# Logical Data Model

*Open API for FSP Interoperability Specification*

# Logical Data Model

## *Open API for FSP Interoperability Specification*

### Table of Contents

## Table of Figures

## Table of Tables

## Table of Listings

# 1 Preface

This section contains information about how to use this document.

## 1.1 Conventions Used in This Document

The following conventions are used in this document to identify the specified types of information

| Type of Information | Convention | Example |
|---|---|---|
| **Elements of the API, such at resources** | Boldface | **/authorization** |
| **Variables** | Italics witin angle brackets | *<ID>* |
| **Glossary terms** | Italics on first occurence; defined in *Glossary* | The purpose of the API is to enable interoperable financial transactions between a *Payer* (a payer of electronic funds in a payment transaction) located in one *FSP* (an entity that provides a digital financial service to an end user) and a *Payee* (a recipient of electronic funds in a payment transaction) located in another FSP. |
| **Library documents** | Italics | User information should, in general, not be used by API deployments; the security measures detailed in *API Signature* and *API Encryption* should be used instead. |

## 1.2 Document Version Information

| Version | Date | Change Description |
|---|---|---|
| **1.0** | 2018-03-13 | Initial version |

# 2 Introduction

This document specifies the logical data model used by Open API (Application Programming Interface) for FSP (Financial Service Provider) Interoperability (hereafter cited as “the API”). .

Section 3 lists elements used by each service.

Section 0 describes the data model in terms of basic elements, simple data types and complex data types.

## 2.1 Open API for FSP Interoperability Specification

The Open API for FSP Interoperability Specification includes the following documents.

### 2.1.1 General Documents

- *Glossary*

### 2.1.2 Logical Documents

- *Logical Data Model*
- *Generic Transaction Patterns*
- *Use Cases*

### 2.1.3 Asynchronous REST Binding Documents

- *API Definition*
- *JSON Binding Rules*
- *Scheme Rules*

#### 2.1.3.1 Data Integrity, Confidentiality, and Non-Repudiation

- *PKI Best Practices*
- *Signature*
- *Encryption*

# 3 API Services Elements

The section identifies and describes elements used by each service.

## 3.1 API Resource Participants

This section describes the data model of services for the resource **Participants**.

### 3.1.1 Requests

This section describes the data model of services that can be requested by a client in the API for the resource **Participants**.

#### 3.1.1.1 Lookup Participant Information

Table 1 contains the data model for **Lookup Participant Information**.

| Name | Cardinality | Type | Description |
|---|---|---|---|
| **partyIdType** | 1 | PartyIdType | The type of the identifier. |
| **partyIdentifier** | 1 | PartyIdentifier | An identifier for the Party. |
| **partySubIdOrType** | 0..1 | PartySubIdOrType | A sub-identifier or sub-type for the Party. |

**Table 1 – Lookup Participant Information data model**

#### 3.1.1.2 Create Participant Information

Table 2 contains the data model for **Create Participant Information**.

| Name | Cardinality | Type | Description |
|---|---|---|---|
| **partyIdType** | 1 | PartyIdType | The type of the identifier. |
| **partyIdentifier** | 1 | PartyIdentifier | An identifier for the Party. |
| **partySubIdOrType** | 0..1 | PartySubIdOrType | A sub-identifier or sub-type for the Party. |
| **fspId** | 1 | FspId | FSP Identifier that the Party belongs to. |
| **currency** | 0..1 | Currency | Indicate that the provided Currency is supported by the Party. |

**Table 2 – Create Participant Information data model**

#### 3.1.1.3 Create Bulk Participant Information

Table 3 contains the data model for **Create Bulk Participant Information**.

| Name | Cardinality | Type | Description |
|---|---|---|---|
| **requestId** | 1 | CorrelationId | The ID of the request, determined by the client. Used for identification of the callback from the server. |
| **partyList** | 1..10000 | PartyIdInfo | List of Party elements that the Client would like to update or create FSP information about. |
| **currency** | 0..1 | Currency | Indicate that the provided Currency is supported by each PartyIdInfo in the list. |

**Table 3 – Create Bulk Participant Information data model**

#### 3.1.1.4 Delete Participant Information

Table 4 contains the data model for **Delete Participant Information**.

| Name | Cardinality | Type | Description |
|---|---|---|---|
| **partyIdType** | 1 | PartyIdType | The type of the identifier. |
| **partyIdentifier** | 1 | PartyIdentifier | An identifier for the Party. |
| **partySubIdOrType** | 0..1 | PartySubIdOrType | A sub-identifier or sub-type for the Party. |

**Table 4 – Delete Participant Information data model**

### 3.1.2 Responses

This section describes the data model of responses used by the server in the API for services provided by the resource **Participants**.

#### 3.1.2.1 Return Participant Information

Table 5 contains the data model for **Return Participant Information**.

| Name | Cardinality | Type | Description |
|---|---|---|---|
| **partyIdType** | 1 | PartyIdType | The type of the identifier. |
| **partyIdentifier** | 1 | PartyIdentifier | An identifier for the Party. |
| **partySubIdOrType** | 0..1 | PartySubIdOrType | A sub-identifier or sub-type for the Party. |
| **fspId** | 0..1 | FspId | FSP Identifier that the Party belongs to. |

**Table 5 – Return Participant Information data model**

#### 3.1.2.2 Return Bulk Participant Information

Table 6 contains the data model for Return Bulk Participant Information.

| Name | Cardinality | Type | Description |
|---|---|---|---|
| **requestId** | 1 | CorrelationId | The ID of the request, determined by the client. Used for identification of the callback from the server. |
| **partyList** | 1..10000 | PartyResult | List of PartyResult elements for which creation was attempted (and either succeeded or failed). |
| **Currency** | 0..1 | Currency | Indicates that the provided Currency was set to be supported by each successfully added PartyIdInfo. |

**Table 6 – Return Bulk Participant Information data model**

### 3.1.3 Error Responses

This section describes the data model of the error responses used by the server in the API for services provided by the resource **Participants**.

#### 3.1.3.1 Return Participant Information Error

Table **7** contains the data model for **Return Participant Information Error**.

| Name | Cardinality | Type | Description |
|---|---|---|---|
| **partyIdType** | 1 | PartyIdType | The type of the identifier. |
| **partyIdentifier** | 1 | PartyIdentifier | An identifier for the Party. |
| **partySubIdOrType** | 0..1 | PartySubIdOrType | A sub-identifier or sub-type for the Party. |
| **errorInformation** | 1 | ErrorInformation | Error code, category description. |

**Table** 7 **– Return Participant Information Error data model**

#### 3.1.3.2 Return Bulk Participant Information Error

Table 8 contains the data model for **Return Bulk Participant Information Error**.

| Name | Cardinality | Type | Description |
|---|---|---|---|
| **requestId** | 1 | CorrelationId | The ID of the request, determined by the client. Used for identification of the callback from the server. |
| **errorInformation** | 1 | ErrorInformation | Error code, category description. |

**Table 8 – Return Bulk Participant Information Error data model**

## 3.2 API Resource Parties

This section describes the data model of services for the resource **Parties**.

### 3.2.1 Requests

This section describes the data model of services that can be requested by a client in the API for the resource **Parties**.

#### 3.2.1.1 Lookup Party Information

Table 9 contains the data model for **Lookup Party Information**.

| Name | Cardinality | Type | Description |
|---|---|---|---|
| **partyIdType** | 1 | PartyIdType | The type of the identifier. |
| **partyIdentifier** | 1 | PartyIdentifier | An identifier for the Party. |
| **partySubIdOrType** | 0..1 | PartySubIdOrType | A sub-identifier or sub-type for the Party. |

**Table 9 – Lookup Party Information data model**

### 3.2.2 Responses

This section describes the data model of responses used by the server in the API for services provided by the resource **Parties**.

#### 3.2.2.1 Return Party Information

Table 10 contains the data model for **Return Party Information**.

<table>
<tr><th>Name</th><th>Cardinality</th><th>Type</th><th>Description</th></tr>
<tr><td><b>partyIdType</b></td><td>1</td><td>PartyIdType</td><td>The type of the identifier.</td></tr>
<tr><td><b>partyIdentifier</b></td><td>1</td><td><b>NrOfRetries</b><br>Table 62 contains the data model for the element <b>NrOfRetries</b>.<br><table><tr><th>Name</th><th>Cardinality</th><th>Type</th><th>Description</th></tr><tr><td><b>NrOfRetries</b></td><td>1</td><td>Integer(1..2)</td><td>Number of retries.</td></tr></table><b>Table 62 – Element NrOfRetries</b><br>PartyIdentifier</td><td>An identifier for the Party.</td></tr>
<tr><td><b>partySubIdOrType</b></td><td>0..1</td><td>PartySubIdOrType</td><td>A sub-identifier or sub-type for the Party.</td></tr>
<tr><td><b>party</b></td><td>1</td><td>Party</td><td>Information regarding the requested Party.</td></tr>
</table>

**Table 10 – Return Party Information data model**

## 3.2.3 Error Responses

### 3.2.3.1 Return Party Information Error

Table 11 contains the data model for **Return Party Information Error**.

<table>
<tr><th>Name</th><th>Cardinality</th><th>Type</th><th>Description</th></tr>
<tr><td><b>partyIdType</b></td><td>1</td><td>PartyIdType</td><td>The type of the identifier.</td></tr>
<tr><td><b>partyIdentifier</b></td><td>1</td><td>NrOfRetries<br>Table 62 contains the data model for the element <b>NrOfRetries</b>.<br><table><tr><th>Name</th><th>Cardinality</th><th>Type</th><th>Description</th></tr><tr><td><b>NrOfRetries</b></td><td>1</td><td>Integer(1..2)</td><td>Number of retries.</td></tr></table><b>Table 62 – Element NrOfRetries</b><br>PartyIdentifier</td><td>An identifier for the Party.</td></tr>
<tr><td><b>partySubIdOrType</b></td><td>0..1</td><td>PartySubIdOrType</td><td>A sub-identifier or sub-type for the Party.</td></tr>
<tr><td><b>errorInformation</b></td><td>1</td><td>ErrorInformation</td><td>Error code, category description.</td></tr>
</table>

**Table 11 – Return Party Information Error data model**

## 3.3 API Resource Transaction Requests

This section describes the data model of services for the resource **Transaction Requests**.

### 3.3.1 Requests

This section describes the data model of services that can be requested by a client in the API for the resource **Transaction Requests**.

