PISP API Specification

1 Revision history

Version	Description	Modified By	Date
1.0	Initial version	M. Richards	4 November 2020
1.1	Following review by Henrik Karlsson	M. Richards	9 November 2020
1.2	Further changes consequent on HK review	M. Richards	13 November 2020
1.3	Following HK and LD reviews	M. Richards	9 February 2021
1.4	Following reviews	M. Richards	30 March 2021
1.5	Following discussion on transaction object	M. Richards	8 June 2021
1.6	Revisions to integrate with Imali interface	M. Richards	4 March 2022
1.7	Revision to include Account lookup with consent	P. Baker	4 July 2023

2 References

The following references are used in this specification:

Reference	Description	Version	URL
Ref. 1	Open API for FSP Interoperability	1.1	https://github.com/mojaloop/mojaloop-specification/blob/master/documents/v1.1-document-set/API%20Definition_v1.1.pdf

3 PISP API

This section describes the content of the API which will be used by PISPs.

The content of the API falls into two sections. The first section manages the process for linking customer accounts and providing a general permission mechanism for PISPs to perform operations on those accounts. The second section manages the transfer of funds at the instigation of a PISP.

The content of the account linking section consists of the following operations:

- The PISP requests a list of accounts for a customer from the customer's account provider. The account provider satisfies itself that association really was requested by their customer through an authentication request through the PISP. The account owner then provides the customer's account information to the PISP.
- The PISP then requests association with a customer account on behalf of the customer.
- The owner of the customer's account satisfies itself that association really was requested by their customer, and the customer has a chance to confirm or modify directly with the account owning DFSP the types of access and the accounts for which the PISP will have permission. The DFSP then notifies the PISP that the customer has authorised access, and provides a token which the PISP can use to continue the process. This part of the process is performed via direct communication between the PISP application and the DFSP, and does not use the API.
- The DFSP requests confirmation from the PISP that the DFSP's customer has confirmed with the PISP that they authorise the PISP to perform operations on their account. Confirmation requires the PISP to provide the bearer token that the DFSP sent the PISP as confirmation of the successful completion of the out-of-band customer authorization described in the previous step.
- The DFSP confirms to the PISP the accounts which it will allow the PISP to access and the access types available to the PISP on each account. It also confirms the following items of information:
    - For each account to which the DFSP grants access, the Mojaloop identifier which the PISP should use in subsequent access requests to identify the account on which the operation should be performed.

- For each association to be made, the PISP asks the user's handset to register a keypair to be used to confirm transfer requests in the future. The public key belonging to this keypair is returned to the DFSP, together with the account identifier provided by the DFSP.
- If the DFSP is not using a local authentication service to verify the challenges it uses to authenticate transfer requests, it asks the scheme's authentication service to register the public key and associate it with the account ID it provides.
- For each association to be made, the DFSP provides a challenge to the PISP. The PISP asks the customer to sign the challenge, and returns the signed challenge to the DFSP.
- The DFSP verifies that the signed challenge matches the value that it holds for the association, using either its own authentication service or the scheme authentication service.

The API is used by the following different types of participant, as follows:

1. PISPs.
2. DFSPs who offer services to their customers which allow the customer to access their account via one or more PISPs.
3. FIDO authorization servers.
4. The Mojaloop switch

Each resource in the API definition is accompanied by a definition of the type(s) of participant allowed to access it.

3.1 Resources

The PISP API will contain the following resources:

3.1.1 tppAccountsRequest

The **/tppAccountsRequest** resource is used to request consent from a user for access to their accounts information. This resource must be called before the /tppAccounts resource can be queried which provides the account information.

3.1.1.1 POST /tppAccountsRequest

Used by: PISP

The **POST /tppAccountsRequest** is used to request access from the user to distribute account information from the user DFSP to the PISP.

Callback and data model for **POST /tppAccountsRequest**:

1. Callback: **PUT /tppAccountsRequest/**<ID>
2. Error callback: **PUT /tppAccountsRequest/**<ID>**/error**
3. Data model – see below

Name	Cardinality	Type	Description
accountRequestId	1	CorrelationId	Common ID between the PISP and the Payer DFSP for the account consent request object. The ID should be reused for resends of the same consent request. A new ID should be generated for each new account consent request.
partyIdInfo	1	PartyIdInfo	The identifier of the customer on behalf of whom the consent request is being made.
authChannels	1..n	ConsentRequestChannelType	A collection of the types of authentication that the DFSP may use to verify that its customer has in fact requested access for the PISP to the accounts requested..

Name	Cardinality	Type	Description
callbackUri	0..1	Uri	The callback URI that the user will be redirected to after completing verification via the WEB authorization channel

3.1.1.1.1 PUT /tppAccountsRequest/<ID>

Used by: DFSP

When a PISP requests a series of permissions from a DFSP on behalf of a DFSP's customer, not all the permissions requested may be granted by the DFSP. Conversely, the out-of-loop authorization process may result in additional privileges being granted by the account holder to the PISP. The **PUT /tppAccountsRequest/**<ID> resource returns the current state of the permissions relating to a particular authorization request. The data model for this call is as follows:

Name	Cardinality	Type	Description
authChannels	1	ConsentRequestChannelType	The authorization channel chosen by the DFSP from the list of supported authorization channels proposed by the PISP in the original request.
callbackUri	0..1	Uri	The callback URI that the user will be redirected to after completing verification via the WEB authorization channel
authUri	0..1	Uri	The URI that the PISP should call to complete the linking procedure if completion is required.
authToken	0..1	BinaryString	The bearer token given to the PISP by the DFSP as part of the out-of-loop authentication process

3.1.1.2 Error callbacks

This section describes the error callbacks that are used by the server under the resource **/tppAccountsRequest**.

3.1.1.2.1 PUT /tppAccountsRequest/<ID>/error

Used by: DFSP

If the server is unable to complete the accounts consent request, or if an out-of-loop processing error or another processing error occurs, the error callback **PUT /tppAccountsRequest/**<ID>**/error** is used. The <ID> in the URI should contain the <ID> that was used in the **POST /tppAccountsRequest** request. The data model for this resource is as follows:

Name	Cardinality	Type	Description
errorInformation	1	ErrorInformation	Error code, category description.

3.1.2 tppAccounts

The **/tppAccounts** resource is used to request information from a DFSP relating to the accounts it holds following from the /tppAccountsRequest API flow that is used to obtain permission for this request. The Correlation Id (accountRequestId) provided refers to the original authentication request, and the signed challenge that must be

validated to complete the authentication. The PartyInfomation for this request will need to be looked up based on the account request Id provided.

The **PUT /tppAccounts** provides the account information to the PISP who can then display the names of the accounts to the user, and allow the user to select the accounts with which they wish to link via the PISP.

The **/tppAccounts** resource supports the endpoints described below.

3.1.3 Requests

This section describes the services that a PISP can request on the **/tppAccounts** resource.

