: C=SK, L=Bratislava, O="Postova banka, a.s", CN=tpptest.postovabanka.sk  
 Accept: application/xml  
 Content-Length: 279  
 Host: apitest.365.bank  
 Connection: Keep-Alive  
 User-Agent: Apache-HttpClient/4.1.1 (java 1.5)

Body:

```
<psd2acc:BalanceCheckRequest xmlns:psd2acc="http://pabk.sk/EAI/API/XSD/PSD2/AccountService">
  <IBAN>SK936500000000096824759</IBAN>
  <amount>2000.99</amount>
  <currency>EUR</currency>
  <messageld>fd374d4d-d6dd-43c5-bd49-7eabd33636d1</messageld>
</psd2acc:BalanceCheckRequest>
```

## 3.6.1.1.2 RESPONSE MESSAGE - BALANCECHECKRESPONSE

Element name	Required	Type	Description
result	yes	xsd:boolean	Result of the check
dateTime	yes	xsd:dateTime	Date and time when check was performed

Example of a response from a bank environment test

HTTP/1.1 200 OK  
 Content-Length: 2602  
 Content-Type: application/xml; charset=UTF-8  
 Server:  
 X-Content-Type-Options: nosniff  
 error: 0  
 Date: Wed, 08 Jun 2022 14:48:54 GMT

```
<psd2acc:BalanceCheckResponse xmlns:psd2acc="http://pabk.sk/EAI/API/XSD/PSD2/AccountService">
  <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
    <ds:SignedInfo>
      <ds:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315" />
      <ds:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256" />
```



```
<ds:Reference URI="">
  <ds:Transforms>
    <ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature" />
  </ds:Transforms>
  <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
  <ds:DigestValue> Value not specified in the example </ds:DigestValue>
</ds:Reference>
</ds:SignedInfo> Value not specified in the example </ds:SignatureValue>
<ds:KeyInfo>
  <ds:X509Data> Value not specified in the example </ds:X509Certificate>
  </ds:X509Data>
</ds:KeyInfo>
</ds:Signature>
<result>>false</result>
<dateTime>2022-06-08T16:48:54.8798416+02:00</dateTime>
</psd2acc:BalanceCheckResponse>
```

## 4 List of error codes

In case of error code and errorDescription is returned in HTTP header. HTTP header parameters:

- error - Error code
- errorDescription – Error description

#	Error code	Error description
	Err_out_001	Unauthorized characters in the InstrId field. The request could not be processed. Please verify the correctness of the entered data.
	Err_out_002	Incorrect amount entered. The request could not be processed. Please verify the correctness of the entered data.
	Err_out_003	The request could not be processed due to an incorrect due date. Please verify the correctness of the entered data.
	Err_out_004	The request could not be processed for insufficient dispo balance.
	Err_out_005	Could not process request due to invalid validity date since. Please verify the correctness of the entered data.
	Err_out_006	The request could not be processed due to an incorrect maturity date. Please verify the correctness of the entered data.
	Err_out_007	The request could not be processed because of an incorrect IBAN format. Please verify the correctness of the entered data.
	Err_out_008	The request could not be processed due to incorrect maturity date. Please verify the correctness of the entered data.
	Err_out_009	The request could not be processed because the recipient bank is not a SEPA bank. Please verify the correctness of the entered data.
	Err_out_010	The request could not be processed, there is not enough money for the account. Please verify the correctness of the entered data.
	Err_out_011	The request could not be processed, the payer's details are not filled in correctly. Please verify the correctness of the entered data.
	Err_out_012	The request could not be processed. Can not make a transfer to the same account. Please verify the correctness of the entered data.
	Err_out_013	The request could not be processed. Please verify the correctness of the entered data, especially the existence of the account and its limitations.
	Err_out_014	The request could not be processed. Please verify the correctness of the entered data, especially the existence of the account and its limitations.
	Err_out_015	The request could not be processed. Please verify the correctness of the entered data, or try again later.
	Err_out_016	The command could not be processed. Please verify the correctness of the entered data, especially the existence of the account (beneficiary) and its limitations.
	Err_out_017	The request was received and now is processing.
	Err_out_018	The request could not be processed. Please verify the correctness of the entered data, especially the account format.
	Err_out_019	The request could not be processed. Please verify the correctness of the entered data, especially the IBAN and BIC of the recipient.
	Err_out_020	The request could not be processed. Please verify the correctness of the entered data, especially the IBAN of the recipient.
	Err_out_021	The request could not be processed. Please verify the correctness of the entered data, especially the beneficiary's bank code.
	Err_out_022	The request could not be processed. Please verify the correctness of the entered data.
	Err_out_023	The request could not be processed. Account is inactive.
	Err_out_024	The request could not be processed. There was a problem converting your IBAN account number. Please verify the correctness of the entered data.
	Err_out_025	IBAN recipient account is incorrect. Please verify the correctness of the entered data

#	Error code	Error description
	Err_out_026	Priority transfer in this amount is not possible.
	Err_out_027	The request could not be processed. The time for authorization has expired.
	Err_out_028	The request could not be processed. Request was not authorized.
	401 Unauthorized	Client certificate isn't owned by any registered TPP.
	403 Forbidden	TPP hasn't required role for this request.