APS Bank plc

PSD2 Open Banking

TPP Technical Documentation

Summary

Document Control Information

1. Document reference

PSD2 Open Banking – TPP Technical Documentation Summary

2. Document Type

Technical Document

3. Synopsis

This document defines the technical specification of the PSD2 Open Banking solution that APS Bank has developed.

4. Document control

Version	Date	Description
1.0	14/09/2019	Documentation for PSD2 API v1
1.1	16/12/2019	Summary documentation. Updated structure and text of section 2.
1.2	07/01/2020	Minor changes in structure
1.3	14/01/2020	Applied comments from Compliance

Table of Contents

APS Open Banking Services

APS Bank has created the APS Development Portal that provides the Third Party Providers (TPPs)[1] an opportunity to increase innovation and broaden the range of services for their customer using APS APIs.

APS Bank offers TPPs access to accounts (XS2A) in a safe and efficient way using APIs in line with the Directive (EU) 2015/2366 (PSD2) and Commission Delegated Regulation EU 2018/389 with regard to regulatory standards for strong customer authentication and common and secure open standards of communication, and following the Berlin Group NextGenPSD2 XS2A Interoperability Framework - Implementation Guidelines. The latter framework allows APS Bank to implement the PSD2 required account information services (AIS) and payment initiation services (PIS) into a common, interoperable and harmonised interface and infrastructure.

With that purpose, APS Bank has made available for the TPPs two environments:

- Sandbox. We have made available a Sandbox Test environment, where you can experiment, develop and test your application for the Open Banking PSD2 Services using dummy data.
- Production. Once tested in our Sandbox, you will be able to start working with the production environment offering the Open Banking services to all your customers.

As a TPP, in order to start using our APIs and to have access to our Sandbox, you will need to follow these prerequisites:

- Be registered in APS Development Portal.
- Have a TPP valid eIDAS (electronic Identification, Authentication and trust Services) certificate issued by a QTSPs (Qualified Trust Service Provider). Please find more information below in section 1. API Security
- Have APS Bank review your identity and complete the verification checks.

[1] Among others [PSD2] contains regulations of new services to be operated by so called Third Party Payment Service Providers (TPP) on behalf of a Payment Service User (PSU). These new services are:
• Payment Initiation Service (PIS),
• Account Information Service (AIS)
• Confirmation of the Availability of Funds

To start using these APIs you need to become an approved Account Information Service Provider (AISP) or Payment Initiation Service Provider (PISP). Once you have completed the above enrolment process, our on boarding team will be on hand to help you get set up and start testing our APIs.

1. API Security

As mentioned in the Berlin Group implementation guide, eIDAS certification is used for QWAC in the transport layer and for QSeal in the application layer.

In this section, it describes how transport and application layer security is involved in the below steps. These steps secure the APIs used in APS Bank Open Banking services.

1.1 TPP Certificate Registration

An authorization of TPP has to be approved/rejected by the National Competent Authority (NCA[2]) that is responsible for payment services in their country. The following are the steps of PSP certificate registration:

1. The TPP submits a certificate that needs to be registered to the Qualified Trusted Service Provider (QTSP[3]).
2. The QTSP performs an identity check and validates PSD2 specific attributes using the information provided by NCA.
3. TPP receive the qualified certificate issued by the QTSP and approved by the NCA.

1.2 TLS and Qualified Certificates

The communication between the TPP and APS Bank is always secured by using a TLS-connection and is setup by the TPP.

The TLS-connection has to be established including TPP authentication. For this authentication, the TPP has to use a qualified certificate for website authentication issued by a qualified trust service provider QTSP according to the eIDAS regulation. The content of the certificate has to be compliant

[2] National competent authorities (NCA) are organisations that have the legally delegated or invested authority, or power to perform a designated function, normally monitoring compliance with the national statutes and regulations. EBA Competent Authorities list.

[3] According to eIDAS definition, a to Qualified Trusted Service Provider (QTSP) is a trust service provider who provides one or more qualified trust services and is granted the qualified status by the supervisory body.

with the requirements of [EBA-RTS]. The certificate of the TPP has to indicate all roles the TPP is authorised to use.

eIDAS Regulation (EU) N°910/2014 on electronic identification and trust services for electronic transactions in the internal market provides a predictable regulatory environment to enable secure and seamless electronic interactions between businesses, citizens and public authorities.