3.1.3.1.1 GET /tppAccounts/<accountRequestId>/<SignedChallenge>

Used by: PISP

The HTTP request **GET /tppAccounts/<**accountRequestId**>/***<SignedChallenge>* is used to lookup information about the requested Party's accounts, defined by <accountRequestId> and *<ID>* (for example, **GET /tppAccounts/12345/56789**).

Callback and data model information for **GET /tppAccounts/***<accountRequestId>***/***<SignedChallenge>*:

- Callback - **PUT /tppAccounts/***<accountRequestId>*
- Error Callback - **PUT /tppAccounts/***<accountRequestId>***/error**
- Data Model – Empty body

3.1.4 Callbacks

The responses for the **/tppAccounts** resource are as follows

3.1.4.1.1 PUT /tppAccounts/<accountRequestId>

Used by: DFSP

The **PUT /tppAccounts/**<accountRequestId> response is used to inform the requester of the result of a request for accounts information. The identifier type and the identifier ID given in the call are the values given in the original request (see Section 3.1.1.1.1 above.)

The data content of the message is given below.

Name	Cardinality	Type	Description
accountList	1	AccountList	Information about the accounts that the DFSP associates with the identifier sent by the PISP.

3.1.4.1.2 PUT /tppAccounts/<accountRequestId>/error

Used by: DFSP

The **PUT /tppAccounts/**<accountRequestId**>/error** response is used to inform the requester that an account list request has given rise to an error. The identifier type and the identifier ID given in the call are the values given in the original request (see Section 3.1.1.1.1 above.)

The data content of the message is given below.

Name	Cardinality	Type	Description
errorInformation	1	ErrorInformation	The result of the authentication check carried out by the authentication service.

3.1.5 tppConsentRequests

The **/tppConsentRequests** resource is used by a PISP to initiate the process of linking with a DFSP's account on behalf of a user. The PISP contacts the DFSP and sends a list of the permissions that it wants to obtain and the accounts for which it wants permission.

3.1.5.1 Requests

This section describes the services that can be requested by a client on the API resource **/tppConsentRequests**.

3.1.5.1.1 GET /tppConsentRequests/<ID>

Used by: PISP

The HTTP request **GET /tppConsentRequests/**<ID> is used to get information about a previously requested consent. The <ID> in the URI should contain the **requestId** that was assigned to the request by the PISP when the PISP originated the request.

Callback and data model information for **GET /tppConsentRequests/***<ID>*:

- Callback – **PUT /tppConsentRequests /***<ID>*
- Error Callback – **PUT /tppConsentRequests /***<ID>***/error**
- Data Model – Empty body

3.1.5.1.2 POST /tppConsentRequests

Used by: PISP

The HTTP request **POST /tppConsentRequests** is used to request a DFSP to grant access to one or more accounts owned by a customer of the DFSP for the PISP who sends the request.

Callback and data model for **POST /tppConsentRequests**:

- Callback: **PUT /tppConsentRequests/**<ID>
- Error callback: **PUT /tppConsentRequests/**<ID>**/error**
- Data model – see below

Name	Cardinality	Type	Description
consentRequestId	1	CorrelationId	Common ID between the PISP and the Payer DFSP for the consent request object. The ID should be reused for resends of the same consent request. A new ID should be generated for each new consent request.
partyIdInfo	1	PartyIdInfo	The identifier of the customer on behalf of whom the consent request is being made.
scopes	1..n	Scope	One or more requests for access to a particular account. In each case, the address of the account and the types of access required are given.
authChannels	1..n	ConsentRequestChannelType	A collection of the types of authentication that the DFSP may use to verify that its customer has in fact requested access for the PISP to the accounts requested..
callbackUri	0..1	Uri	The callback URI that the user will be redirected to after completing verification via the WEB authorization channel

3.1.5.2 Callbacks

This section describes the callbacks that are used by the server under the resource **/tppConsentRequests**.

3.1.5.2.1 PATCH /tppConsentRequests/<ID>

Used by: DFSP, PISP

When a party intends to change the content of a consent request, it can do this via the **PATCH /tppConsentRequests/**<ID> resource. The syntax of this call complies with the JSON Merge Patch specification[1] rather than the JSON Patch specification[2]: that is to say, . The **PATCH /tppConsentRequests/**<ID> resource contains a set of proposed changes to the current state of the permissions relating to a particular authorization request. The data model for this call is as follows:

Name	Cardinality	Type	Description
scopes	0..n	Scope	One or more requests for access to a particular account. In each case, the address of the account and the types of access required are given.
authChannels	0..1	ConsentRequestChannelType	The authorization channel chosen by the DFSP from the list of supported authorization channels proposed by the PISP in the original request.
callbackUri	0..1	Uri	The callback URI that the user will be redirected to after completing verification via the WEB authorization channel
authUri	0..1	Uri	The URI that the PISP should call to complete the linking procedure if completion is required.
authToken	0..1	BinaryString	The bearer token given to the PISP by the DFSP as part of the out-of-loop authentication process

3.1.5.2.2 PUT /tppConsentRequests/<ID>

Used by: DFSP

When a PISP requests a series of permissions from a DFSP on behalf of a DFSP's customer, not all the permissions requested may be granted by the DFSP. Conversely, the out-of-loop authorization process may result in additional privileges being granted by the account holder to the PISP. The **PUT /tppConsentRequests/**<ID> resource returns the current state of the permissions relating to a particular authorization request. The data model for this call is as follows:

Name	Cardinality	Type	Description
Scopes	1..n	Scope	One or more requests for access to a particular account. In each case, the address of the account and the types of access required are given.
authChannels	1	ConsentRequestChannelType	The authorization channel chosen by the DFSP from the list of supported authorization channels proposed by the PISP in the original request.
callbackUri	0..1	Uri	The callback URI that the user will be redirected to after completing verification via the WEB authorization channel

[1] As defined in https://tools.ietf.org/html/rfc7386
[2] As defined in https://tools.ietf.org/html/rfc6902

Name	Cardinality	Type	Description
authUri	0..1	Uri	The URI that the PISP should call to complete the linking procedure if completion is required.
authToken	0..1	BinaryString	The bearer token given to the PISP by the DFSP as part of the out-of-loop authentication process

3.1.5.3 Error callbacks

This section describes the error callbacks that are used by the server under the resource **/tppConsentRequests**.

3.1.5.3.1 **PUT /tppConsentRequests/**<ID>**/error**

Used by: DFSP

If the server is unable to complete the consent request, or if an out-of-loop processing error or another processing error occurs, the error callback **PUT /tppConsentRequests/**<ID>**/error** is used. The <ID> in the URI should contain the <ID> that was used in the **GET /tppConsentRequests/**<ID> request or the **POST /tppConsentRequests** request. The data model for this resource is as follows:

Name	Cardinality	Type	Description
errorInformation	1	ErrorInformation	Error code, category description.