#### 3.3.1.1 Retrieve Transaction Request

Table 12 contains the data model for **Retrieve Transaction Request**.

| Name | Cardinality | Type | Description |
|---|---|---|---|
| **transactionRequestId** | 1 | CorrelationId | The common ID between the FSPs for the transaction request object, determined by the Payee FSP. The ID should be re-used for re-sends of the same transaction request. A new ID should be generated for each new transaction request. |

**Table 12 – Retrieve Transaction Request data model**

#### 3.3.1.2 Perform Transaction Request Information

Table 13 contains the data model for Perform Transaction Request Information.

| Name | Cardinality | Type | Description |
|---|---|---|---|
| **transactionRequestId** | 1 | CorrelationId | The common ID between the FSPs for the transaction request object, determined by the Payee FSP. The ID should be re-used for re-sends 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 | The requested amount to be transferred from the Payer to Payee. |
| **transactionType** | 1 | TransactionType | The type of transaction. |
| **note** | 0..1 | Note | Reason for the transaction request, intended to the Payer. |
| **geoCode** | 0..1 | GeoCode | Longitude and Latitude of the initiating party. Can be used to detect fraud. |
| **authenticationType** | 0..1 | AuthenticationType | OTP or QR Code, otherwise empty. |
| **expiration** | 0..1 | DateTime | Expiration is optional. It can be set to get a quick failure in case the peer FSP takes too long to respond. Also useful for notifying Consumer, Agent, Merchant that their request has a time limit. |
| **extensionList** | 0..1 | ExtensionList | Optional extension, specific to deployment. |

**Table 13 – Perform Transaction Request Information data model**

### 3.3.2 Responses

This section describes the data model of responses used by the server in the API for services provided by the resource **Transaction Requests**.

### 3.3.2.1 Return Transaction Request Information

Table 14 contains the data model for **Return Transaction Request Information**.

| Name | Cardinality | Type | Description |
|---|---|---|---|
| **transactionRequestId** | 1 | CorrelationId | The common ID between the FSPs for the transaction request object, determined by the Payee FSP. The ID should be re-used for re-sends of the same transaction request. A new ID should be generated for each new transaction request. |
| **transactionId** | 0..1 | CorrelationId | Identifies related /transactions (if a transaction has been created). |
| **transactionRequestState** | 1 | TransactionRequestState | The state of the transaction request. |
| **extensionList** | 0..1 | ExtensionList | Optional extension, specific to deployment. |

**Table 14 – Return Transaction Request Information data model**

## 3.3.3 Error Responses

This section describes the data model of the error responses used by the server in the API for services provided by the resource **Transaction Requests**.

### 3.3.3.1 Return Transaction Request Information Error

Table 15 contains the data model for **Return Transaction Request Information Error**.

| Name | Cardinality | Type | Description |
|---|---|---|---|
| **transactionRequestId** | 1 | CorrelationId | The common ID between the FSPs for the transaction request object, determined by the Payee FSP. The ID should be re-used for re-sends of the same transaction request. A new ID should be generated for each new transaction request. |
| **errorInformation** | 1 | ErrorInformation | Error code, category description. |

**Table 15 – Return Transaction Request Information Error data model**

## 3.4 API Resource Quotes

This section describes the data model of services for the resource **Quotes**.

### 3.4.1 Requests

This section describes the data model of services that can be requested by a client in the API for the resource **Quotes**.

#### 3.4.1.1 Retrieve Quote Information

Table 16 contains the data model for **Retrieve Quote Information**.

| Name | Cardinality | Type | Description |
|---|---|---|---|
| **quoteId** | 1 | CorrelationId | Identifies quote message. |

**Table 16 – Retrieve Quote Information data model**

### 3.4.1.2 Calculate Quote

Table 17 contains the data model for Calculate Quote.

| Name | Cardinality | Type | Description |
|---|---|---|---|
| **quoteId** | 1 | CorrelationId | Identifies quote message. |
| **transactionId** | 1 | CorrelationId | Identifies transaction message. |
| **transactionRequestId** | 1 | CorrelationId | Identifies transactionRequest message. |
| **payee** | 1 | Party | Information about the Payee in the proposed financial transaction. |
| **payer** | 1 | Party | Information about the Payer in the proposed financial transaction. |
| **amountType** | 1 | AmountType | SEND for sendAmount, RECEIVE for receiveAmount.. |
| **amount** | 1 | Money | Depending on amountType:<br>If SEND: The amount the Payer would like to send, that is, the amount that should be withdrawn from the Payer account including any fees. The amount is updated by each participating entity in the transaction.<br>If RECEIVE: The amount the Payee should receive, that is, the amount that should be sent to the receiver exclusive any fees. The amount is not updated by any of the participating entities. |
| **fees** | 0..1 | Money | The fees in the transaction.<br>• The fees element should be empty if fees should be non-disclosed.<br>• The fees element should be non-empty if fees should be disclosed. |
| **transactionType** | 1 | TransactionType | The type of transaction for which the quote is requested. |
| **geoCode** | 0..1 | GeoCode | Longitude and Latitude of the initiating party. Can be used to detect fraud. |
| **note** | 0..1 | Note | A memo that will be attached to the transaction. |
| **expiration** | 0..1 | DateTime | Expiration is optional. It can be set to get a quick failure in case the peer FSP takes too long to respond. Also useful for notifying Consumer, Agent, Merchant that their request has a time limit. |
| **extensionList** | 0..1 | ExtensionList | Optional extension, specific to deployment. |

**Table 17 – Calculate Quote data model**

### 3.4.2 Responses

This section describes the data model of responses used by the server in the API for services provided by the resource **Quotes**.

#### 3.4.2.1 Return Quote Information

Table 18 contains the data model for **Return Quote Information**.

| Name | Cardinality | Type | Description |
|---|---|---|---|
| **quoteId** | 1 | CorrelationId | Identifies quote message. |
| **transferAmount** | 1 | Money | The amount of money that the Payer FSP should transfer to the Payee FSP. |
| **payerReceiveAmount** | 1 | Money | The amount of money that the Payee should receive in the end-to-end transaction. Optional, as the Payee FSP might not want to disclose any optional Payee fees. |
| **payeeFspFee** | 0..1 | Money | Payee FSP's part of the transaction fee. |
| **payeeFspCommission** | 0..1 | Money | Transaction commission from the Payee FSP. |
| **expiration** | 1 | DateTime | Date and time until when the quotation is valid and can be honored when used in the subsequent transaction. |
| **geoCode** | 0..1 | GeoCode | Longitude and Latitude of the Payee. Can be used to detect fraud. |
| **ilpPacket** | 1 | IlpPacket | The ILP Packet that must be attached to the transfer by the Payer. |
| **condition** | 1 | IlpCondition | The condition that must be attached to the transfer by the payer. |
| **extensionList** | 0..1 | ExtensionList | Optional extension, specific to deployment |

**Table 18 – Return Quote Information data model**

### 3.4.3 Error Responses

This section describes the data model of the error responses used by the server in the API for services provided by the resource **Quotes**.

#### 3.4.3.1 Return Quote Information Error

Table 19 contains the data model for **Return Quote Information Error**.

| Name | Cardinality | Type | Description |
|---|---|---|---|
| **quoteId** | 1 | CorrelationId | Identifies quote message. |
| **errorInformation** | 1 | ErrorInformation | Error code, category description. |

**Table 19 – Return Quote Information Error data model**

## 3.5 API Resource Authorizations

This section describes the data model of services for the resource **Authorizations**.

### 3.5.1 Requests

This section describes the data model of services that can be requested by a client in the API for the resource **Authorizations**.

#### 3.5.1.1 Perform Authorization

Table **20** contains the data model for **Perform Authorization**.

| Name | Cardinality | Type | Description |
|---|---|---|---|
| **authorizationId** | 1 | CorrelationId | Identifies authorization message. |
| **authenticationType** | 0..1 | AuthenticationType | The type of authentication. |
| **retriesLeft** | 0..1 | NrOfRetries | Number of retries left. |
| **amount** | 0..1 | Money | The authorization amount. |
| **currency** | 0..1 | CurrencyCode | The authorization currency. |

**Table** 20 **– Perform Authorization data model**

### 3.5.2 Responses

This section describes the data model of responses used by the server in the API for services provided by the resource **Authorizations**.

#### 3.5.2.1 Return Authorization Result

Table 21 contains the data model for **Return Authorization Result**.

| Name | Cardinality | Type | Description |
|---|---|---|---|
| **authorizationId** | 1 | CorrelationId | Identifies authorization message. |
| **authenticationInfo** | 0..1 | AuthenticationInfo | OTP or QR Code if entered, otherwise empty. |
| **responseType** | 1 | AuthorizationResponse | Enum containing response information, if the customer entered the authentication value, rejected the transaction, or requested a resend of the authentication value. |

**Table 21 – Return Authorization Result data model**

### 3.5.3 Error Responses

This section describes the data model of the error responses used by the server in the API for services provided by the resource **Authorizations**.

#### 3.5.3.1 Return Authorization Error

Table 22 contains the data model for **Return Authorization Error**.

| Name | Cardinality | Type | Description |
|---|---|---|---|
| **authorizationId** | 1 | CorrelationId | Identifies authorization message. |
| **errorInformation** | 1 | ErrorInformation | Error code, category description. |

**Table 22 – Return Authorization Error data model**

## 3.6 API Resource Transfers

This section describes the data model of services for the resource **Transfers**.

### 3.6.1 Requests

This section describes the data model of services that can be requested by a client in the API for the resource **Transfers**.

#### 3.6.1.1 Retrieve Transfer Information

Table 23 contains the data model for **Retrieve Transfer Information**.

| Name | Cardinality | Type | Description |
|---|---|---|---|
| **transferId** | 1 | CorrelationId | The common ID between the FSPs and the optional Switch for the transfer object, determined by the Payer FSP. The ID should be re-used for re-sends of the same transfer. A new ID should be generated for each new transfer. |

**Table 23 – Retrieve Transfer Information data model**

#### 3.6.1.2 Perform Transfer

Table 24 contains the data model for **Perform Transfer**.

| Name | Cardinality | Type | Description |
|---|---|---|---|
| **transferId** | 1 | CorrelationId | The common ID between the FSPs and the optional Switch for the transfer object, determined by the Payer FSP. The ID should be re-used for re-sends of the same transfer. A new ID should be generated for each new transfer. |
| **payeeFsp** | 1 | FspId | Payee FSP in the proposed financial transaction. |
| **payerFsp** | 1 | FspId | Payer FSP in the proposed financial transaction. |
| **amount** | 1 | Money | The transfer amount to be sent. |
| **ilpPacket** | 1 | IlpPacket | The ILP Packet containing the amount delivered to the payee and the ILP Address of the payee and any other end-to-end data. |
| **condition** | 1 | IlpCondition | The condition that must be fulfilled to commit the transfer. |
| **expiration** | 1 | DateTime | Expiration can be set to get a quick failure Expiration of the transfer. The transfer should be rolled back if no fulfilment is delivered before this time. |
| **extensionList** | 0..1 | ExtensionList | Optional extension, specific to deployment. |

**Table 24 – Perform Transfer data model**

### 3.6.2 Responses

This section describes the data model of responses used by the server in the API for services provided by the resource **Transfers**.