APS Bank Open Banking solution requires the QWAC certificates.

eIDAS Regulation defines the Qualified Website Authentication Certificates (QWAC) and Qualified Certificate for Electronic Seals (Qseal). These certificates are trusted certificates by ensuring the validity against certificate status services (Online Certificate Status Protocol (OCSP) and Certificate Revocation List (CRL)) provided by the QTSP. TPPs are fully responsible and liable for the revocation and renewal of their eIDAS certificates issued to them.

Certificates for Website Authentication (QWAC)

QWAC is used for two purposes: to establish a secure communication Transport Layer Security Encryption (TLS) and to identifying and authenticating the communicating parties (PISP, AISP and ASPSP), protecting the communication from potential attackers on the network.

APS Bank Open Banking solution requires the QWAC certificates and checks the validity using CRL according to PSD2 Directive, restricting the access to APIs it the validation is not successful.

2. Sandbox - Test Environment

Our Sandbox test environment allows TPPs to test against our PSD2 compliant XS2A Test Interface, which system is a copy of the production configuration in operational use.

We strongly recommend you to build and test your applications using our Sandbox test environment.

APS Bank offers a harmonized API endpoint. This allows us to provide you with a uniform test interface (Sandbox). The Sandbox simulates API responses as in a real production environment accessing to mock data. Resources will be generated dynamically in the Sandbox and will be a reflection of the production API system.

Once you have created an application into the Development Portal, you will be able to proceed with its integration into the testing environment.

Our technical documentation for the Sandbox provides the necessary information for you to interact with APS Bank XS2A test interface, such as:

- Access to Account Information Services (AIS) for accessing account information, balances and transactions including consent creation.
- Access to Payment Initiation Services (PIS) for initiating payments and checking the status of such payments.

3. Swagger UI

APS Bank has developed a Swagger UI, which allows you to visualize and interact with our API's resources. It will be available into our Sandbox environment that can be accessed from the following link once registration in the APS Development Portal has been completed and approved:

https://sbobapi.apsbank.com.mt/swagger/index.html

API endpoints are divided in:

1. Consents
2. Accounts
3. Funds confirmation
4. Payments

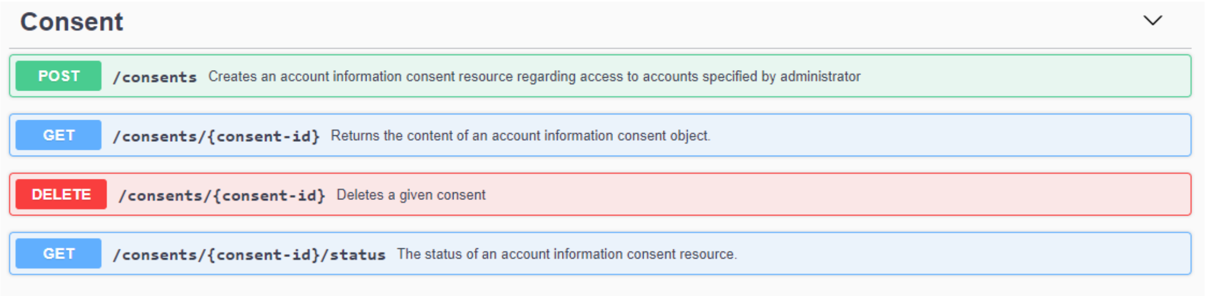
4. API Endpoints

4.1 Consents

The following text describes current solution for Open Banking PSD2 Consent API methods. APS Bank Consent interface is based on the "Consent Request without Indication of Accounts" according to the Berlin Group Guidelines.

APS Bank will agree bilaterally directly with the PSU on which accounts the requested access consent (accounts, balances and/or transactions) should be supported. This is authorised by the PSU with an SCA.

You will find the following structure into the APS Bank Swagger UI:



Consent

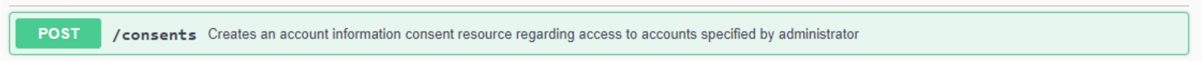
POST /consents Creates an account information consent resource regarding access to accounts specified by administrator

GET /consents/{consent-id} Returns the content of an account information consent object.