3.1.6 tppConsents

The **/tppConsents** resource is used to negotiate a series of permissions between the PISP and the DFSP which owns the account(s) on behalf of which the PISP wants to transact.

The **/tppConsents** request is originally sent to the PISP by the DFSP following the original consent request process described in Section 3.1.3 above. At the close of this process, the DFSP which owns the customer's account(s) will have satisfied itself that its customer really has requested that the PISP be allowed access to their accounts, and will have defined the accounts in question and the type of access which is to be granted.

3.1.6.1 Requests

The **/tppConsents** resource will support the following requests.

3.1.6.1.1 **GET /tppConsents/**<ID>

Used by: DFSP

The **GET tppConsents/**<ID> resource allows a party to enquire after the status of a consent. The <ID> used in the URI of the request should be the consent request ID which was used to identify the consent when it was created.

Callback and data model information for **GET /tppConsents/**<ID>:

- Callback – **PUT /tppConsents/**<ID>
- Error Callback – **PUT /tppConsents/**<ID>**/error**
- Data Model – Empty body

3.1.6.1.2 **POST /tppConsents**

Used by: DFSP

The **POST /tppConsents** request is used to request the creation of a consent for interactions between a PISP and the DFSP who owns the account which a PISP's customer wants to allow the PISP access to.

Callback and data model information for **POST /tppConsents/**<ID>:

- Callback – **PUT /tppConsents/**<ID>
- Error Callback – **PUT /tppConsents/**<ID>**/error**

Data Model – defined below

Name	Cardinality	Type	Description
consentId	1	CorrelationId	Common ID between the PISP and the Payer DFSP for the consent object. The ID should be reused for resends of the same consent. A new ID should be generated for each new consent.
consentRequestId	1	CorrelationId	The ID given to the original consent request on which this consent is based.
initiatorId	1	FspId	The ID of the PISP associated with the consent.
issuerId	1	FspId	The ID of the DFSP which created the consent.
scopes	1..n	Scope	One or more accounts on which the DFSP is prepared to grant specified permissions to the PISP.
credential	0..1	Credential	The credential which is being used to support the consents.
extensionList	0..1	ExtensionList	Optional extension, specific to deployment

3.1.6.1.3 DELETE /tppConsents/<ID>

The **DELETE /tppConsents/**<ID> request is used to request the deletion of a previously agreed consent. The switch should be sure not to delete the consent physically; instead, information relating to the consent should be marked as deleted and requests relating to the consent should not be honoured.

Note: the ALS should verify that the participant who is requesting the deletion is either the initiator named in the consent or the account holding institution named in the consent. If any other party attempts to delete a consent, the request should be rejected, and an error raised.

Callback and data model information for **DELETE /tppConsents/**<ID>:

- Callback – **PUT /tppConsents/**<ID>
- Error Callback – **PUT /tppConsents/**<ID>**/error**

3.1.6.2 Callbacks

The **/tppConsents** resource will support the following callbacks:

3.1.6.2.1 PATCH/tppConsents/<ID>

Used by: PISP

When a party intends to change the content of a consent, it can do this via the **PATCH /tppConsents/**<ID> resource. The syntax of this call complies with the JSON Merge Patch specification[3] rather than the JSON Patch specification[4]. The **PATCH /tppConsents/**<ID> resource contains a set of proposed changes to the current state of the permissions relating to a particular authorization grant. The data model for this call is as follows:

Name	Cardinality	Type	Description
scopes	0..n	Scope	The scopes covered by the consent.
consentState	0	CredentialState	State of the consent
credential	0		
extensionList	0..1	ExtensionList	Optional extension, specific to deployment

[3] As defined in https://tools.ietf.org/html/rfc7386
[4] As defined in https://tools.ietf.org/html/rfc6902

3.1.6.2.2 PUT /tppConsents/<ID>

Used by: PISP

The **PUT /tppConsents/**<ID> resource is used to return information relating to the *consent* object whose *consentId* is given in the URI. The data returned by the call is as follows:

Name	Cardinality	Type	Description
scopes	1..n	Scope	The scopes covered by the consent.
consentState	1	ConsentState	State of the consent
credential	1	Credential	Information about the challenge which relates to the consent.
extensionList	0..1	ExtensionList	Optional extension, specific to deployment

3.1.6.3 Error callbacks

This section describes the error callbacks that are used by the server under the resource **/tppConsents**.

3.1.6.3.1 PUT /tppConsents/<ID>/error

Used by: PISP

If the server is unable to complete the consent, or if an out-of-loop processing error or another processing error occurs, the error callback **PUT /tppConsents/**<ID>**/error** is used. The <ID> in the URI should contain the <ID> that was used in the **GET /tppConsents/**<ID> request or the **POST /tppConsents** request. The data model for this resource is as follows:

Name	Cardinality	Type	Description
errorInformation	1	ErrorInformation	Error code, category description.

3.1.7 parties

The **parties** resource will be used by the PISP to identify a party to a transfer. This will be used by the PISP to identify the payee DFSP when it requests a transfer.

The PISP will be permitted to issue a PUT /parties response. Although it does not own any transaction accounts, there are circumstances in which another party may want to pay a customer via their PISP identification: for instance, where the customer is at a merchant's premises and tells the merchant that they would like to pay via their PISP app. In these circumstances, the PISP will need to be able to confirm that it does act for the customer.

3.1.7.1 Requests

The **parties** resource will support the following requests.

3.1.7.1.1 GET /parties

Used by: DFSP and optionally the PISP

The **GET /parties** resource will use the same form as the resource described in Section 6.3.3.1 of Ref. 1 above.

3.1.7.2 Callbacks

The **parties** resource will support the following callbacks.

3.1.7.2.1 PUT /parties

Used by: DFSP

The **PUT /parties** resource will use the same form as the resource described in Section 6.3.4.1 of Ref. 1 above.

It should be noted, however, that the **Party** object returned from this resource has a different format from the **Party** object described in Section 7.4.11 of Ref. 1 above. The structure of this object is described in Section 3.2.1.29 below.

3.1.8 services

The **services** resource is a new resource which enables a participant to query for other participants who offer a particular service. The requester will issue a **GET** request, specifying the type of service for which information is required as part of the query string. The switch will respond with a list of the current DFSPs in the scheme which are registered as providing that service.

3.1.8.1 Requests

The **services** resource will support the following requests.

*3.1.8.2 **GET /services/**<Type>*

Used by: DFSP, PISP

The HTTP request **GET /services/**<Type> is used to find out the names of the participants in a scheme which provide the type of service defined in the <Type> parameter. The <Type> parameter should specify a value from the ServiceType enumeration. If it does not, the request will be rejected with an error.

Callback and data model information for GET /services/<Type>:

- Callback - PUT /services/<Type>
- Error Callback - PUT /services/<Type>/error
- Data Model – Empty body

3.1.8.3 Callbacks

This section describes the callbacks that are used by the server for services provided by the resource **/services**.