#### 3.6.2.1 Return Transfer Information

Table 25 contains the data model for **Return Transfer Information**.

| Name | Cardinality | Type | Description |
|---|---|---|---|
| **transferId** | 1 | CorrelationId | The common ID between the FSPs and the optional Switch for the transfer object, determined by the Payer FSP. The ID should be re-used for re-sends of the same transfer. A new ID should be generated for each new transfer. |
| **fulfilment** | 0..1 | IlpFulfilment | The fulfilment of the condition specified with the transaction. Mandatory if transfer has completed successfully. |
| **completedTimestamp** | 0..1 | DateTime | The time and date when the transaction was completed. |
| **transferState** | 1 | TransferState | The state of the transfer. |
| **extensionList** | 0..1 | ExtensionList | Optional extension, specific to deployment. |

**Table 25 – Return Transfer Information data model**

### 3.6.3 Error Responses

This section describes the data model of error responses used by the server in the API for services provided by the resource **Transfers**.

#### 3.6.3.1 Return Transfer Information Error

Table 26 contains the data model for **Return Transfer Information Error**.

| Name | Cardinality | Type | Description |
|---|---|---|---|
| **transferId** | 1 | CorrelationId | The common ID between the FSPs and the optional Switch for the transfer object, determined by the Payer FSP. The ID should be re-used for re-sends of the same transfer. A new ID should be generated for each new transfer. |
| **errorInformation** | 1 | ErrorInformation | Error code, category description. |

**Table 26 – Return Transfer Information Error data model**

## 3.7 API Resource Transactions

This section describes the data model of services for the resource **Transactions**.

### 3.7.1 Requests

This section describes the data model of services that can be requested by a client in the API for the resource **Transactions**.

#### 3.7.1.1 Retrieve Transaction Information

Table 27 contains the data model for **Retrieve Transaction Information**.

| Name | Cardinality | Type | Description |
|---|---|---|---|
| **transactionId** | 1 | CorrelationId | Identifies transaction message. |

**Table 27 – Retrieve Transaction Information data model**

### 3.7.2 Responses

This section describes the data model of responses used by the server in the API for services provided by the resource **Transactions**.

#### 3.7.2.1 Return Transaction Information

Table 28 contains the data model for **Return Transaction Information**.

| Name | Cardinality | Type | Description |
|---|---|---|---|
| **transactionId** | 1 | CorrelationId | Identifies transaction message. |
| **completedTimestamp** | 0..1 | DateTime | The time and date when the transaction was completed. |
| **transactionState** | 1 | TransactionState | The state of the transaction. |
| **code** | 0..1 | Code | Optional redemption information provided to Payer after transaction has been completed. |
| **extensionList** | 0..1 | ExtensionList | Optional extension, specific to deployment. |

**Table 28 – Return Transaction Information data model**

### 3.7.3 Error Responses

This section describes the data model of the error responses used by the server in the API for services provided by the resource **Transactions**.

#### 3.7.3.1 Return Transaction Information Error

Table 29 contains the data model for **Return Transaction Information Error**.

| Name | Cardinality | Type | Description |
|---|---|---|---|
| **transactionId** | 1 | CorrelationId | Identifies transaction message. |
| **errorInformation** | 1 | ErrorInformation | Error code, category description. |

**Table 29 – Return Transaction Information Error data model**

## 3.8 API Resource Bulk Quotes

This section describes the data model of services for the resource **Bulk Quotes**.

### 3.8.1 Requests

This section describes the data model of services that can be requested by a client in the API for the resource **Bulk Quotes**.

#### 3.8.1.1 Retrieve Bulk Quote Information

Table **30** contains the data model for **Retrieve Bulk Quote Information**.

| Name | Cardinality | Type | Description |
|---|---|---|---|
| **bulkQuoteId** | 1 | CorrelationId | The common ID between the FSPs for the bulk quote object, determined by the Payer FSP. The ID should be re-used for re-sends of the same bulk quote. A new ID should be generated for each new bulk quote. |

**Table** 30 **– Retrieve Bulk Quote data model**

#### 3.8.1.2 Calculate Bulk Quote

Table 31 contains the data model for **Calculate Bulk Quote**.

| Name | Cardinality | Type | Description |
|---|---|---|---|
| **bulkQuoteId** | 1 | CorrelationId | The common ID between the FSPs for the bulk quote object, determined by the Payer FSP. The ID should be re-used for re-sends of the same bulk quote. A new ID should be generated for each new bulk quote. |
| **payer** | 1 | PartyIdInfo | Information about the Payer type, id, sub-type/id, FSP Id in the proposed financial transaction. |
| **GeoCode** | 0..1 | GeoCode | Longitude and Latitude of the initiating Party. Can be used to detect fraud. |
| **expiration** | 0..1 | DateTime | Proposed expiration of the quote. |
| **individualQuotes** | 1..1000 | IndividualQuote | List of IndividualQuote elements. |
| **extensionList** | 0..1 | ExtensionList | Optional extension, specific to deployment. |

**Table 31 – Calculate Bulk Quote data model**

### 3.8.2 Responses

This section describes the data model of responses used by the server in the API for services provided by the resource **Bulk Quotes**.

### 3.8.2.1 Return Bulk Quote Information

Table 32 contains the data model for **Return Bulk Quote Information**.

| Name | Cardinality | Type | Description |
|---|---|---|---|
| **bulkQuoteId** | 1 | CorrelationId | The common ID between the FSPs for the bulk quote object, determined by the Payer FSP. The ID should be re-used for re-sends of the same bulk quote. A new ID should be generated for each new bulk quote. |
| **individualQuoteResults** | 0..1000 | IndividualQuoteResult | Fees for each individual transaction (if any are charged per transaction). |
| **expiration** | 1 | DateTime | Date and time until when the quotation is valid and can be honored when used in the subsequent transaction request. |
| **extensionList** | 0..1 | ExtensionList | Optional extension, specific to deployment. |

**Table 32 – Return Bulk Quote Information data model**

## 3.8.3 Error Responses

This section describes the data model of the error responses used by the server in the API for services provided by the resource **Bulk Quotes**.

### 3.8.3.1 Return Bulk Quote Information Error

Table 33 contains the data model for **Return Bulk Quote Information Error**.

| Name | Cardinality | Type | Description |
|---|---|---|---|
| **bulkQuoteId** | 1 | CorrelationId | The common ID between the FSPs for the bulk quote object, determined by the Payer FSP. The ID should be re-used for re-sends of the same bulk quote. A new ID should be generated for each new bulk quote. |
| **errorInformation** | 1 | ErrorInformation | Error code, category description. |

**Table 33 – Return Bulk Quote Information Error data model**

## 3.9 API Resource Bulk Transfers

This section describes the data model of services for the resource **Bulk Transfers**.

### 3.9.1 Requests

This section describes the data model of services that can be requested by a client in the API for the resource **Bulk Transfers**.

#### 3.9.1.1 Retrieve Bulk Transfer Information

Table **34** contains the data model for **Retrieve Bulk Transfer Information**.

| Name | Cardinality | Type | Description |
|---|---|---|---|
| **bulkTransferId** | 1 | CorrelationId | The common ID between the FSPs and the optional Switch for the bulk transfer object, determined by the Payer FSP. The ID should be re-used for re-sends of the same bulk transfer. A new ID should be generated for each new bulk transfer. |

**Table** 34 **– Retrieve Bulk Transfer Information data model**

#### 3.9.1.2 Perform Bulk Transfer

Table **35** contains the data model for **Perform Bulk Transfer**.

| Name | Cardinality | Type | Description |
|---|---|---|---|
| **bulkTransferId** | 1 | CorrelationId | The common ID between the FSPs and the optional Switch for the bulk transfer object, determined by the Payer FSP. The ID should be re-used for re-sends of the same bulk transfer. A new ID should be generated for each new bulk transfer. |
| **bulkQuoteId** | 1 | CorrelationId | This identifies previous quotation request. The fees specified in the previous quotation will have to be honored in the transfer. |
| **payeeFsp** | 1 | FspId | Payee FSP identifier. |
| **payerFsp** | 1 | FspId | Information about the Payer in the proposed financial transaction. |
| **individualTransfers** | 1..1000 | IndividualTransfer | List of IndividualTransfer elements. |
| **expiration** | 1 | DateTime | Expiration time of the transfers. |
| **extensionList** | 0..1 | ExtensionList | Optional extension, specific to deployment. |

**Table** 35 **– Perform Bulk Transfer data model**

### 3.9.2 Responses

This section describes the data model of responses used by the server in the API for services provided by the resource **Bulk Transfers**.

#### 3.9.2.1 Return Bulk Transfer Information

Table **36** contains the data model for **Return Bulk Transfer Information**.

| Name | Cardinality | Type | Description |
|---|---|---|---|
| **bulkTransferId** | 1 | CorrelationId | The common ID between the FSPs and the optional Switch for the bulk transfer object, determined by the Payer FSP. The ID should be re-used for re-sends of the same bulk transfer. A new ID should be generated for each new bulk transfer. |
| **completedTimestamp** | 0..1 | DateTime | The time and date when the bulk transfer was completed. |
| **individualTransferResults** | 0..1000 | IndividualTransferResult | List of IndividualTransferResult elements. |
| **bulkTransferState** | 1 | BulkTransferState | The state of the bulk transaction. |
| **extensionList** | 0..1 | ExtensionList | Optional extension, specific to deployment. |

**Table** 36 **– Return Bulk Transfer Information data model**

### 3.9.3 Error Responses

This section describes the data model of the error responses used by the server in the API for services provided by the resource **Bulk Transfers**.

#### 3.9.3.1 Return Bulk Transfer Information Error

Table **37** contains the data model for **Return Bulk Transfer Information Error**.

| Name | Cardinality | Type | Description |
|---|---|---|---|
| **bulkTransferId** | 1 | CorrelationId | The common ID between the FSPs and the optional Switch for the bulk transfer object, determined by the Payer FSP. The ID should be re-used for re-sends of the same bulk transfer. A new ID should be generated for each new bulk transfer. |
| **errorInformation** | 1 | ErrorInformation | Error code, category description. |

**Table** 37 **– Return Bulk Transfer Information Error data model**

# 4 API Supporting Data Model

This section defines the data model.

section 4.1 introduces the formats used for the element data types used by the API.

Section 4.2 defines the element data types used by the API.