DELETE /consents/{consent-id} Deletes a given consent

GET /consents/{consent-id}/status The status of an account information consent resource.

POST/consents



POST /consents Creates an account information consent resource regarding access to accounts specified by administrator

Creates an account information consent resource regarding access to accounts specified by administrator.

When this Consent Request is a request where the 'recurringIndicator' equals true, and if it exists already a former consent for recurring access on account information for the addressed PSU submitted by this TPP, then the former consent automatically expires as soon as the new consent request is authorised by the PSU. Bank offered consents are only supported. This is why only elements in the request where sub-attributes 'access' is not present.

GET/consents/{consent-id}

Returns the content of an account information consent object. This is returning the data for the TPP for given consent resource.

DELETE/consents/{consent-id}

Deletes a given consent.

GET/consents/{consent-id}/status

The status of an account information consent resource. Provides the status for the given consent resource.

The following table describe the items used for each method:

Name	POST/ consents	GET/ consents/ {consent-id}	DELETE/ consents/ {consent-id}	GET/ consents/ {consent-id}/ status
request ** (body)	X			
TPP-Redirect-URI * string (header)	X			
TPP-Nok-Redirect-URI * string (header)	X			
consent-id * string (path)		X	X	X
APP-ID * string (header)	X	X	X	X
X-Request-ID * string (header)	X	X	X	X
PSU-IsCorporate * boolean (header)	X	X	X	X
PSU-IP-Address * string (header)	X	X	X	X
PSU-IP-Port string (header)	X	X	X	X

** Mandatory Parameter*

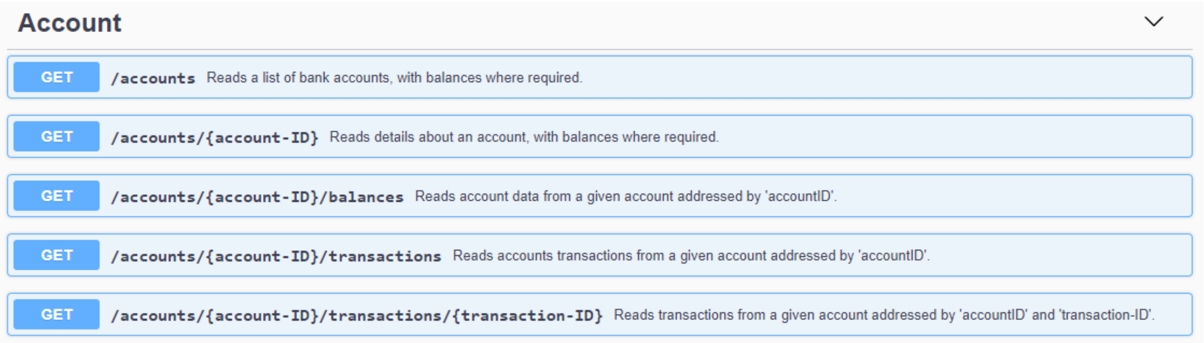
*** Mandatory Parameter. Request structure explained below explained in section 4.6*

4.2 Accounts

The following text describes current solution for Open Banking PSD2 Accounts Information methods.

These APIs provide the ability for approved/authorised account information service providers (AISPs) to access a customer’s (payment service user, PSU) account only when the PSU grants consent.

You will find the following structure into the APS Bank Swagger UI:



Account

GET /accounts Reads a list of bank accounts, with balances where required.

GET /accounts/{account-ID} Reads details about an account, with balances where required.

GET /accounts/{account-ID}/balances Reads account data from a given account addressed by 'accountID'.

GET /accounts/{account-ID}/transactions Reads accounts transactions from a given account addressed by 'accountID'.

GET /accounts/{account-ID}/transactions/{transaction-ID} Reads transactions from a given account addressed by 'accountID' and 'transaction-ID'.

GET/accounts

Reads a list of bank accounts, with balances where required.