3.1.8.3.1 PUT /services/<Type>

Used by: Switch

The callback **PUT /services/**<Type> is used to inform the client of a successful result of the service information lookup. The information is returned in the following form:

Name	Cardinality	Type	Description
serviceProviders	1…n	FspId	A list of the Ids of the participants who provide the service requested.

3.1.8.3.2 PUT /services/<Type>/error

Used by: Switch

If the server encounters an error in fulfilling a request for a list of participants who provide a service, the error callback **PUT /services/**<Type>**/error** is used to inform the client that an error has occurred.

Name	Cardinality	Type	Description
errorInformation	1	ErrorInformation	Error code, category description.

3.1.9 tppAuthorizations

The **/tppAuthorizations** resource is used by a Payer DFSP to request authorization from the Payer to make a payment. This use case is applicable when the Payer DFSP receives a request to pay (a Payee initiated payment) and chooses to authorize the payment through the third party provider. This functionality is analogous to the authorisation provided in the **/tppTransfers** resource described in Section 3.1.10 below. On receipt of the request to pay, the Payer DFSP uses it to request authorisation through the PISP to execute a transfer. If the DFSP must included a challenge in its response to the **tppTransactionRequests** request which the PISP sent to initiate the transfer sequence, then the PISP includes the signed challenge in its request to the DFSP to execute the challenge:

1. Display the information defining the terms of a proposed transfer to its customer;

2. Obtain the customer's confirmation that they want the transfer to proceed;
3. Return a signed version of the terms which the DFSP can use to verify the consent

The **/tppAuthorizations** resource supports the endpoints described below.

3.1.9.1 Requests

This section describes the services that a client can request on the **/tppAuthorizations** resource.

3.1.9.1.1 GET /tppAuthorizations/<ID>

Used by: DFSP

The HTTP request **GET /tppAuthorizations/**<ID> is used to get information relating to a previously issued authorization request. The <ID> in the request should match the authorizationRequestId executionRequestId which was given when the authorization request was created.

Callback and data model information for **GET /tppAuthorizations***<ID>*:

- Callback - **PUT /tppAuthorizations/***<ID>*
- Error Callback - **PUT /tppAuthorizations/***<ID>***/error**
- Data Model – Empty body

3.1.9.1.2 POST /tppAuthorizations

Used by: DFSP

The HTTP request **POST /tppAuthorizations** is used to request the validation by a customer for the transfer described in the request.

Callback and data model information for **POST /tppAuthorizations**:

- Callback - **PUT /tppAuthorizations /**<ID>
- Error Callback - **PUT /tppAuthorizations/**<ID>**/error**
- Data Model – See Table below

Name	Cardinality	Type	Description
authorizationRequestId	1	CorrelationId	Common ID between the PISP and the Payer DFSP for the transaction execution request object. The ID should be reused for resends of the same transaction execution request. A new ID should be generated for each new transaction execution request.
transactionRequestId	1	CorrelationId	The unique identifier of the transaction request for which authorization execution is being requested.
challengeauthenticationInfo	0..1	BinaryStringAuthenticationInfo	The challenge that the PISP's client is to signEvidence that the challenge issued by the DFSP was correctly signed by the PISP application, and therefore that the customer has authorized the transfer..
transactionId	1	CorrelationId	The unique identifier for the proposed transaction. It is set by the payer DFSP and signed by the payee DFSP as part of the terms of the transfer
transferAmount	1	Money	The amount that will be debited from the sending customer's account as a consequence of the transaction.
payeeReceiveAmount	1	Money	The amount that will be credited to the receiving customer's account as a consequence of the transaction.
fees	1	Money	The amount of fees that the paying customer will be charged as part of the transaction.
payer	1	Party	Information about the Payer in the proposed transaction
payee	1	Party	Information about the Payee in the proposed transaction

Name	Cardinality	Type	Description
transactionType	1	TransactionType	The type of the transaction.
condition	1	IlpCondition	The condition returned by the payee DFSP to the payer DFSP as a cryptographic guarantee of the transfer.
expiration	1	DateTime	The time by which the transfer must be completed, set by the payee DFSP.
extensionList	0..1	ExtensionList	Optional extension, specific to deployment.

3.1.9.2 Callbacks

The following callbacks are supported for the **/tppAuthorizations** resource

3.1.9.2.1 PUT /tppAuthorizations/<ID>

Used by: PISP

The value in the <ID> field will be the executionRequestId that was included in the **POST /tppAuthorizations** to which this is a response. The **PUT /tppAuthorizations/**<ID> resource will have the same content as the **PUT /tppAuthorizations/**<ID> resource described in Section 6.6.4.1 of Ref. 1 above.

3.1.9.3 Error callbacks

This section describes the error callbacks that are used by the server under the resource **/tppAuthorizations**.

3.1.9.3.1 PUT /tppAuthorizations/<ID>/error

Used by: DFSP

The **PUT /tppAuthorizations/**<ID>**/error** resource will have the same content as the **PUT /tppAuthorizations/**<ID>**/error** resource described in Section 6.6.5.1 of Ref. 1 above.

3.1.10 tppTransfers

1. The **/tppTransfers** resource is analogous to the **/transfers** resource described in Section 6.6 of Ref. 1 above. The PISP uses it to request the DFSP to execute a transfer. If the DFSP has included a challenge in its response to the **tppTransactionRequests** request which the PISP sent to initiate the transfer sequence, then the PISP includes the signed challenge in its request to the DFSP to execute the challenge

The **/tppTransfers** resource supports the endpoints described below.

3.1.10.1 Requests

This section describes the services that a client can request on the **/tppTransfers** resource.

3.1.10.1.1 GET /tppTransfers/<ID>

Used by: DFSP

The HTTP request **GET /tppTransfers /**<ID> is used to get information relating to a previously issued authorization request. The <ID> in the request should match the executionRequestId which was given when the authorization request was created.

Callback and data model information for **GET /tppTransfers/***<ID>*:

- Callback - **PUT /tppTransfers /***<ID>*
- Error Callback - **PUT /tppTransfers /***<ID>***/error**
- Data Model – Empty body

3.1.10.1.2 POST /tppTransfers

Used by: DFSP

The HTTP request **POST /tppTransfers** is used to request the validation by a customer for the transfer described in the request.

Callback and data model information for **POST /tppTransfers**:

- Callback - **PUT /tppTransfers /**<ID>
- Error Callback - **PUT /tppTransfers /**<ID>**/error**
- Data Model – See Table below

Name	Cardinality	Type	Description
executionRequestId	1	CorrelationId	Common ID between the PISP and the Payer DFSP for the execution request object. The ID should be reused for resends of the same execution request. A new ID should be generated for each new execution request.
transactionRequestId	1	CorrelationId	The unique identifier of the transaction request for which execution is being requested.
authenticationInfo	0..1	AuthenticationInfo	Evidence that the challenge issued by the DFSP was correctly signed by the PISP application, and therefore that the customer has authorized the transfer..
extensionList	0..1	ExtensionList	Optional extension, specific to deployment.

