



Open Banking interface

Technical Registration services

Version 2.0

Please visit openbanking.budapestbank.hu to get more information on Budapest Bank OpenBanking, including operational information on the live service.

Contact

In case you have a question or any problem regarding our services, please send us a message using the following email address and we will reply in 2 working days:

openbanking@budapestbank.hu

- 1 Introduction..... 3
- 2 Supported Operations 3
- 3 Operation Definitions 4
 - 3.1 getRegistrationStatus Operation..... 4
 - 3.1.1 Description 4
 - 3.1.2 Input Fields 4
 - 3.1.3 Output Fields 4
 - 3.2 initiateTechnicalRegistration Operation 10
 - 3.2.1 Description 10
 - 3.2.2 Input Fields 11
 - 3.2.3 Output Fields 12

1 Introduction

This document describes the TechnicalRegistration_V2 service provided by Budapest Bank Open Banking interface. The operations described in this document are required for all TPPs. These operations are used to register the TPP in the Budapest Bank PSD2 API. All other services are available only if the TPP has a valid registration.

This service is used to renew the registration or replace the existing registered public keys for the TPP, and query the status of existing registration.

Timezone handling in the API for all date and datetime fields:

- In message header all fields are treated in UTC if timezone not present.
- In message body (business data) if timezone not present values are treated as Budapest, Hungary actual timezone (CET or CDT depending on the time of the year)
- In the response data TPPs should treat all date and datetime fields in the same manner

Fields presented in this document should be treated as described here. This document may impose further restrictions over the WSDL. Where there is a conflict between the WSDL definition and this document, this document overrules all definitions in the WSDL. In requests fields not present in this document may be ignored by the API. In responses if there is a field not described in this document those should be treated as informational values. TPP should not rely on those fields for business logic.

2 Supported Operations

Operation Name	Description	Role of the TPP	SCA Approval required by PSU(s)	Consent token required
initiateTechnicalRegistration	<p>TPP's first interaction with the Bank. Must share basic information, including the role of TPP and the public key used to TLS and sign messages.</p> <p>This API has to be called periodically aligned with TPP's certificates renewal.</p> <p>In case of successful technical registration, the Bank grants access to the defined services (see in roles column).</p>	AISP, PISP, PIISP	no	no
getRegistrationStatus	Inquiry status of the registration.	AISP, PISP, PIISP	no	no

All operations are available via SOAP over HTTPS interface.

3 Operation Definitions

This section provides information about the TechnicalRegistration_V2 operations. The base definition of the service can be found in the TechnicalRegistration_V2.wsdl file. This document provides more details on input and output fields. Requirements or restrictions that cannot be put into the WSDL can be found here.

All business date/datetime fields are Budapest local time CET or CDT if timezone is not present. All header fields are UTC timezone if not marked otherwise.

The c: prefix in the documentation refers to the bbCommon namespace in the wsdl definitions.

3.1 getRegistrationStatus Operation

3.1.1 Description

The purpose of this operation is to query the registration status of the technical registration of the TPP.

3.1.2 Input Fields

None. The request body should not contain any fields. Message level signature will be used for the identification.

3.1.3 Output Fields

getRegistrationStatus Operation - Output Fields				
#	Name	Type	Num	Description
1	registration	complex type	[1..1]	Transaction id of the revoke request.
1.1	partnerName	c:partnerNameType	[1..1]	Name of the TPP String, maximum length is 100 characters.
1.2	partnerOrganizationIdentifier	c:partnerOrganizationIdentifierType	[1..1]	Organization Identifier of the TPP. String, maximum length is 64 characters.
1.3	apps	complex type	[0..1]	App definitions for the TPP
1.3.1	app	complex type	[1..*]	Description for the TPP app
1.3.1.1	name	c:appNameType	[1..1]	Name of the applications where the interface is used. Maximum 100 characters
1.3.1.2	id	c:appldType	[1..1]	Id of the application where the interface is used. Maximum 36 characters

getRegistrationStatus Operation - Output Fields				
#	Name	Type	Num	Description
1.4	description	c:descriptionType	[0..1]	Other information. String, maximum length is 200 characters.
1.5	contactEmail	c:contactEmailType	[1..1]	Contact email address. . String, maximum length is 50 characters.
1.6	requestedRole	c:roleType	[1..1]	AISP (account information), PISP (payment initiation), PIISP (issuing of card-based payment instruments)
1.7	publicKey	xs:string	[1..1]	ETSI TS 119 495 certificate of the TPP. This is used for TLS mutual authentication and message signing on TPP's side. Given authorization number and role have to be match with the certificate content.
1.8	partnerAddress	c:partnerAddressType	[0..1]	Official address. String, maximum length is 200 characters.
1.9	partnerURL	c:partnerURLType	[0..1]	Official web address. String, maximum length is 200 characters.
1.10	expirationDate	xs:dateTime	[0..1]	Expiration date of the registration, calculated by the given publicKey's expiration date.
1.11	status	c:registrationStatusType	[1..1]	Status of the registration request (pending, accepted, declined).