It is assumed that a consent of the PSU to this access is already given and stored on the ASPSP system. The addressed list of accounts depends then on the PSU ID and the stored consent addressed by ‘consentID’, respectively the OAuth2 access token.

GET/accounts/{account-ID}

Reads details about an account, with balances where required.

It is assumed that a consent of the PSU to this access is already given and stored on the ASPSP system. The addressed details of this account depends then on the stored consent addressed by ‘consentID’ or respectively the OAuth2 access token. The ‘accountID’ can represent a multicurrency account. In this case, the currency code is set to 'XXX'.

GET/accounts/{account-ID}/balances

Reads account data from a given account addressed by 'accountID'. This 'accountID' then can be retrieved by calling endpoint: /accounts

GET/accounts/{account-ID}/transactions

Reads accounts transactions from a given account addressed by 'accountID'. This 'accountID' then can be retrieved by calling endpoint: /accounts

GET/accounts/{account-ID}/transactions/{transaction-ID}

Reads transactions from a given account addressed by 'accountID' and 'transaction-ID'. This 'accountID' then can be retrieved by calling endpoint: /accounts

The following table describes the items used for each method:

Name	GET/ accounts	GET/ accounts/ {account-ID}	GET/ accounts/ {account-ID}/ balances	GET/ accounts/ {account-ID}/ transactions	GET/ accounts/ {account-ID}/ transactions/ {transaction-ID}
account-ID * string (path)		X	X	X	X
transaction-ID * string (path)					X
Consent-ID * string (header)	X	X	X	X	X
Authorization * string (header)	X	X	X	X	X
APP-ID * string (header)	X	X	X	X	X
X-Request-ID * string (header)	X	X	X	X	X
PSU-IsCorporate * boolean (header)	X	X	X	X	X
PSU-IP-Address * string (header)	X	X	X	X	X
PSU-IP-Port string (header)	X	X	X	X	X
withBalance boolean (query)	X	X		X	
dateFrom string (query)				X	
dateTo string (query)				X	
bookingStatus string (query)				X	

** Mandatory Parameter*

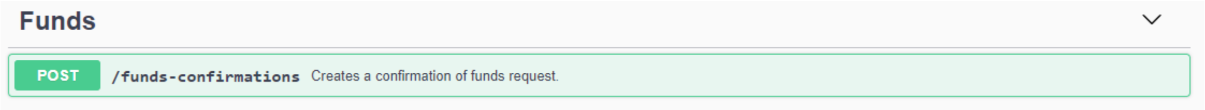
*** Mandatory Parameter. Request structure explained below explained in section 4.6*

4.3 Confirmation of Funds

The following text describes current solution for Open Banking PSD2 Confirmation of Funds method.

Confirmation of Funds request, allows you to confirm that the payments are available for the specific payment account.

You will find the following structure into the APS Bank Swagger UI:



POST/funds-confirmations

Creates a confirmation of funds request. Validate funds availability for the provided elements.

The following table describe the items used for each method:

Name	POST/ funds-confirmations
request ** (body) Example Value Model	X
APP-ID * string (header)	X
X-Request-ID * string (header)	X
PSU-IsCorporate * boolean (header)	X
PSU-IP-Address * string (header)	X
PSU-IP-Port string (header)	X

** Mandatory Parameter*

*** Mandatory Parameter. Request structure explained below explained in section 4.6*

4.4 Payments

The following text describes current solution for Open Banking PSD2 Payment Initiation methods.

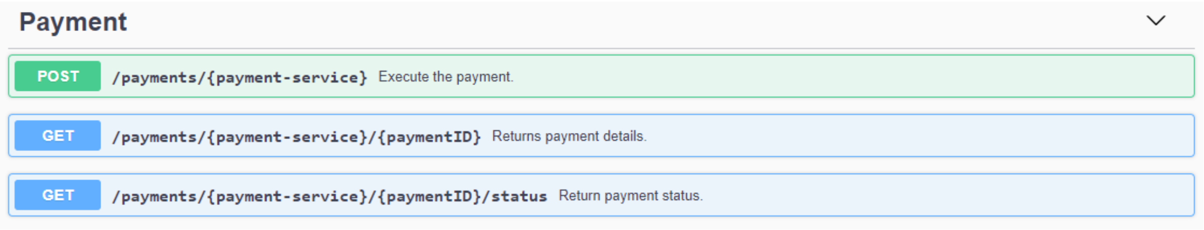
You will be able to initiate payments from Bank accounts through the Payment Initiation API's, in the form of a credit transfer from an account of the PSU to the credit account.