3.1.10.2 Callbacks

The following callbacks are supported for the **/tppTransfers** resource

3.1.10.2.1 PUT /tppTransfers/<ID>

Used by: PISP

The value in the <ID> field will be the executionRequestId that was included in the **POST /tppTransfers** to which this is a response. The **PUT /tppTransfers/**<ID> resource will have the same content as the **PUT /transfers/**<ID> resource described in Section 6.6.4.1 of Ref. 1 above.

3.1.10.3 Error callbacks

This section describes the error callbacks that are used by the server under the resource **/tppTransfers**.

3.1.10.3.1 PUT /tppTransfers/<ID>/error

Used by: DFSP

The **PUT /tppTransfers/**<ID>**/error** resource will have the same content as the **PUT /transfers/**<ID>**/error** resource described in Section 6.6.5.1 of Ref. 1 above.

3.1.11 tppTransactionRequests

The **/tppTransactionRequests** resource is analogous to the **/transactionRequests** resource described in Section 6.4 of Ref. 1 above. The PISP uses it to request the owner of the PISP's customer's account to transfer a specified amount from the customer's account with the DFSP to a named Payee.

The **/tppTransactionRequests** resource supports the endpoints described below.

3.1.11.1 Requests

This section describes the services that a client can request on the **/tppTransactionRequests** resource.

3.1.11.1.1 GET /tppTransactionRequests/<ID>

Used by: PISP

The HTTP request **GET /tppTransactionRequests/**<ID> is used to get information relating to a previously issued transaction request. The <ID> in the request should match the transactionRequestId which was given when the transaction request was created (see Section 3.1.7.1.2 below).

Callback and data model information for **GET /tppTransactionRequests/***<ID>*:

- Callback - **PUT /tppTransactionRequests /***<ID>*
- Error Callback - **PUT /tppTransactionRequests /***<ID>***/error**

- Data Model – Empty body

3.1.11.1.2 POST /tppTransactionRequests

Used by: PISP

The HTTP request **POST /tppTransactionRequests** is used to request the creation of a transaction request on the server for the transfer described in the request.

Callback and data model information for **POST /tppTransactionRequests**:

- Callback - **PUT /tppTransactionRequests /**<ID>
- Error Callback - **PUT /tppTransactionRequests /**<ID>**/error**
- Data Model – See Table below

Name	Cardinality	Type	Description
transactionRequestId	1	CorrelationId	Common ID between the PISP and the Payer DFSP for the transaction request object. The ID should be reused for resends of the same transaction request. A new ID should be generated for each new transaction request.
payee	1	Party	Information about the Payee in the proposed financial transaction.
payer	1	PartyIdInfo	Information about the Payer type, id, sub-type/id, FSP Id in the proposed financial transaction.
amount	1	Money	Requested amount to be transferred from the Payer to Payee.
transactionType	1	TransactionType	Type of transaction
note	0..1	Note	Reason for the transaction request, intended for the Payer.
authenticationType	0..1	AuthenticationType	OTP, FIDO or QR Code, otherwise empty.
expiration	0..1	DateTime	Can be set to get a quick failure in case the peer FSP takes too long to respond. Also, it may be beneficial for Consumer, Agent, Merchant to know that their request has a time limit.
extensionList	0..1	ExtensionList	Optional extension, specific to deployment.

3.1.11.2 Callbacks

The following callbacks are supported for the **/tppTransactionRequests** resource

3.1.11.2.1 PUT /tppTransactionRequests/<ID>

Used by: DFSP

The **PUT /tppTransactionRequests/**<ID> resource will have the following content. The <ID> in the request should match the transactionRequestId which was given when the transaction request was created (see Section 3.1.7.1.2 above). :

Name	Cardinality	Type	Description
transactionRequestId	1	CorrelationId	The unique identifier of the transaction request to which this is a response.
challenge	1	BinaryString	The challenge that the PISP's client is to sign.
transactionId	1	CorrelationId	The unique identifier for the proposed transaction. It is set by the payer DFSP and signed by the payee DFSP as part of the terms of the transfer
transferAmount	1	Money	The amount that will be debited from the sending customer's account as a consequence of the transaction.
payeeReceiveAmount	1	Money	The amount that will be credited to the receiving customer's account as a consequence of the transaction.

Name	Cardinality	Type	Description
fees	1	Money	The amount of fees that the paying customer will be charged as part of the transaction.
payer	1	Party	Information about the Payer in the proposed transaction
payee	1	Party	Information about the Payee in the proposed transaction
transactionType	1	TransactionType	The type of the transaction.
condition	1	IlpCondition	The condition returned by the payee DFSP to the payer DFSP as a cryptographic guarantee of the transfer.
expiration	1	DateTime	The time by which the transfer must be completed, set by the payee DFSP.
extensionList	0..1	ExtensionList	Optional extension, specific to deployment.

3.1.11.3 Error callbacks

This section describes the error callbacks that are used by the server under the resource **/tppTransactionRequests.**

3.1.11.3.1 PUT /tppTransactionRequests/<ID>/error

Used by: DFSP

The **PUT /tppTransactionRequests/**<ID>**/error** resource will have the same content as the **PUT /transactionRequests/**<ID>**/error** resource described in Section 6.4.5.1 of Ref. 1 above.

3.1.12 tppVerifications

The **/tppVerifications** resource is used by a Payer DFSP to verify that an authorization response received from a PISP was signed using the correct key, in cases where the authentication service to be used is implemented by the switch and not internally by the DFSP. The DFSP sends the original challenge and the signed response to the authentication service, together with the consent ID to be used for the verification. The authentication service compares the response with the result of signing the challenge with the private key associated with the consent ID, and, if the two match, it returns a positive result. Otherwise, it returns an error.

The **/tppVerifications** resource supports the endpoints described below.

3.1.12.1 Requests

This section describes the services that a client can request on the **/tppVerifications** resource.

3.1.12.1.1 GET /tppVerifications/<ID>

Used by: DFSP

The HTTP request **/tppVerifications** <ID> is used to get information regarding a previously created or requested authorization. The <ID> in the URI should contain the verification request ID (see Section 3.1.9.1.2 below) that was used for the creation of the transfer.Callback and data model information for GET **/tppVerifications/**<ID>:

Callback – PUT **/tppVerifications/**<ID>

Error Callback – PUT **/tppVerifications/**<ID>/error

Data Model – Empty body

3.1.12.1.2 POST /tppVerifications

Used by: DFSP

The **POST /tppVerifications** resource is used to request confirmation from an authentication service that a challenge has been signed using the correct private key.