Section 4.2.16.2 defines the elements types used by the API.

Section 4.3.35 identifies the complex types used by the API.

Section 4.5 identifies the enumerations used by the API.

## 4.1 Length Specification

All element data types have both a minimum and maximum length. These lengths are indicated by one of the following:

- A minimum and maximum length
- An exact length
- A regular expression limiting the element such that only a specific length or lengths can be used.

### 4.1.1 Minimum and Maximum Length

If a minimum and maximum length is used, this will be indicated after the data type in parentheses: First the minimum value (inclusive value), followed by two period characters (..), and then the maximum value (inclusive value).

Examples:

- String(1..32) – A String that is minimum one character and maximum 32 characters long.

### 4.1.2 Integer(3..10) – An UndefinedEnum

The API data type **UndefinedEnum** is a JSON String consisting of one to 32 uppercase characters including an underscore character ( _) .

#### 4.1.2.1 Regular Expression

The regular expression for restricting the **UndefinedEnum** type appears in Listing 1.

```
^[A-Z_]{1,32}$
```

**Listing 1 – Regular expression for data type UndefinedEnum**

### 4.1.3 Name

The API data type **Name** is a JSON String, restricted by a regular expression to avoid characters which are generally not used in a name.

#### 4.1.3.1 Regular Expression

The regular expression for restricting the **Name** type is shown in Listing 2. The restriction will not allow a string consisting of whitespace only, all Unicode characters should be allowed, as well as period (.), apostrophe (“), dash (-), comma (,) and space ( ) characters. The maximum number of characters in the **Name** is 128.

**Note:** In some programming languages, Unicode support needs to be specifically enabled. As an example, if Java is used the flag UNICODE_CHARACTER_CLASS needs to be enabled to allow Unicode characters.

```
^(?!\s*$)[\w .,'-]{1,128}$
```

**Listing 2 – Regular expression for data type Name**

- Integer that is minimum 3 digits, but maximum 10 digits long.

### 4.1.4 Exact Length

If an exact length is used, this will be indicated after the data type in parentheses containing only one exact value. Other lengths are not allowed.

Examples:

- String(3) – A String that is exactly three characters long.

### 4.1.5 Integer(4) – An UndefinedEnum

The API data type **UndefinedEnum** is a JSON String consisting of one to 32 uppercase characters including an underscore character ( _) .

#### 4.1.5.1 Regular Expression

The regular expression for restricting the **UndefinedEnum** type appears in Listing 1.

```
^[A-Z_]{1,32}$
```

**Listing 1 – Regular expression for data type UndefinedEnum**

### 4.1.6 Name

The API data type **Name** is a JSON String, restricted by a regular expression to avoid characters which are generally not used in a name.

#### 4.1.6.1 Regular Expression

The regular expression for restricting the **Name** type is shown in Listing 2. The restriction will not allow a string consisting of whitespace only, all Unicode characters should be allowed, as well as period (.), apostrophe (“), dash (-), comma (,) and space ( ) characters. The maximum number of characters in the **Name** is 128.

**Note:** In some programming languages, Unicode support needs to be specifically enabled. As an example, if Java is used the flag UNICODE_CHARACTER_CLASS needs to be enabled to allow Unicode characters.

```
^(?!\s*$)[\w .,'-]{1,128}$
```

**Listing 2 – Regular expression for data type Name**

- Integer that is exactly four digits long.

### 4.1.7 Regular Expressions

Some element data types are restricted using regular expressions. The regular expressions in this document use the standard for syntax and character classes established by the programming language Perl[1].

[1] https://perldoc.perl.org/perlre.html#Regular-Expressions - perlre - Perl regular expressions

## 4.2 Element Data Type Formats

This section contains the definition of element data types used by the API.

### 4.2.1 String

The API data type **String** is a normal JSON String[2], limited by a minimum and maximum number of characters.

#### 4.2.1.1 Example Format

**String(1..32)** – A String that is minimum one character and maximum 32 characters long.

##### 4.2.1.1.1 Example

An example of **String(1..32)** appears below:

**This String is 28 characters**

#### 4.2.1.2 Example Format

**String(1..128)** – A String that is minimum one character and maximum 128 characters long.

##### 4.2.1.2.1 Example

An example of **String(1..128)** appears below:

**This String is longer than 32 characters, but less than 128**

### 4.2.2 Enum

The API data type **Enum** is a restricted list of allowed JSON String (see Section 4.2.1) values; an enumeration of values. Other values than the ones defined in the list are not allowed.

#### 4.2.2.1 Example Format

**Enum of String(1..32)** – A String that is minimum one character and maximum 32 characters long and restricted by the allowed list of values. The description of the element contains a link to the enumeration.

### 4.2.3 UndefinedEnum

The API data type **UndefinedEnum** is a JSON String consisting of one to 32 uppercase characters including an underscore character ( _) .

#### 4.2.3.1 Regular Expression

The regular expression for restricting the **UndefinedEnum** type appears in Listing 1.

```
^[A-Z_]{1,32}$
```

**Listing 1 – Regular expression for data type UndefinedEnum**

[2] https://tools.ietf.org/html/rfc7159#section-7 – The JavaScript Object Notation (JSON) Data Interchange Format - Strings

### 4.2.4 Name

The API data type **Name** is a JSON String, restricted by a regular expression to avoid characters which are generally not used in a name.

#### 4.2.4.1 Regular Expression

The regular expression for restricting the **Name** type is shown in Listing 2. The restriction will not allow a string consisting of whitespace only, all Unicode[3] characters should be allowed, as well as period (.), apostrophe ("), dash (-), comma (,) and space ( ) characters. The maximum number of characters in the **Name** is 128.

**Note:** In some programming languages, Unicode support needs to be specifically enabled. As an example, if Java is used the flag UNICODE_CHARACTER_CLASS needs to be enabled to allow Unicode characters.

```
^(?!\s*$)[\w .,'-]{1,128}$
```

**Listing 2 – Regular expression for data type Name**

### 4.2.5 Integer

The API data type **Integer** is a JSON String consisting of digits only. Negative numbers and leading zeroes are not allowed. The data type is always limited by a number of digits.

#### 4.2.5.1 Regular Expression

The regular expression for restricting an Integer is shown in Listing 3.

```
^[1-9]\d*$
```

**Listing 3 – Regular expression for data type Integer**

#### 4.2.5.2 Example format

**Integer(1..6)** – An **Integer** that is at minimum one digit long, maximum six digits.

#### 4.2.5.3 Example

An example of **Integer(1..6)** appears below:

**123456**

### 4.2.6 OtpValue

The API data type **OtpValue** is a JSON String of three to 10 characters, consisting of digits only. Negative numbers are not allowed. One or more leading zeros are allowed.

[3] http://www.unicode.org/ - The Unicode Consortium

#### 4.2.6.1 Regular Expression

The regular expression for restricting the **OtpValue** type appears in Listing 4**Error! Reference source not found.**

```
^\d{3,10}$
```

**Listing 4 – Regular expression for data type OtpValue**

### 4.2.7 BopCode

The API data type **BopCode** is a JSON String of three characters, consisting of digits only. Negative numbers are not allowed. A leading zero is not allowed.

#### 4.2.7.1 Regular Expression

The regular expression for restricting the **BopCode** type appears in Listing **5**.

```
^[1-9]\d{2}$
```

**Listing** 5 **– Regular expression for data type BopCode**

### 4.2.8 ErrorCode

The API data type ErrorCode is a JSON String of four characters, consisting of digits only. Negative numbers are not allowed. A leading zero is not allowed.

#### 4.2.8.1 Regular Expression

The regular expression for restricting the **ErrorCode** type appears in Listing **6**.

```
^[1-9]\d{3}$
```

**Listing** 6 **– Regular expression for data type ErrorCode**

### 4.2.9 TokenCode

The API data type **TokenCode** is a JSON String between four and 32 characters, consisting of digits or upper or lowercase characters from **a** to **z**.

#### 4.2.9.1 Regular Expression

The regular expression for restricting the **TokenCode** type appears in Listing 7.

```
^[0-9a-zA-Z]{4,32}$
```

**Listing 7 – Regular expression for data type TokenCode**

### 4.2.10 MerchantClassificationCode

The API data type **MerchantClassificationCode** is a JSON String consisting of one to four digits.

#### 4.2.10.1 Regular Expression

The regular expression for restricting the **MerchantClassificationCode** type appears in Listing 8.

```
^[\d]{1,4}$
```

**Listing 8 - Regular expression for data type MerchantClassificationCode**

### 4.2.11 Latitude

The API data type **Latitude** is a JSON String in a lexical format that is restricted by a regular expression for interoperability reasons.

#### 4.2.11.1 Regular Expression

The regular expression for restricting the **Latitude** type appears in Listing 9.

```
^(\+|-)?(?:90(?:(?:\.0{1,6})?)|(?:[0-9]|[1-8][0-9])(?:(?:\.[0-9]{1,6})?))$
```

**Listing 9 – Regular expression for data type Latitude**

### 4.2.12 Longitude

The API data type **Longitude** is a JSON String in a lexical format that is restricted by a regular expression for interoperability reasons.

#### 4.2.12.1 Regular Expression

The regular expression for restricting the **Longitude** type is shown in Listing 10.

```
^(\+|-)?(?:180(?:(?:\.0{1,6})?)|(?:[0-9]|[1-9][0-9]|1[0-7][0-9])(?:(?:\.[0-
9]{1,6})?))$
```

**Listing 10 – Regular expression for data type Longitude**

### 4.2.13 Amount

The API data type Amount is a JSON String in a canonical format that is restricted by a regular expression for interoperability reasons.

#### 4.2.13.1 Regular Expression

The regular expression for restricting the **Amount** type appears in Listing 11. This pattern:

- Does not allow trailing zeroes.
- Allows an amount without a minor currency unit.
- Allows only four digits in the minor currency unit; a negative value is not allowed.
- Does not allow more than 18 digits in the major currency unit.

The regular expression for restricting the **Longitude** type is shown in Listing 10.

```
^([0]|([1-9][0-9]*))([.][0-9]{0,3}[1-9])?$
```

**Listing 11 – Regular expression for data type Amount**

### 4.2.13.2 Examples

See Table 38 for validation results for some example **Amount** values using the regular expression in Section 4.2.13.1.

| Value | Validation result |
|---|---|
| 5 | Accepted |
| 5.0 | Rejected |
| 5. | Rejected |
| 5.00 | Rejected |
| 5.5 | Accepted |
| 5.50 | Rejected |
| 5.5555 | Accepted |
| 5.55555 | Rejected |
| 555555555555555555 | Accepted |
| 5555555555555555555 | Rejected |
| -5.5 | Rejected |
| 0.5 | Accepted |
| .5 | Rejected |
| 00.5 | Rejected |
| 0 | Accepted |

**Table 38 – Example results for different values for Amount type**

## 4.2.14 DateTime

The API data type **DateTime** is a JSON String in a lexical format that is restricted by a regular expression for interoperability reasons.