The transaction at the XS2A interface is initiated by the PSU at the PSU-TPP interface.

The ASPSP will reject the transaction if the TPP cannot be identified correctly at the XS2A interface and/or if it does not have the role PISP.

The Payment Initiation API consists of the flows and payloads required for a general payment initiation. All payment transactions will be subject to SCA.

You will find the following structure into the APS Bank Swagger UI:



Payment

POST /payments/{payment-service} Execute the payment.

GET /payments/{payment-service}/{paymentID} Returns payment details.

GET /payments/{payment-service}/{paymentID}/status Return payment status.

POST/payments/{payment-service}

Executes the payment. Stores the requested payment. Payment needs to pass Strong Customer Authentication flow over the digital banking channels.

GET/payments/{payment-service}/{paymentID}

Returns payment details. Only works for the submitted payment over this API.

GET/payments/{payment-service}/{paymentID}/status

Returns payment status. Only works for the submitted payment over this API.

The following table describes the items used for each method:

Name	POST/ payments/ {payment-service}	GET/payments/ {payment-service}/ {paymentID}	GET/payments/ {payment-service}/ {paymentID}/status
payment-service * string (path)	x	x	x
request ** (body)	x		
paymentID * string (path)		x	x

Name	POST/ payments/ {payment-service}	GET/payments/ {payment-service}/ {paymentID}	GET/payments/ {payment-service}/ {paymentID}/status
APP-ID * string (header)	X	X	X
TPP-Redirect-URI * string (header)	X		
TPP-Nok-Redirect-URI * string (header)	X		
X-Request-ID * string (header)	X	X	X
PSU-IsCorporate * boolean (header)	X	X	X
PSU-IP-Address * string (header)	X	X	X
PSU-IP-Port string (header)	X	X	X

** Mandatory Parameter*

*** Mandatory Parameter. Request structure explained below explained in section 4.6*

4.5 Description of the parameters

The following table provides a description of the parameters used on our API Calls:

Request Parameter	Description
account-ID * string (path)	Unique ID of the account resource. This identification is denoting the addressed account. The account-id is the "resourceId" attribute of the account structure. Its value is constant at least throughout the lifecycle of a given consent.
Authorization String (header)	Bearer Token.
bookingStatus string (query)	Permitted codes are "booked", “pending” and "both".
consent-id * string (path)	Identification of the corresponding consent as granted by the PSU.
dateFrom string (query)	Starting date (inclusive the date dateFrom) of the transaction list.

Request Parameter	Description
dateTo **string** **(query)**	End date (inclusive the data dateTo) of the transaction list, default is “now” if not given.
entryReferenceFrom **string** **(query)**	This data attribute is indicating that the AISP is in favour to get all transactions after the transaction with identification entryReferenceFrom alternatively to the period defined above. Not supported.
paymentID * **string** **(path)**	Resource identification of the related payment initiation
payment-service * **string** **(path)**	Available values : sepa-credit-transfers, cross-border-credit-transfers, internal-transfer
PSU-Accept **string** **(header)**	The forwarded IP Accept header fields consist of the corresponding HTTP request Accept header fields between PSU and TPP, if available. Not in use
PSU-Accept-Charset **string** **(header)**	The forwarded IP Accept header fields consist of the corresponding HTTP request Accept header fields between PSU and TPP, if available. Not in use
PSU-Accept-Encoding **string** **(header)**	The forwarded IP Accept header fields consist of the corresponding HTTP request Accept header fields between PSU and TPP, if available. Not in use
PSU-Accept-Language **string** **(header)**	The forwarded IP Accept header fields consist of the corresponding HTTP request Accept header fields between PSU and TPP, if available. Not in use
PSU-Device-ID **string** **(header)**	UUID (Universally Unique Identifier) for a device, which is used by the PSU if available. UUID identifies either a device or a device dependent application installation. In case of an installation identification this ID needs to be unaltered until removal from device. Not in use.
PSU-Geo-Location **string** **(header)**	The forwarded Geo Location of the corresponding HTTP request between PSU and TPP if available. Not in use.
PSU-Http-Method **string** **(header)**	HTTP method used at the PSU – TPP interface, if available. Not in use.
PSU-IP-Address * **string** **(header)**	The forwarded IP Address header field consists of the corresponding HTTP request IP Address field between PSU and TPP. It shall be contained if and only if this request was actively initiated by the PSU.
PSU-IP-Port **string** **(header)**	The forwarded IP Port header field consists of the corresponding HTTP request IP Port field between PSU and TPP, if available.
request * **(body)**	Request body in JSON format always.
TPP-Nok-Redirect-URI * **string** **(header)**	URI of the TPP, where the transaction flow shall be redirected to after a Redirect in case authorisation was not successful. It is mandate to send it (including OAuth2 SCA approach), specifically when TPP-Redirect-Preferred equals 'true'. It is recommended to use this header field always.