Callback and data model information for **POST /tppVerifications**:

- Callback - **PUT /tppVerifications /**<ID>
- Error Callback - **PUT /tppVerifications /**<ID>**/error**
- Data Model – See Table below

Name	Cardinality	Type	Description
verificationRequestId	1	CorrelationId	Common ID between the DFSP and authentication service for the verification request object. The ID should be reused for resends of the same authorization request. A new ID should be generated for each new authorization request.
challenge	1	Challenge	The challenge originally sent to the PISP
value	1	authenticationValue	The signed challenge returned by the PISP.
consentId	1	CorrelationId	Common Id between the DFSP and the authentication service for the agreement against which the authentication service is to evaluate the signature

3.1.12.2 Callbacks

This section describes the callbacks that are used by the server under the resource **/tppVerifications/**

3.1.12.2.1 **PUT /tppVerifications/**<ID>

Used by: FIDO

The callback **PUT /tppVerifications/**<ID> is used to inform the client of the result of an authorization check. The <ID> in the URI should contain the authorizationRequestId (see Section 3.1.9.1.2 above) which was used to request the check, or the <ID> that was used in the **GET /tppVerifications/**<ID>. The data model for this resource is as follows:

Name	Cardinality	Type	Description
authorizationResponse	1	AuthenticationResponse	The result of the authorization check.

3.1.12.3 Error callbacks

This section describes the error callbacks that are used by the server under the resource **/tppVerifications**.

3.1.12.3.1 **PUT /tppVerifications/**<ID>**/error**

Used by: FIDO

If the server is unable to complete the authorization request, or another processing error occurs, the error callback **PUT /tppVerifications/**<ID>**/error** is used. The <ID> in the URI should contain the <ID> that was used in the call which requested the authorization. The data model for this resource is as follows:

Name	Cardinality	Type	Description
errorInformation	1	ErrorInformation	Error code, category description.

3.2 Data Models

The following additional data models will be required to support the PISP API

3.2.1 Element definitions

3.2.1.1 Account

The Account data model contains information relating to an account

Name	Cardinality	Type	Description
address	0..1	AccountAddress	An address which can be used to identify the account
currency	1	Currency	The currency in which the account is denominated
accountNickname	0..1	Name	Display name of the account, as set by the account owning DFSP. This will normally be a type name, such as “Transaction Account” or “Savings Account”

3.2.1.2 AccountAddress

The `AccountAddress` data type is a variable length string with a maximum size of 1023 characters and consists of:

- Alphanumeric characters, upper or lower case. (Addresses are case-sensitive so that they can contain data encoded in formats such as base64url.)
- Underscore (_)
- Tilde (~)
- Hyphen (-)
- Period (.) Addresses MUST NOT end in a period (.) character

An entity providing accounts to parties (i.e. a participant) can provide any value for an `AccountAddress` that is **routable** to that entity. It does not need to provide an address that makes the account identifiable outside the entity’s domain. i.e. This is an address not an identifier

For example, a participant (Blue DFSP) that has been allocated the address space `moja.blue` might allocate a random UUID to the account and return the value:

```json
{
  "address": "moja.blue.8f027046-b82a-4fa9-838b-70210fcf8137",
  "currency": "ZAR"
}
```

*This address is **routable** to Blue DFSP because it uses the prefix `moja.blue`*

Blue DFSP may also simply use their own address if that is sufficient (in combination with the remainder of the `PartyIdInfo`) to uniquely identify the payee and the destination account.

```json
{
  "address": "moja.blue",
  "currency": "ZAR"
}
```

*This address is also **routable** to Blue DFSP because it uses the prefix `moja.blue`*

IMPORTANT: The policy for defining addresses and the life-cycle of these is at the discretion of the address space owner (the payer DFSP in this case).

3.2.1.3 AccountList

The AccountList data model is used to hold information about the accounts that a party controls.

Name	Cardinality	Type	Description
account	1..n	Account	Information relating to an account that a party controls.

3.2.1.4 AuthenticationChannel

The *AuthenticationChannel* data model is used to specify the type of out-of-loop authentication to use in verifying a customer's wish to grant permissions to a PISP.

Name	Cardinality	Type	Description
AuthenticationChannel	1	Enum of String(1..32)	See Section 3.2.2.2 below for more information on allowed values.

3.2.1.5 AuthenticationInfo

The **AuthenticationInfo** data type used in these definitions is as defined in Section 7.4.1 of Ref. 1 above.

3.2.1.6 AuthenticationResponse

The **AuthenticationResponse** data type is an enumeration of type AuthenticationResponse.

3.2.1.7 AuthenticationType

The **AuthenticationType** data type used in these definitions is as defined in Section 7.5.2 of Ref. 1 above. It is enumerated by the AuthorizationChannelType enumeration.

3.2.1.8 AuthenticationValue

The AuthenticationValue data element contains a response returned by the recipient of an authorization request. It is described in Section 7.3.3 of Ref. 1 above, and is extended to support the new authentication type used for PISP. The data model is as follows:

Name	Cardinality	Format	Description
AuthenticationValue	1	Depending on AuthenticationType: If OTP: OtpValue; If QRCODE: String(1..64); If U2F: BinaryString	Contains the authentication value. The format depends on the authentication type used in the AuthenticationInfo complex type.

3.2.1.9 AuthenticatorAttestationResponse

The AuthenticatorAttestationResponse object is used to store information relating to a credential which a PISP has created on a user's device. It contains the following items of information.

Name	Cardinality	Type	Description
type	1	WebAuthenticationType	An enumeration which describes whether the information relates to a newly created credential or to the receipt of an existing credential.
challenge	1	BinaryString	The base64url encoded version of the cryptographic challenge sent from the relying party's server.
origin	1	string	The fully qualified origin of the requester which has been given by the client/browser to the authenticator.
tokenBindingId	0..1	TokenBindingState	An object describing the state of the token binding protocol for the communication with the relying party.

3.2.1.10 BinaryString

The **BinaryString** type used in these definitions is as defined in Section 7.2.17 of Ref. 1 above.

3.2.1.11 BinaryString32

The **BinaryString32** type used in these definitions is as defined in Section 7.2.18 of Ref. 1 above.

3.2.1.12 Challenge

The Challenge object is used to hold a FIDO challenge and its associated signature.

Name	Cardinality	Type	Description
payload	1	String	The value to be signed by the PISP
signature	0..1	BinaryString(256)	The signature produced by the application of the PISP's private key to the payload.

3.2.1.13 ConsentRequestChannelType

The **ConsentRequestChannelType** is used to hold an instance of the ConsentRequestChannelType enumeration. Its data model is as follows:

Name	Cardinality	Type	Description
ConsentRequestChannelType	1	Enum of String(1..32)	See Section 3.2.2.4 below (ConsentRequestChannelType) for more information on allowed values.