3.1.4 Sample XML request

Below you can find a sample XML request:

```
<soapenv:Envelope xmlns:bbrt="http://bbrt.hu/" xmlns:ns="http://bbrt.hu/openApiServices/TechnicalRegistration/1/"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
```

```
<soapenv:Header>
```

```
<wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
```



```
<ds:KeyInfo Id="AA-*****">
  <wsse:SecurityTokenReference wsu:Id="ABC-*****">
    <wsse:Reference URI="#A*_*****" ValueType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
x509-token-profile-1.0#X509v3"/>
  </wsse:SecurityTokenReference>
</ds:KeyInfo>
</ds:Signature>
</wsse:Security>
<bbrt:Head>
  <RequestDate>2019-09-12</RequestDate>
  <RequestTime>08:12:55</RequestTime>
  <RequestId>5b0e5096-a578-4586-bc4b-6288974069bf</RequestId>
</bbrt:Head>
</soapenv:Header>
<soapenv:Body wsu:Id="id-*****" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
utility-1.0.xsd">
  <ns:getRegistrationStatus/>
</soapenv:Body>
</soapenv:Envelope>
```

3.2 initiateTechnicalRegistration Operation

3.2.1 Description

TPP's first interaction with the Bank. Must share basic information, including the role of TPP and the public key used to TLS and sign messages.

This API has to be called periodically aligned with TPP's certificate renewal.

Accessible to all TPPs.

3.2.2 Input Fields

initiateTechnicalRegistration Operation - Input Fields				
#	Name	Type	Num	Description
1	partnerName	c:partnerNameType	[1..1]	Name of the TPP String, maximum length is 100 characters.
2	partnerOrganizationIdentifier	c:partnerOrganizationIdentifierType	[1..1]	Organization Identifier of the TPP. String, maximum length is 50 characters.
3	apps	complex type	[0..1]	App definitions for the TPP
3.1	app	complex type	[1..*]	Description for the TPP app
3.1.1	name	c:appNameType	[1..1]	Name of the applications where the interface is used. Maximum 100 characters
3.1.2	id	c:appIdType	[1..1]	Id of the application where the interface is used. Maximum 36 characters
4	description	c:descriptionType	[0..1]	Other information. String, maximum length is 200 characters.
5	contactEmail	c:contactEmailType	[1..1]	Contact email address. . String, maximum length is 50 characters.
6	requestedRole	c:roleType	[1..1]	AISP (account information), PISP (payment initiation), PIISP (issuing of card-based payment instruments)
7	publicKey	xs:string	[1..1]	ETSI TS 119 495 certificate of the TPP. This is used for TLS mutual authentication and message signing on TPP's side. Given authorization number and role have to be match with the certificate content.
8	partnerAddress	c:partnerAddressType	[0..1]	Official address. String, maximum length is 200 characters.
9	partnerURL	c:partnerURLType	[0..1]	Official web address. String, maximum length is 200 characters.

3.2.3 Output Fields

initiateTechnicalRegistration Operation - Output Fields				
#	Name	Type	Num	Description
1	transactionId	c:transactionIdType	[1..1]	The transaction ID of the registration request.
2	registration	complex type	[1..1]	Transaction id of the revoke request.
2.1	partnerName	c:partnerNameType	[1..1]	Name of the TPP String, maximum length is 100 characters.
2.2	partnerAuthorizationNumber	c:partnerAuthorizationNumberType	[1..1]	Authorization Number of the TPP. String, maximum length is 10 characters.
2.3	apps	complex type	[0..1]	
2.3.1	app	complex type	[1..*]	Transaction history element
2.3.1.1	appName	c:appNameType	[1..1]	Name of the applications where the interface is used. Maximum 100 characters
2.3.1.2	id	c:appIdType	[1..1]	Id of the application where the interface is used. Maximum 36 characters
2.4	description	c:descriptionType	[0..1]	Other information. String, maximum length is 200 characters.
2.5	contactEmail	c:contactEmailType	[1..1]	Contact email address. . String, maximum length is 50 characters.
2.6	requestedRole	c:roleType	[1..1]	AISP (account information), PISP (payment initiation), PIISP (issuing of card-based payment instruments)
2.7	publicKey	xs:string	[1..1]	ETSI TS 119 495 certificate of the TPP. This is used for message signing on TPP's side. Given authorization number and role have to be match with the certificate content.
2.8	partnerAddress	c:partnerAddressType	[0..1]	Official address. String, maximum length is 200 characters.
2.9	partnerURL	c:partnerURLType	[0..1]	Official web address. String, maximum length is 200 characters.
2.10	expirationDate	xs:dateTime	[0..1]	Expiration date of the registration, calculated by the given publicKey's expiration date.
2.11	status	c:registrationStatusType	[1..1]	Status of the registration request (pending, accepted, declined).