Request Parameter	Description
TPP-Redirect-URI * **string** **(header)**	URI of the TPP, where the transaction flow shall be redirected to after a Redirect. It is mandate to send it (including OAuth2 SCA approach), specifically when TPP-Redirect-Preferred equals 'true'. It is recommended to use this header field always.
transaction-ID * **string** **(path)**	ID of the transaction as determined by the initiating party, unique to the call.
withBalance **boolean** **(query)**	If contained, this function reads the list of transactions including the booking balance.
X-Request-ID * **string** **(header)**	ID of the request, unique to the call, as determined by the initiating party. This is unique identifier traced through system as CorrelationID

Additionally to the above request parameters, APS has defined the below detailed parameters which allow TPPs to connect use their TPP Development account details and identify themselves and the application used, when using the APIs endpoint.

Request Parameter	Description
APP-ID * **string** **(header)**	Identification code of the Application created for the TPP. It is generated when a new Application is generated into the TPP Development Portal.
PSU-IsCorporate * **boolean** **(header)**	Identification of the type of account that the PSU is using. True if PSU is Corporate, otherwise is Retail.

4.6 Request Structure

4.6.1 Request Url

The following Request Urls are used for the API Endpoints mentioned above:

CONSENT

- POST/consents

https://sbobapi.apsbank.com.mt:4432/consents/

- GET/consents/{consent-id}
- DELETE/consents/{consent-id}

*https://sbobapi.apsbank.com.mt:4432/consents/*consent-id**

- GET/consents/{consent-id}/status

  *https://sbobapi.apsbank.com.mt:4432/consents/*consent-id*/status*

Note:

*consent-id * is the ID of the existing consent

ACCOUNTS

- GET/accounts

  https://sbobapi.apsbank.com.mt:4432/accounts

- GET/accounts/{account-ID}

  *https://sbobapi.apsbank.com.mt:4432/accounts/*account-ID**

- GET/accounts/{account-ID}/balances

  *https://sbobapi.apsbank.com.mt:4432/accounts/*account-ID*/balances*

- GET/accounts/{account-ID}/transactions

  *https://sbobapi.apsbank.com.mt:4432/accounts/*account-ID*/transactions*

- GET/accounts/{account-ID}/transactions/{transaction-ID}

  *https://sbobapi.apsbank.com.mt:4432/accounts/*account-ID*/transactions/*transaction-ID**

Note:

account-ID is the Unique ID of the account resource

transaction-ID stands for the ID of the referenced transaction

CONFIRMATION OF FUNDS

- POST/ funds-confirmations

  https://sbobapi.apsbank.com.mt:4432/funds-confirmations

PAYMENTS

- POST/payments/{payment-service}

  *https://sbobapi.apsbank.com.mt:4432/payments/*payment-service**

- GET/payments/{payment-service}/{paymentID}

  *https://sbobapi.apsbank.com.mt:4432/payments/*payment-service*/*paymentID**

- GET/payments/{payment-service}/{paymentID}/status

  *https://sbobapi.apsbank.com.mt:4432/payments/*payment-service*/*paymentID*/status*

Note:

payment-service being SEPA, cross-border or internal transfers

**payment-ID* stands for the ID of the referenced payment*

4.6.2 Request (body)

Request body in JSON format always.