3.2.1.14 ConsentState

The ConsentState type stores the status of a consent request, as described in Section 3.1.3.2.2 above. Its data model is as follows:

Name	Cardinality	Type	Description
ConsentState	1	Enum of String(1..32)	See Section 3.2.2.5 below (ConsentStatusType) for more information on allowed values.

3.2.1.15 CorrelationId

The **CorrelationId** type used in these definitions is as defined in Section 7.3.8 of Ref. 1 above.

3.2.1.16 Credential

The Credential object is used to store information about a challenge which is exchanged with an authentication service. The data model is as follows:

Name	Cardinality	Type	Description
credentialId	1	CorrelationId	A unique identifier for the credential.
credentialType	1	AuthenticationChannel	The type of credential this is
status	0..1	CredentialState	The current status of the credential.
payload	1	The type of this field depends on the type of credential this is, as defined in the **credentialType** field: • If the credential type is FIDO, then the type of the payload will be PublicKeyCredential. • If the credential type is	A description of the credential and information which allows the recipient of the credential to test its veracity.

Name	Cardinality	Type	Description
		GENERIC, then the type of the payload will be GenericCredential.	

3.2.1.17 CredentialState

The **CredentialState** data type stores the state of a credential request. Its data model is as follows.

Name	Cardinality	Type	Description
CredentialState	1	Enum of String(1..32)	See Section 3.2.2.5 below (CredentialState) for more information on allowed values.

3.2.1.18 DateTime

The **DateTime** data type used in these definitions is as defined in Section 7.2.14 of Ref. 1 above.

3.2.1.19 ErrorInformation

The **ErrorInformation** data type used in these definitions is as defined in Section 7.4.2 of Ref. 1 above

3.2.1.20 ExtensionList

The **ExtensionList** data type used in these definitions is as defined in Section 7.4.4 of Ref. 1 above.

3.2.1.21 FspId

The **FspId** data type used in these definitions is as defined in Section 7.3.16 of Ref. 1 above.

3.2.1.22 GeoCode

The **GeoCode** data type used in these definitions is as defined in Section 7.4.9 of Ref. 1 above.

3.2.1.23 GenericCredential

The GenericCredential object stores the payload for a credential which is validated according to a comparison of the signature created from the challenge using a private key against the same challenge signed using a public key. Its content is as follows.

Name	Cardinality	Type	Description
publicKey	0..1	BinaryString32	The public key to be used in checking the signature. Only required if the public key has not already been registered.
signature	1	BinaryString32	The signature to be checked against the public key.

3.2.1.24 ilpCondition

The **ilpCondition** type used in these definitions is as defined in Section 7.3.17 of Ref. 1 above.

3.2.1.25 Integer

The **Integer** type used in these definitions is as defined in Section 7.2.5 of Ref. 1 above.

3.2.1.26 Money

The **Money** type used in these definitions is a defined in Section 7.4.10 of Ref. 1 above.

3.2.1.27 Note

The **Note** data type used in these definitions is as defined in Section 7.3.23 of Ref. 1 above.

3.2.1.28 Party

The following shows a proposed revision of the Party data element to support the additional information required to support PISP interactions.

Name	Cardinality	Type	Description
partyIdInfo	1	PartyIdInfo	Party Id type, id, sub ID or type, and FSP Id.
merchantClassificationCode	0..1	MerchantClassificationCode	Used in the context of Payee Information, where the Payee happens to be a merchant accepting merchant payments.
name	0..1	PartyName	Display name of the Party, could be a real name or a nick name.
personalInfo	0..1	PartyPersonalInfo	Personal information used to verify identity of Party such as first, middle, last name and date of birth.
accounts	0..1	AccountList	A list of the accounts that the party has.

3.2.1.29 PartyIdInfo

The **PartyIdInfo** data type used in these definitions is as defined in Section 7.4.13 of Ref. 1 above.

3.2.1.30 PublicKeyCredential

The PublicKeyCredential object contains information about a credential created on a device by a PISP. It contains the following items of information.

Name	Cardinality	Type	Description
credentialId	1	CorrelationId	An identifier for the credential
attestationResponse	1	AuthenticationAttestationResponse	Information about the credential that was set up on the user's device.

3.2.1.31 Quote

The **Quote** object is used to collect the information on the terms of a transfer which a Payee DFSP returns as part of the positive response to a quotation. This information is forwarded to the PISP by the Payer DFSP so that the PISP's customer can make an informed consent to the transfer, and is forwarded to the authentication service (if one is used) to confirm the *bona fides* of the authorization received from the PISP.

Name	Cardinality	Type	Description
transferAmount	1	Money	The amount that the sender's account will be debited
payeeReceiveAmount	1	Money	The amount of Money that the Payee should receive in the end-to-end transaction
fees	0..1	Money	The fees that the sender will pay as part of the transfer.
expiration	0..1	DateTime	Date and time until when the quotation is valid and can be honored when used in the subsequent transaction.
transactionType	1	TransactionType	Type of the transaction.
note	0..1	Note	Memo associated to the transaction, intended to the Payee.
extensionList	0..1	ExtensionList	Optional extension, specific to deployment.

3.2.1.32 Scope

The Scope element contains an identifier defining, in the terms of a DFSP, an account on which access types can be requested or granted. It also defines the access types which are requested or granted.

Name	Cardinality	Type	Description
accountId	1	AccountAddress	The address of the account to which the PISP wishes to be permitted access, or is being granted access
actions	1..n	ScopeAction	The action that the PISP wants permission to take in relation to the customer's account, or that it has been granted in relation to the customer's account
credential	0..1	Credential	The credential which is to be applied to the scope.
partyIdInfo	0..1	PartyIdInfo	The identifier which the PISP should use to access the account.

3.2.1.33 ScopeAction

The ScopeAction element contains an access type which a PISP can request from a DFSP, or which a DFSP can grant to a PISP. It must be a member of the appropriate enumeration.

Name	Cardinality	Type	Description
scopeAction	1	Enum of String(1..32)	See Section 0 below (ScopeEnumeration) for more information on allowed values.

3.2.1.34 ServiceType

The ServiceType element contains a type of service where the requester wants a list of the participants in the scheme which provide that service. It must be a member of the appropriate enumeration.

Name	Cardinality	Type	Description
serviceType	1	Enum of String(1..32)	See Section 3.2.2.9 below ServiceType) for more information on allowed values.

3.2.1.35 TokenBindingState

The **TokenBindingState** object describes the state of a token binding protocol for communication with a relying party for a public key credential. It contains the following items of information.

Name	Cardinality	Type	Description
status	1	TokenBindingStateStatus	Denotes whether or not token binding has been used to negotiate with the relying party.
id	1	String	The base64url encoding of the token binding ID which was used for the communication.

3.2.1.36 Transaction

The **Transaction** type used in these definitions is as defined in Section 7.4.17 of Ref. 1 above, but with extensions to include the additional information required for verification and consent in the PISP ecosystem.