### 4.2.14.1 Regular Expression

The regular expression for restricting **DateTime** appears in Listing **12**. The format is according to ISO 8601[4], expressed in a combined date, time and time format. A more readable version of the format is *yyyy-MM-dd'T'HH:mm:ss.SSS[-HH:MM]*

```
^(?:[1-9]\d{3}-(?:(?:0[1-9]|1[0-2])-(?:0[1-9]|1\d|2[0-8])|(?:0[13-9]|1[0-2])-
(?:29|30)|(?:0[13578]|1[02])-31)|(?:[1-
9]\d(?:0[48]|[2468][048]|[13579][26])|(?:[2468][048]|[13579][26])00)-02-
29)T(?:[01]\d|2[0-3]):[0-5]\d:[0-5]\d(?:(\.\d{3}))(?:Z|[+-][01]\d:[0-5]\d)$
```

**Listing** 12 **– Regular expression for data type DateTime**

[4] https://www.iso.org/iso-8601-date-and-time-format.html - Date and time format - ISO 8601

### 4.2.14.2 Examples

Two examples of the **DateTime** type appear below:

**2016-05-24T08:38:08.699-04:00**

**2016-05-24T08:38:08.699Z** (where **Z** indicates Zulu time zone, which is the same as UTC).

## 4.2.15 Date

The API data type **Date** is a JSON String in a lexical format that is restricted by a regular expression for interoperability reasons.

### 4.2.15.1 Regular Expression

The regular expression for restricting the **Date** type appears in Listing 13. This format, as specified in ISO 8601, contains a date only. A more readable version of the format is *yyyy-MM-dd*.

```
^(?:[1-9]\d{3}-(?:(?:0[1-9]|1[0-2])-(?:0[1-9]|1\d|2[0-8])|(?:0[13-9]|1[0-2])-
(?:29|30)|(?:0[13578]|1[02])-31)|(?:[1-
9]\d(?:0[48]|[2468][048]|[13579][26])|(?:[2468][048]|[13579][26])00)-02-29)$
```

**Listing 13 – Regular expression for data type Date**

### 4.2.15.2 Examples

Two examples of the **Date** type appear below:

**1982-05-23**

**1987-08-05**

## 4.2.16 UUID

The API data type UUID (Universally Unique Identifier) is a JSON String in canonical format, conforming to RFC 4122[5], that is restricted by a regular expression for interoperability reasons. A UUID is always 36 characters long, 32 hexadecimal symbols and four dashes (-).

### 4.2.16.1 Regular Expression

The regular expression is shown in Listing 14.

```
^[0-9a-f]{8}-[0-9a-f]{4}-[1-5][0-9a-f]{3}-[89ab][0-9a-f]{3}-[0-9a-f]{12}$
```

**Listing** 14 **– Regular expression for data type UUID**

### 4.2.16.2 Example

An example of a **UUID** type appears below:

**a8323bc6-c228-4df2-ae82-e5a997baf898**

[5] https://tools.ietf.org/html/rfc4122 - A Universally Unique IDentifier (UUID) URN Namespace

### 4.2.17 BinaryString

The API data type BinaryString is a JSON String. The String is the base64url[6] encoding of a string of raw bytes. The length restrictions for the Binary String type apply to the underlying binary data, indicating the allowed number of bytes. For data types with a fixed size no padding character is used.

#### 4.2.17.1 Regular Expression

The regular expression for restricting the **BinaryString** type appears in Listing 15.

```
^[A-Za-z0-9-_]+[=]?$
```

**Listing 15 – Regular expression for data type BinaryString**

#### 4.2.17.2 Example Format

**BinaryString(32)** –32 bytes of data base64url encoded.

##### 4.2.17.2.1 Example

An example of a **BinaryString(32..256)** appears below. Note that a padding character ('**=**') has been added to ensure that the string is a multiple of four characters.

**QmlsbCAmIE1lbGluZGEgR2F0ZXMgRm91bmRhdGlvbiE=**

### 4.2.18 BinaryString32

The API data type **BinaryString32** is a fixed size variation of the API data type BinaryString in Section 4.2.17, in which the raw underlying data is always of 32 bytes. The data type **BinaryString32** should not use a padding character because the size of the underlying data is fixed.

#### 4.2.18.1 Regular Expression

The regular expression for restricting the **BinaryString32** type appears in Listing 16.

```
^[A-Za-z0-9-_]{43}$
```

**Listing 16 – Regular expression for data type BinaryString32**

#### 4.2.18.2 Example Format

**BinaryString(32)** –32 bytes of data base64url encoded.

##### 4.2.18.2.1 Example

An example of a **BinaryString32** appears below. Note that this is the same binary data as the example in Section 4.2.17.2.1, but due to the underlying data being fixed size, the padding character '**=**' is excluded.

**QmlsbCAmIE1lbGluZGEgR2F0ZXMgRm91bmRhdGlvbiE**

[6] https://tools.ietf.org/html/rfc4648#section-5 - The Base16, Base32, and Base64 Data Encodings - Base 64 Encoding with URL and Filename Safe Alphabet

## 4.3 Element Definitions

This section contains the definition of the elements types that are used by the API.

### 4.3.1 AmountType

Table 39 contains the data model for the element **AmountType**.

| Name | Cardinality | Type | Description |
|---|---|---|---|
| AmountType | 1 | **Error! Reference source not found.** of String(1..32) | This element contains the amount type. See Section 4.5.1 (AmountType) for more information on allowed values. |

**Table 39 – Element AmountType**

### 4.3.2 AuthenticationType

Table 40 contains the data model for the element **AuthenticationType**.

| Name | Cardinality | Format | Description |
|---|---|---|---|
| Authentication | 1 | **Error! Reference source not found.** of String(1..32) | This element contains the authentication type. See Section 4.5.2 (AuthenticationType) for possible enumeration values. |

**Table 40 – Element AuthenticationType**

### 4.3.3 AuthenticationValue

Table 41 contains the data model for the element **AuthenticationValue**.

| Name | Cardinality | Format | Description |
|---|---|---|---|
| AuthenticationValue | 1 | Depending on AuthenticationType.<br>If OTP: Example<br>An example of **Integer(1..6)** appears below:<br>**123456**<br>OtpValue<br>If QRCODE: String(1..64) | This element contains the authentication value. The format depends on the authentication type used in the AuthenticationInfo complex type. |

**Table 41 – Element AuthenticationValue**

### 4.3.4 AuthorizationResponse

Table **42** contains the data model for the element **AuthorizationResponse**.

| Name | Cardinality | Type | Description |
|---|---|---|---|
| **AuthorizationResponse** | 1 | **Error! Reference source not found.** of String(1..32) | This element contains the authorization response. See Section 4.5.3 (AuthorizationResponse) for possible enumeration values. |

**Table** 42 **– Element AuthorizationResponse**

### 4.3.5 BalanceOfPayments

Table 43 contains the data model for the element **BalanceOfPayment**.

| Name | Cardinality | Type | Description |
|---|---|---|---|
| BalanceOfPayments | 1 | BopCode | The possible values and meaning are defined in https://www.imf.org/external/np/sta/bopcode/ |

**Table 43 – Element BalanceOfPayments**

### 4.3.6 BulkTransferState

Table 44 contains the data model for the element **BulkTransferState**.

<table>
<tr><th>Name</th><th>Cardinality</th><th>Type</th><th>Description</th></tr>
<tr><td>BulkTransferState</td><td>1</td><td><b>Error! Reference source not found.</b> of String(1..32)</td><td>See Section 4.5.3 (AuthorizationResponse<br>Table 95 contains the allowed values for the enumeration <b>AuthorizationResponse</b>.
<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td>ENTERED</td><td>Consumer entered the authentication value.</td></tr>
<tr><td>REJECTED</td><td>Consumer rejected the transaction.</td></tr>
<tr><td>RESEND</td><td>Consumer requested to resend the authentication value.</td></tr>
</table>
<b>Table 95 – Enumeration AuthorizationResponse</b><br>) for more information on allowed values.</td></tr>
</table>

**Table 44 – Element BulkTransferState**

### 4.3.7 Code

Table 45 contains the data model for the element **Code**.

| Name | Cardinality | Type | Description |
|---|---|---|---|
| Code | 1 | TokenCode | Any code/token returned by the Payee FSP. |

**Table 45 – Element Code**

### 4.3.8 CorrelationId

Table 46 contains the data model for the element **CorrelationId**.

| Name | Cardinality | Type | Description |
|---|---|---|---|
| **CorrelationId** | 1 | Examples<br>Two examples of the **Date** type appear below:<br>**1982-05-23**<br>**1987-08-05**<br>UUID | Identifier that correlates all messages of the same sequence. |

**Table 46 – Element CorrelationId**

### 4.3.9 Currency

Table 47 contains the data model for the element **Currency**.