CONSENT

The Schema for Consent request:

Parameter	Description
recurringIndicator (boolean)	true, if the consent is for recurring access to the account data; false, if the consent is for one access to the account data **default: false**
validUntil **(string)**	This parameter is requesting a valid until date for the requested consent. The content is the local ASPSP date in ISODate Format, e.g. 2017-10-30. (Max. 3 months)
frequencyPerDay **(integer)**	This field indicates the requested maximum frequency for an access without PSU involvement per day. For a one-off access, this attribute is set to "1". **default: 0**
combinedServiceIndicator **(boolean)**	The Combinedserviceindicator Schema. Always set to false. **default: false**

Example Value Model:

```
{
  "recurringIndicator": false,
  "validUntil": "2019-12-10",
  "frequencyPerDay": 3,
  "combinedServiceIndicator": false
}
```

FUNDS CONFIRMATION

The Schema for Funds Confirmation request:

Parameter		Description
iban **(string)**		Iban number of PSU Account.
instructedAmount	**currency** **(string)**	The Currency of the instructed amount
	amount **(string)**	The Amount of the instructed amount

Example Value Model:

```
{
  "account": {
  "iban": " MT51XXXXXXXXXXXXXXXXXXXXXXXXXXX19"
 },
 "instructedAmount": {
  "currency": "EUR",
  "amount": "123"
 }
}
```

PAYMENTS

Schema for Payment request:

Parameter		Description
debtorAccount	**iban** **(string)**	Iban number of PSU Account.
instructedAmount	**Currency** **(string)**	The Currency of the instructed amount
	amount **(string)**	The Amount of the instructed amount
creditorAccount	**iban** **(string)**	Iban number of the creditor Account.
	Currency **(string)**	Currency of the creditor Account.

Parameter		Description
creditorAgent **(BICFI)**		Creditor Bank swift code
creditorSortCode **(Int32)**		Creditor Bank sort code
creditorAgentName **(Max70Text)**		Creditor Bank name
creditorAddress	**Address** **(Max70Text)**	Creditor Bank address details
	Street **(Max70Text)**	
	Country **(ISO 3166)**	
intermediaryAgent **(BICFI)**		Intermediary Bank Swift code
creditorName **(Max70Text)**		Beneficiary name
creditorAgentAddress	**buildingNumber** **(String)**	Beneficiary address details
	Street **(Max70Text)**	
	City **(string)**	
	postalCode **(String)**	
	Country **(ISO 3166)**	
purposeCode **(Purpose Code)**		ISO 20022 4-char purpose code
remittanceInformationUnstructured **(Max140Text)**		Transaction description. Alpha/Numerical only as per SWIFT

Example value:

```
{
"debtorAccount":
            {"iban": "MT85APSB77013000000030053010013"},
"instructedAmount":
```

```
            {"currency": "EUR",
            "amount": "250.00"},
"creditorAccount":
            {"iban": "MT24MMEB44860000000086000031007",
             "currency": "EUR"},
"creditorAgent": "MMEBMTMT",
"creditorSortCode": "123456",
"creditorAgentName": "HSBC BANK",
"creditorAgentAddress": {
   "Address":"Address cred Ag",
   "street": "Street credit",
   "country": "MT"},
"intermediaryAgent": "IntermediaryAg",
"creditorName": "TPP2" ,
"creditorAddress": {
   "buildingNumber": "Address line 1",
   "street": "Address line 2",
   "city": "SLIEMA",
   "postalCode": "SL1414",
   "country": "MT"},
"purposeCode": "123",
"remittanceInformationUnstructured": "PaymentTest4",
}
```

References / Pages of interest

Berlin Group NextGenPSD2 Downloads Documentation: Most recent version of the NextGenPSD2 Access to Account Framework documents.

- Joint Initiative on a PSD2 Compliant XS2A Interface, NextGenPSD2 XS2A Framework, Implementation Guidelines, version 1.3.4, published 5 July 2019.

NISP: The NextGenPSD2 Implementation Support Program.

European Commission

- Trust Services and Electronic identification (eID)
- Trusted List Browser

eIDAS Regulation - 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

RTS - 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

PSD2 Directive - 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