Name	Cardinality	Type	Description
transactionId	1	CorrelationId	ID of the transaction. Decided by the Payer FSP during the creation of the quote.
quoteId	1	CorrelationId	ID of the quote. Decided by the Payer FSP during the creation of the quote.

Name	Cardinality	Type	Description
transactionRequestId	1	CorrelationId	ID of the transaction request which the PISP used to request the transfer
payee	1	Party	Information about the Payee in the proposed financial transaction.
payer	1	Party	Information about the Payer in the proposed financial transaction.
amount	1	Money	Transaction amount to be sent.
payeeReceiveAmount	1	Money	The amount of Money that the Payee should receive in the end-to-end transaction.
customerCost	0..1	Money	The charges that the customer will pay as part of the transaction.
expiration	0..1	DateTime	Date and time until when the quotation is valid and can be honored when used in the subsequent transaction.
transactionType	1	TransactionType	Type of the transaction.
note	0..1	Note	Memo associated to the transaction, intended to the Payee.
extensionList	0..1	ExtensionList	Optional extension, specific to deployment.

3.2.1.37 TransactionType

The *TransactionType* type used in these definitions is as defined in Section 7.4.18 of Ref. 1 above.

3.2.1.38 TransferState

The *TransferState* type used in these definitions is as defined in Section 7.3.35 of Ref. 1 above.

3.2.1.39 Uri

The API data type **Uri** is a JSON string in a canonical format that is restricted by a regular expression for interoperability reasons. The regular expression for restricting the **Uri** type is as follows:

```
^(([^:/?#]+):)?(//([^/?#]*))?([^?#]*)(\?([^#]*))?(#(.*))?
```
[5]

3.2.2 Enumerations

3.2.2.1 AuthenticationResponse

The AuthenticationResponse enumeration describes the result of authenticating a FIDO challenge.

Name	Description
VERIFIED	The challenge was correctly signed.
REJECTED	The challenge was not correctly signed.
RESEND	A problem occurred. Please re-submit.

3.2.2.2 AuthorizationChannelType

This is an extension of the AuthenticationType enumeration described in Section 7.5.2 of Ref. 1 above.

Name	Description
OTP	One-time password generated by the Payer FSP.
QRCODE	QR code used as One Time Password.
U2F	A FIDO challenge

3.2.2.3 AuthorizationResponse

The **AuthorizationResponseType** enumeration is the same as the **AuthorizationResponse** enumeration described in Section 7.5.3 of Ref. 1 above.

[5] Taken from RFC 3986, Appendix B

3.2.2.4 ConsentRequestChannelType

Name	Description
WEB	PISP can support authorization via a web-based login
OTP	PISP can support authorization via a One Time PIN

3.2.2.5 ConsentStatusType

The ConsentStatusType enumeration describes the allowed status values that a consent item can have. These are as follows:

Name	Description
PENDING	The consent item has been proposed but not yet approved.
VERIFIED	The consent item has been verified and approved.

3.2.2.6 CredentialState

This contains the allowed values for the state of a credential state

Name	Description
RECEIVED	FIDO Server has received the credential.
PENDING	Authentication service is validating the credential.
COMPLETED	Authentication service has successfully validated the credential.
REJECTED	Authentication service has rejected the credential.
VERIFIED	Authentication service has verified the credential

3.2.2.7 CredentialType

The CredentialType enumeration contains the allowed values for the type of credential which is associated with a permission.

Name	Description
FIDO	The credential is based on a FIDO challenge. Its payload is a PublicKeyCredential object.
GENERIC	The credential is based on a simple public key validation. Its payload is a GenericCredential object

3.2.2.8 PartyIdType

The PartyIdType enumeration is extended for PISPs to include a definition for the identifier which represents a link between a specific PISP and an account at a DFSP which a customer has given the PISP permission to access.

Name	Description
MSISDN	An MSISDN (Mobile Station International Subscriber Directory Number; that is, a phone number) is used in reference to a Party. The MSISDN identifier should be in international format according to the ITU-T E.164[37] standard. Optionally, the MSISDN may be prefixed by a single plus sign, indicating the international prefix.
EMAIL	An email is used in reference to a Party. The format of the email should be according to the informational RFC 3696[38].
PERSONAL_ID	A personal identifier is used in reference to a participant. Examples of personal identification are passport number, birth certificate number, and national registration number. The identifier number is added in the **PartyIdentifier** element. The personal identifier type is added in the **PartySubIdOrType** element.

Name	Description
BUSINESS	A specific Business (for example, an organization or a company) is used in reference to a participant. The BUSINESS identifier can be in any format. To make a transaction connected to a specific username or bill number in a Business, the **PartySubIdOrType** element should be used.
DEVICE	A specific device (for example, POS or ATM) ID connected to a specific business or organization is used in reference to a Party. For referencing a specific device under a specific business or organization, use the **PartySubIdOrType** element.
ACCOUNT_ID	A bank account number or FSP account ID should be used in reference to a participant. The ACCOUNT_ID identifier can be in any format, as formats can greatly differ depending on country and FSP.
IBAN	A bank account number or FSP account ID is used in reference to a participant. The IBAN identifier can consist of up to 34 alphanumeric characters and should be entered without whitespace.
ALIAS	An alias is used in reference to a participant. The alias should be created in the FSP as an alternative reference to an account owner. Another example of an alias is a username in the FSP system. The ALIAS identifier can be in any format. It is also possible to use the **PartySubIdOrType** element for identifying an account under an Alias defined by the **PartyIdentifier**.
THIRD_PARTY_LINK	A third-party link which represents an agreement between a specific PISP and a customer's account at a DFSP. The content of the link is created by the DFSP at the time when it gives permission to the PISP for specific access to a given account.

3.2.2.9 ScopeEnumeration

Name	Description
BALANCE_ENQUIRY	PISP can request a balance for the linked account
FUNDS_TRANSFER	PISP can request a transfer of funds from the linked account in the DFSP
STATEMENT	PISP can request a statement of individual transactions on a user's account
PAYMENT_REQUEST	PISP can initiate a payment request into the user's linked account

3.2.2.10 ServiceType

The **ServiceType** enumeration describes the types of role for which a DFSP may query using the **/services** resource.

Name	Description
THIRD_PARTY_DFSP	DFSPs which will support linking with PISPs
PISP	PISPs
FIDO_SERVER	Servers which provide FIDO authentication services

3.2.2.11 TokenBindingStateStatus

The TokenBindingStateStatus enumeration describes the possible status values for a token binding state object associated with a public key credential. It forms part of the **TokenBindingState** object.

Name	Description
supported	The client supports token binding but did not negotiate with the relying party.
present	Token binding was used already.

3.2.2.12 WebAuthenticationType

The **WebAuthenticationType** enumeration defines the type of a web authentication credential. It forms part of the **AuthenticatorAttestationResponse** object.

Name	Description
webauthn.get	An existing credential was retrieved
webauthn.create	A new credential was created