<table>
<tr><th>Name</th><th>Cardinality</th><th>Type</th><th>Description</th></tr>
<tr><td><b>Currency</b></td><td>1</td><td><b>Error! Reference source not found.</b> of String(3)</td><td>See Section 4.5.4 (BulkTransferState<br>Table 96 contains the allowed values for the enumeration <b>BulkTransferState</b>.
<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td><b>RECEIVED</b></td><td>Payee FSP has received the bulk transfer from the Payer FSP.</td></tr>
<tr><td><b>PENDING</b></td><td>Payee FSP has validated the bulk transfer.</td></tr>
<tr><td><b>ACCEPTED</b></td><td>Payee FSP has accepted to process the bulk transfer.</td></tr>
<tr><td><b>PROCESSING</b></td><td>Payee FSP has started to transfer fund to the Payees.</td></tr>
<tr><td><b>COMPLETED</b></td><td>Payee FSP has completed transfer of funds to the Payees.</td></tr>
<tr><td><b>REJECTED</b></td><td>Payee FSP has rejected to process the transfer.</td></tr>
</table>
<b>Table 96 – Enumeration BulkTransferState</b><br>CurrencyCode) for more information on allowed values.</td></tr>
</table>

**Table 47 – Element Currency**

### 4.3.10 DateOfBirth

Table 48 contains the data model for the element **DateOfBirth**.

| Name | Cardinality | Type | Description |
|---|---|---|---|
| DateOfBirth | 1 | Examples<br>Two examples of the **DateTime** type appear below:<br>**2016-05-24T08:38:08.699-04:00**<br>**2016-05-24T08:38:08.699Z** (where **Z** indicates Zulu time zone, which is the same as UTC).<br>Date | Date of Birth of the Party. |

**Table 48 – Element DateOfBirth**

### 4.3.11 ErrorCode

Table 49 contains the data model for the element **ErrorCode**.

| Name | Cardinality | Type | Description |
|---|---|---|---|
| ErrorCode | 1 | ErrorCode | Four digit error code, see Section 4.6 for more information. |

**Table 49 – Element ErrorCode**

### 4.3.12 ErrorDescription

Table 50 contains the data model for the element **ErrorDescription**.

| Name | Cardinality | Type | Description |
|---|---|---|---|
| ErrorDescription | 1 | String(1..128) | Error description string. |

**Table 50 – Element ErrorDescription**

### 4.3.13 ExtensionKey

Table 51 contains the data model for the element **ExtensionKey**.

| Name | Cardinality | Type | Description |
|---|---|---|---|
| ExtensionKey | 1 | String(1..32) | The extension key. |

**Table 51 – Element ExtensionKey**

### 4.3.14 ExtensionValue

Table 52 contains the data model for the element **ExtensionValue**.

| Name | Cardinality | Type | Description |
|---|---|---|---|
| ExtensionValue | 1 | String(1..128) | The extension value. |

**Table 52 – Element ExtensionValue**

### 4.3.15 FirstName

Table 53 contains the data model for the element **FirstName**.

| Name | Cardinality | Type | Description |
|---|---|---|---|
| FirstName | 1 | Name | First name of the Party |

**Table 53 – Element FirstName**

### 4.3.16 FspId

Table 54 contains the data model for the element **FspId**.

| Name | Cardinality | Type | Description |
|---|---|---|---|
| FspId | 1 | String(1..32) | The FSP identifier. |

**Table 54 – Element FspId**

### 4.3.17 IlpCondition

Table 55 contains the data model for the element **IlpCondition**.

| Name | Cardinality | Type | Description |
|---|---|---|---|
| IlpCondition | 1 | BinaryString32 | The condition that must be attached to the transfer by the Payer. |

**Table 55 – Element IlpCondition**

### 4.3.18 IlpFulfilment

Table 56 contains the data model for the element **IlpFulfilment**.

| Name | Cardinality | Type | Description |
|---|---|---|---|
| IlpFulfilment | 1 | BinaryString32 | The fulfilment that must be attached to the transfer by the Payee. |

**Table 56 – Element IlpFulfilment**

### 4.3.19 IlpPacket

Table 57 contains the data model for the element **IlpPacket**.

| Name | Cardinality | Type | Description |
|---|---|---|---|
| IlpPacket | 1 | Example<br>An example of a **UUID** type appears below:<br>**a8323bc6-c228-4df2-ae82-e5a997baf898**<br>BinaryString(1..32768) | Information for recipient (transport layer information). |

**Table 57 – Element IlpPacket**

### 4.3.20 LastName

Table 58 contains the data model for the element **LastName**.

| Name | Cardinality | Type | Description |
|---|---|---|---|
| LastName | 1 | Name | Last name of the Party (ISO 20022 definition). |

**Table 58 – Element LastName**

### 4.3.21 MerchantClassificationCode

Table 59 contains the data model for the element **MechantClassificationCode**.

| Name | Cardinality | Type | Description |
|---|---|---|---|
| MerchantClassificationCode | 1 | MerchantClassificationCode | A limited set of pre-defined numbers. This list would be a limited set of numbers identifying a set of popular merchant types like School Fees, Pubs and Restaurants, Groceries, and so on. |

**Table 59 – Element MerchantClassificationCode**

### 4.3.22 MiddleName

Table 60 contains the data model for the element **MiddleName**.

| Name | Cardinality | Type | Description |
|---|---|---|---|
| MiddleName | 1 | Name | Middle name of the Party (ISO 20022 definition). |

**Table 60 – Element MiddleName**

### 4.3.23 Note

Table 61 contains the data model for the element **Note**.

| Name | Cardinality | Type | Description |
|---|---|---|---|
| Note | 1 | String(1..128) | Memo assigned to transaction. |

**Table 61 – Element Note**

### 4.3.24 NrOfRetries

Table 62 contains the data model for the element **NrOfRetries**.

| Name | Cardinality | Type | Description |
|---|---|---|---|
| NrOfRetries | 1 | Integer(1..2) | Number of retries. |

**Table 62 – Element NrOfRetries**

### 4.3.25 PartyIdentifier

Table 63 contains the data model for the element **PartyIdentifier**.

| Name | Cardinality | Type | Description |
|---|---|---|---|
| **PartyIdentifier** | 1 | String(1..128) | Identifier of the Party. |

**Table 63 – Element PartyIdentifier**

### 4.3.26 PartyIdType

Table 64 contains the data model for the element **PartyIdType**.

| Name | Cardinality | Type | Description |
|---|---|---|---|
| **PartyIdType** | 1 | **Error! Reference source not found.** of String(1..32) | See Section 4.5.6 (PartyIdType) for more information on allowed values. |

**Table 64 – Element PartyIdType**

### 4.3.27 PartyName

Table 65 contains the data model for the element **PartyName**.

| Name | Cardinality | Type | Description |
|---|---|---|---|
| **PartyName** | 1 | Name | Name of the Party. Could be a real name or a nickname. |

**Table 65 – Element PartyName**

### 4.3.28 PartySubIdOrType

Table 66 contains the data model for the element **PartySubIdOrType**.

| Name | Cardinality | Type | Description |
|---|---|---|---|
| **PartySubIdOrType** | 1 | String(1..128) | Either a sub-identifier of a PartyIdentifier, or a sub-type of the PartyIdType, normally a PersonalIdentifierType. |

**Table 66 – Element PartySubIdOrType**

### 4.3.29 RefundReason

Table 67 contains the data model for the element **RefundReason**.

| Name | Cardinality | Type | Description |
|---|---|---|---|
| **RefundReason** | 1 | String(1..128) | Reason for the refund. |

**Table 67 – Element RefundReason**

### 4.3.30 TransactionInitiator

Table 68 contains the data model for the element **TransactionInitiator**.

| Name | Cardinality | Type | Description |
|---|---|---|---|
| TransactionInitiator | 1 | **Error! Reference source not found.** of String(1..32) | See Section 4.5.8 (TransactionInitiator) for more information on allowed values. |

**Table 68 – Element TransactionInitiator**

### 4.3.31 TransactionInitiatorType

Table 69 contains the data model for the element **TransactionInitiatorType**.

| Name | Cardinality | Type | Description |
|---|---|---|---|
| TransactionInitiatorType | 1 | **Error! Reference source not found.** of String(1..32) | See Section 4.5.9 (TransactionInitiatorType) for more information on allowed values. |

**Table 69 – Element TransactionInitiatorType**

### 4.3.32 TransactionRequestState

Table 70 contains the data model for the element **TransactionRequestState**.

<table>
<tr><th>Name</th><th>Cardinality</th><th>Type</th><th>Description</th></tr>
<tr><td><b>TransactionRequestState</b></td><td>1</td><td><b>Error! Reference source not found.</b> of String(1..32)</td><td>See Section 4.5.10 (TransactionInitiator<br>Table 99 contains the allowed values for the enumeration <b>TransactionInitiator</b>.
<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td><b>PAYER</b></td><td>The sender of funds is initiating the transaction. The account to send from is either owned by the Payer or is connected to the Payer in some way.</td></tr>
<tr><td><b>PAYEE</b></td><td>The recipient of the funds is also initiating the transaction by sending a transaction request. The Payer must approve the transaction, either automatically by a pre-generated OTP, pre-approval of the Payee, or by manually approving in his or her own Device.</td></tr>
</table>
<b>Table 99 – Enumeration TransactionInitiator</b><br>
<b>4.3.33 TransactionInitiatorType</b><br>
Table 100 contains the allowed values for the enumeration <b>TransactionInitiatorType</b>.
<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td><b>CONSUMER</b></td><td>Consumer is the initiator of the transaction.</td></tr>
<tr><td><b>AGENT</b></td><td>Agent is the initiator of the transaction.</td></tr>
<tr><td><b>BUSINESS</b></td><td>Business is the initiator of the transaction.</td></tr>
<tr><td><b>DEVICE</b></td><td>Device is the initiator of the transaction.</td></tr>
</table>
<b>Table 100 – Enumeration TransactionInitiatorType</b><br>
TransactionRequestState) for more information on allowed values.</td></tr>
</table>

**Table 70 – Element TransactionRequestState**

### 4.3.34 TransactionScenario

Table 71 contains the data model for the element **TransactionScenario**.

| Name | Cardinality | Type | Description |
|---|---|---|---|
| **TransactionScenario** | 1 | **Error! Reference source not found.** of String(1..32) | See Section 4.5.11 (TransactionScenario) for more information on allowed values. |

**Table 71 – Element TransactionScenario**

### 4.3.35 TransactionState

Table 72 contains the data model for the element **TransactionState**.

| Name | Cardinality | Type | Description |
|---|---|---|---|
| TransactionState | 1 | **Error! Reference source not found.** of String(1..32) | See Section 4.5.12 (TransactionState) for more information on allowed values. |

**Table 72 – Element TransactionState**

### 4.3.36 TransferState

Table 73 contains the data model for the element **TransferState**.

<table>
<tr><th>Name</th><th>Cardinality</th><th>Type</th><th>Description</th></tr>
<tr><td>TransferState</td><td>1</td><td><b>Error! Reference source not found.</b> of String(1..32)</td><td>See Section 4.5.12 (TransactionState<br>Table 103 contains the allowed values for the enumeration <b>TransactionState</b>.
<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td>RECEIVED</td><td>Payee FSP has received the transaction from the Payer FSP.</td></tr>
<tr><td>PENDING</td><td>Payee FSP has validated the transaction.</td></tr>
<tr><td>COMPLETED</td><td>Payee FSP has successfully performed the transaction.</td></tr>
<tr><td>REJECTED</td><td>Payee FSP has failed to perform the transaction.</td></tr>
</table>
<b>Table 103 – Enumeration TransactionState</b><br>TransferState) for more information on allowed values.</td></tr>
</table>

**Table 73 – Element TransferState**

### 4.3.37 TransactionSubScenario

Table 74 contains the data model for the element **TransactionSubScenario**.

| Name | Cardinality | Type | Description |
|---|---|---|---|
| TransactionSubScenario | 1 | UndefinedEnum | Possible sub-scenario, defined locally within the scheme. |

**Table 74 – Element TransactionSubScenario**

## 4.4 Complex Types

This section contains the complex types that are used by the API.

### 4.4.1 AuthenticationInfo

Table 75 contains the data model for the complex type **AuthenticationInfo**.

| Name | Cardinality | Type | Description |
|---|---|---|---|
| authentication | 1 | AuthenticationType | The type of authentication. |
| authenticationValue | 1 | AuthenticationValue | The authentication value. |

**Table 75 – Complex type AuthenticationInfo**

### 4.4.2 ErrorInformation

Table 76 contains the data model for the complex type **ErrorInformation**.

| Name | Cardinality | Type | Description |
|---|---|---|---|
| errorCode | 1 | ErrorCode | Specific error number. |
| errorDescription | 1 | ErrorDescription | Error description string. |
| extensionList | 0..1 | ExtensionList | Optional list of extensions, specific to deployment. |

**Table 76 – Complex type ErrorInformation**

### 4.4.3 Extension

Table 77 contains the data model for the complex type **Extension**.

| Name | Cardinality | Type | Description |
|---|---|---|---|
| key | 1 | ExtensionKey | The extension key. |
| value | 1 | ExtensionValue | The extension value. |

**Table 77 – Complex type Extension**

### 4.4.4 ExtensionList

Table 78 contains the data model for the complex type **ExtensionList**.

| Name | Cardinality | Type | Description |
|---|---|---|---|
| extension | 1..16 | Extension | A number of Extension elements. |

**Table 78 – Complex type ExtensionList**

### 4.4.5 IndividualQuote

Table 79 contains the data model for the complex type **IndividualQuote**.

| Name | Cardinality | Type | Description |
|---|---|---|---|
| quoteId | 1 | CorrelationId | Identifies quote message. |
| transactionId | 1 | CorrelationId | Identifies transaction message. |

| | | | |
|---|---|---|---|
| **payee** | 1 | Party | Information about the Payee in the proposed financial transaction. |
| **amountType** | 1 | AmountType | SEND_AMOUNT for sendAmount, RECEIVE_AMOUNT for receiveAmount. |
| **amount** | 1 | Money | Depending on amountType:<br>If SEND: The amount the Payer would like to send, that is, the amount that should be withdrawn from the Payer account including any fees. The amount is updated by each participating entity in the transaction.<br>If RECEIVE: The amount the Payee should receive, that is, the amount that should be sent to the receiver exclusive any fees. The amount is not updated by any of the participating entities. |
| **fees** | 0..1 | Money | The fees in the transaction.<br>• The fees element should be empty if fees should be non-disclosed.<br>• The fees element should be non-empty if fees should be disclosed. |
| **transactionType** | 1 | TransactionType | The type of transaction that the quote is requested for. |
| **note** | 0..1 | Note | A memo that will be attached to the transaction. This is sent in the quote so it can be included in the ILP Packet. |
| **extensionList** | 0..1 | ExtensionList | Optional extension, specific to deployment. |

**Table 79 – Complex type IndividualQuote**

### 4.4.6 IndividualQuoteResult

Table 80 contains the data model for the complex type **IndividualQuoteResult**.

| Name | Cardinality | Type | Description |
|---|---|---|---|
| **quoteId** | 1 | CorrelationId | Identifies the quote message. |
| **payeeReceiveAmount** | 0..1 | Money | Amount that the Payee should receive in the end-to-end transaction. Optional as the Payee FSP might not want to disclose any optional Payee fees. |
| **payeeFspFee** | 0..1 | Money | Payee FSP's part of the transaction fee. |
| **payeeFspCommission** | 0..1 | Money | Transaction commission from the Payee FSP. |
| **ilpPacket** | 0..1 | IlpPacket | The ILP Packet that must be attached to the transfer by the payer. |
| **condition** | 0..1 | IlpCondition | The condition that must be attached to the transfer by the payer. |

| | | | |
|---|---|---|---|
| **errorInformation** | 0..1 | ErrorInformation | Error code, category description.<br>**Note**: If errorInformation is set, the following are not set:<br>• receiveAmount<br>• payeeFspFee<br>• payeeFspCommission<br>• ilpPacket<br>• condtion |
| **extensionList** | 0..1 | ExtensionList | Optional extension, specific to deployment |

**Table 80 – Complex type IndividualQuoteResult**

### 4.4.7 IndividualTransfer

Table 81 contains the data model for the complex type **IndividualTransfer**.

| Name | Cardinality | Type | Description |
|---|---|---|---|
| **transferId** | 1 | CorrelationId | Identifies messages related to the same /transfers sequence. |
| **transferAmount** | 1 | Money | The transaction amount to be sent. |
| **ilpPacket** | 1 | IlpPacket | The ILP Packet containing the amount delivered to the payee and the ILP Address of the payee and any other end-to-end data. |
| **condition** | 1 | IlpCondition | The condition that must be fulfilled to commit the transfer. |
| **extensionList** | 0..1 | ExtensionList | Optional extension, specific to deployment. |

**Table 81 – Complex type IndividualTransfer**

### 4.4.8 IndividualTransferResult

Table 82 contains the data model for the complex type **IndividualTransferResult**.

| Name | Cardinality | Type | Description |
|---|---|---|---|
| **transferId** | 1 | CorrelationId | Identifies messages related to the same /transfers sequence. |
| **fulfilment** | 0..1 | IlpFulfilmentMoney | The fulfilment of the condition specified with the transaction.<br>**Note**: Either fulfilment is set or errorInformation is set, not both. |
| **errorInformation** | 0..1 | ErrorInformation | If transactionState is REJECTED, error information may be provided.<br>**Note**: Either fulfilment is set or errorInformation is set, not both. |
| **extensionList** | 0..1 | ExtensionList | Optional extension, specific to deployment. |

**Table 82 – Complex type IndividualTransferResult**

### 4.4.9 GeoCode

Table 83 contains the data model for the complex type **GeoCode**.

| Name | Cardinality | Type | Description |
|---|---|---|---|
| **latitude** | 1 | Latitude | The Latitude of the service initiating Party. |
| **longitude** | 1 | Longitude | The Longitude of the service initiating Party. |

**Table 83 – Complex type GeoCode**

### 4.4.10 Money

Table 84 contains the data model for the complex type **Money**.

| Name | Cardinality | Type | Description |
|---|---|---|---|
| **currency** | 1 | Currency | The currency of the Amount. |
| **amount** | 1 | Amount | The amount of Money. |

**Table 84 – Complex type Money**

### 4.4.11 Party

Table 85 contains the data model for the complex type **Party**.

| Name | Cardinality | Type | Description |
|---|---|---|---|
| **partyIdInfo** | 1 | PartyIdInfo | Party Id type, id, sub ID or type, and FSP Id. |
| **merchantClassificationCode** | 0..1 | MerchantClassificationCode | Optional: Used in the context of Payee Information, where the Payee happens to be a merchant accepting merchant payments. |
| **name** | 0..1 | PartyName | The 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. |

**Table 85 – Complex type Party**

### 4.4.12 PartyComplexName

Table 86 contains the data model for the complex type **PartyComplextype**.

| Name | Cardinality | Type | Description |
|---|---|---|---|
| **firstName** | 0..1 | FirstName | First name. |
| **middleName** | 0..1 | MiddleName | Middle name. |
| **lastName** | 0..1 | LastName | Last name. |

**Table 86 – Complex type PartyComplexName**

### 4.4.13 PartyIdInfo

Table 87 – Complex type PartyIdInfo contains the data model for the complex type **PartyIdInfo**.

| Name | Cardinality | Type | Description |
|---|---|---|---|
| **partyIdType** | 1 | PartyIdType | The type of the identifier. |
| **partyIdentifier** | 1 | PartyIdentifier | An identifier for the Party. |
| **partySubIdOrType** | 0..1 | PartySubIdOrType | A sub-identifier or sub-type for the Party. |
| **fspId** | 0..1 | FspId | The FSP ID (if known) |

**Table 87 – Complex type PartyIdInfo**

### 4.4.14 PartyPersonalInfo

Table 88 contains the data model for the complex type **PartyPersonalInfo**.

| Name | Cardinality | Type | Description |
|---|---|---|---|
| **complexName** | 0..1 | PartyComplexName | First, middle and last name. |
| **dateOfBirth** | 0..1 | DateOfBirth | Date of birth. |

**Table 88 – Complex type PartyPersonalInfo**

### 4.4.15 PartyResult

Table 89 contains the data model for the complex type **PartyResult**.

| Name | Cardinality | Type | Description |
|---|---|---|---|
| **partyId** | 1 | PartyIdInfo | Party Id type, id, sub ID or type, and FSP Id. |
| **errorInformation** | 0..1 | ErrorInformation | If the Party failed to be added, error information should be provided. Otherwise, this parameter should be empty to indicate success. |

**Table 89 – Complex type PartyPersonalInfo**

### 4.4.16 Refund

Table 90 contains the data model for the complex type **Refund**.

| Name | Cardinality | Type | Description |
|---|---|---|---|
| **originalTransactionId** | 1 | CorrelationId | Reference to the original transaction ID that is requested to be refunded. |
| **refundReason** | 0..1 | RefundReason | Free text indicating the reason for the refund. |

**Table 90 – Complex type Refund**

### 4.4.17 Transaction

Table 91 contains the data model for the complex type **Transaction**.

| Name | Cardinality | Type | Description |
|---|---|---|---|
| **transactionId** | 1 | CorrelationId | ID of the transaction, the ID is determined by the Payer FSP during the creation of the quote.. |
| **quoteId** | 1 | CorrelationId | ID of the quote, the ID is determined by the Payer FSP during the creation of the quote. |
| **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 | The transaction amount to be sent. |
| **transactionType** | 1 | TransactionType | The 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. |

**Table 91 – Complex type Transaction**

### 4.4.18 TransactionType

Table 92 contains the data model for the complex type **TransactionType**.

| Name | Cardinality | Type | Description |
|---|---|---|---|
| **scenario** | 1 | TransactionScenario | Deposit, withdrawal, refund, … |
| **subScenario** | 0..1 | TransactionSubScenario | Possible sub-scenario, defined locally within the scheme. |
| **initiator** | 1 | TransactionInitiator | Who is initiating the transaction: payer or payee |
| **initiatorType** | 1 | TransactionInitiatorType | Consumer, agent, business, … |
| **refundInfo** | 0..1 | Refund | Extra information specific to a refund scenario. Should only be populated if scenario is REFUND. |
| **balanceOfPayments** | 0..1 | BalanceOfPayments | Balance of Payments code. |

**Table 92 – Complex type TransactionType**

## 4.5 Enumerations

This section contains the enumerations used by the API.

### 4.5.1 AmountType

Table 93 contains the allowed values for the enumeration **AmountType**.

| Name | Description |
|---|---|
| SEND | The amount the Payer would like to send, that is, the amount that should be withdrawn from the Payer account including any fees. |
| RECEIVE | The amount the Payer would like the Payee to receive, that is, the amount that should be sent to the receiver exclusive fees. |

**Table 93 – Enumeration AmountType**

### 4.5.2 AuthenticationType

Table 94 contains the allowed values for the enumeration **AuthenticationType**.

| Name | Description |
|---|---|
| OTP | One-time password, either chosen by the Party, or generated by the Party's FSP. |
| QRCODE | QR code used as One Time Password. |

**Table 94 – Enumeration AuthenticationType**

### 4.5.3 AuthorizationResponse

Table 95 contains the allowed values for the enumeration **AuthorizationResponse**.

| Name | Description |
|---|---|
| ENTERED | Consumer entered the authentication value. |
| REJECTED | Consumer rejected the transaction. |
| RESEND | Consumer requested to resend the authentication value. |

**Table 95 – Enumeration AuthorizationResponse**

### 4.5.4 BulkTransferState

Table 96 contains the allowed values for the enumeration **BulkTransferState**.

| Name | Description |
|---|---|
| RECEIVED | Payee FSP has received the bulk transfer from the Payer FSP. |
| PENDING | Payee FSP has validated the bulk transfer. |
| ACCEPTED | Payee FSP has accepted to process the bulk transfer. |
| PROCESSING | Payee FSP has started to transfer fund to the Payees. |
| COMPLETED | Payee FSP has completed transfer of funds to the Payees. |
| REJECTED | Payee FSP has rejected to process the transfer. |

**Table 96 – Enumeration BulkTransferState**

### 4.5.5 CurrencyCode

The currency codes defined in ISO 4217[7] as three-letter alphabetic codes are used as the standard naming representation for currencies. The currency codes from ISO 4217 are not is shown in this document, implementers are instead encouraged to use the information provided by the ISO 4217 standard directly.

### 4.5.6 PartyIdType

Table 97 contains the allowed values for the enumeration **PartyIdType**.

| Name | Description |
|---|---|
| MSISDN | Used when an MSISDN (Mobile Station International Subscriber Directory Number, that is, the phone number) is used as reference to a participant. The MSISDN identifier should be in international format according to the ITU-T E.164 standard. Optionally, the MSISDN may be prefixed by a single plus sign, indicating the international prefix. |
| EMAIL | Used when an email should be used as reference to a participant. The format of the email should be according to the informational RFC 3696. |
| PERSONAL_ID | Used when some kind of personal identifier should be used as reference to a participant. Examples of personal identification can be passport number, birth certificate number, national registration number or similar. The identifier number shall be added in the PartyIdentifier element. The personal identifier type shall be added in the PartySubIdOrType element. |
| BUSINESS | Used when a specific Business (for example, an Organization or a Company) should be used as 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 | Used when a specific Device (for example, a POS or ATM) ID connected to a specific Business or Organization should be used as reference to a Party. To use a specific device under a specific Business or Organization, the PartySubIdOrType element should be used. |
| ACCOUNT_ID | Used when a bank account number or FSP account ID should be used as reference to a participant. The *ACCOUNT_ID* identifier can be in any format, as formats can greatly differ depending on country and FSP. |

[7] https://www.iso.org/iso-4217-currency-codes.html - Currency codes - ISO 4217

| | |
|---|---|
| **IBAN** | Used when a bank account number or FSP account ID should be used as reference to a participant. The *IBAN* identifier can consist of up to 34 alphanumeric characters and should be entered without any whitespace. |
| **ALIAS** | Used when an alias should be used as 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 can be 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. |

**Table 97 – Enumeration PartyIdType**

### 4.5.7 PersonalIdentifierType

Table 98 contains the allowed values for the enumeration **PersonalIdentifierType**.

| Name | Description |
|---|---|
| **PASSPORT** | Used when a passport number should be used as reference to a Party |
| **NATIONAL_REGISTRATION** | Used when a national registration number should be used as reference to a Party |
| **DRIVING_LICENSE** | Used when a driving license should be used as reference to a Party |
| **ALIEN_REGISTRATION** | Used when an alien registration number should be used as reference to a Party |
| **NATIONAL_ID_CARD** | Used when a national ID card number should be used as reference to a Party |
| **EMPLOYER_ID** | Used when a tax identification number should be used as reference to a Party |
| **TAXI_ID_NUMBER** | Used when a tax identification number should be used as reference to a Party. |
| **SENIOR_CITIZENS_CARD** | Used when a senior citizens card number should be used as reference to a Party |
| **MARRIAGE_CERTIFICATE** | Used when a marriage certificate number should be used as reference to a Party |
| **HEALTH_CARD** | Used when a health card number should be used as reference to a Party |
| **VOTERS_ID** | Used when a voters identification number should be used as reference to a Party |
| **UNITED_NATIONS** | Used when an UN (United Nations) number should be used as reference to a Party |
| **OTHER_ID** | Used when any other type of identification type number should be used as reference to a Party |

**Table 98 – Enumeration PersonalIdentifierType**

### 4.5.8 TransactionInitiator

Table 99 contains the allowed values for the enumeration **TransactionInitiator**.

| Name | Description |
|---|---|
| **PAYER** | The sender of funds is initiating the transaction. The account to send from is either owned by the Payer or is connected to the Payer in some way. |
| **PAYEE** | The recipient of the funds is also initiating the transaction by sending a transaction request. The Payer must approve the transaction, either automatically by a pre-generated OTP, pre-approval of the Payee, or by manually approving in his or her own Device. |

**Table 99 – Enumeration TransactionInitiator**

### 4.5.9 TransactionInitiatorType

Table 100 contains the allowed values for the enumeration **TransactionInitiatorType**.

| Name | Description |
|---|---|
| CONSUMER | Consumer is the initiator of the transaction. |
| AGENT | Agent is the initiator of the transaction. |
| BUSINESS | Business is the initiator of the transaction. |
| DEVICE | Device is the initiator of the transaction. |

**Table 100 – Enumeration TransactionInitiatorType**

### 4.5.10 TransactionRequestState

Table 101 contains the allowed values for the enumeration **TransactionRequestState**.

| Name | Description |
|---|---|
| RECEIVED | Payer FSP has received the transaction from the Payee FSP. |
| PENDING | Payer FSP has sent the transaction request to the Payer. |
| ACCEPTED | Payer has approved the transaction. |
| REJECTED | Payer has rejected the transaction. |

**Table 101 – Enumeration TransactionRequestState**

### 4.5.11 TransactionScenario

Table **102** contains the allowed values for the enumeration **TransactionScenario**.

| Name | Description |
|---|---|
| DEPOSIT | Used for performing a Cash-In (deposit) transaction, where in the normal case electronic funds are transferred from a Business account to a Consumer account, and physical cash is given from the Consumer to the Business User. |
| WITHDRAWAL | Used for performing a Cash-Out (withdrawal) transaction, where in the normal case electronic funds are transferred from a Consumer's account to a Business account, and physical cash is given from the Business User to the Consumer. |
| TRANSFER | Used for performing a P2P (Peer to Peer, or Consumer to Consumer) transaction. |
| PAYMENT | Typically used for performing a transaction from a Consumer to a Merchant/Organization, but could also be for a B2B (Business to Business) payment. The transaction could be online to purchase in an Internet store, in a physical store where both the Consumer and Business User are present, a bill payment, a donation, and so on. |
| REFUND | Used for performing a refund of transaction. |

**Table** 102 **– Enumeration TransactionScenario**

### 4.5.12 TransactionState

Table 103 contains the allowed values for the enumeration **TransactionState**.

| Name | Description |
|---|---|
| RECEIVED | Payee FSP has received the transaction from the Payer FSP. |
| PENDING | Payee FSP has validated the transaction. |
| COMPLETED | Payee FSP has successfully performed the transaction. |
| REJECTED | Payee FSP has failed to perform the transaction. |

**Table 103 – Enumeration TransactionState**

### 4.5.13 TransferState

Table 104 contains the allowed values for the enumeration **TransferState**.

| Name | Description |
|---|---|
| RECEIVED | The next ledger has received the transfer. |
| RESERVED | The next ledger has reserved the transfer. |
| COMMITTED | The next ledger has successfully performed the transfer. |
| ABORTED | The next ledger has aborted the transfer due a rejection or failure to perform the transfer. |

**Table 104 – Enumeration TransferState**

## 4.6 Error Codes

Each error code in the API is a four-digit number, such as **1234**, where the first number (**1** in the example) represents the high-level error category, the second number (**2** in the example) represents the low-level error category, and the last two numbers (**34** in the example) represents the specific error. Figure 1 shows the structure of an error code. The following sub-sections contains information and defined error codes per high-level error category.

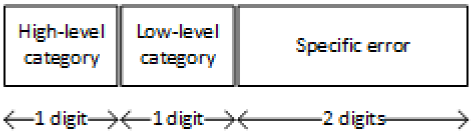

**Figure 1 – Error code structure**

Each defined high- and low-level category combination contains a generic error (*x***0***xx*), which can be used if there is no specific error, or if the server would not like to return information which is considered private.

All specific errors below *xx***40**, that is, *xx***00** to *xx***39**, are reserved for future use by the API. All specific errors above and including *xx40*, can be used for scheme-specific errors. If a client receives an unknown scheme-specific error, the unknown scheme-specific error should be interpreted as a generic error for the high- and low-level category combination instead (*xx***00**).

### 4.6.1 Communication Errors – 1xxx

All possible communication or network errors that could arise that cannot be represented by an HTTP status code should use the high-level error code 1 (error codes **1***xxx*). Because all services in the API are asynchronous, these error codes should generally be used by a Switch in the Callback to the client FSP if the Peer FSP could not be reached, or if a callback is not received from the Peer FSP within an agreed timeout.

Low level categories defined under Communication Errors:

- **Generic Communication Error** – **10***xx*

See Table 105 for all communication errors defined in the API.

| Error Code | Name | Description | /participants | /parties | /transactionRequests | /quotes | /authorizations | /transfers | /transactions | /bulkQuotes | /bulkTransfers |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **1000** | Communication error | Generic communication error. | X | X | X | X | X | X | X | X | X |
| **1001** | Destination communication error | The Destination of the request failed to be reached. This usually indicates that a Peer FSP failed to respond from an intermediate entity. | X | X | X | X | X | X | X | X | X |

**Table 105 – Communication errors – 1xxx**

### 4.6.2 Server Errors – 2xxx

All errors occurring in the server in which the server failed to fulfil an apparently valid request from the client should use the high-level error code 2 (error codes **2***xxx*). These error codes should indicate that the server is aware that it has encountered an error or is otherwise incapable of performing the requested service.

Low-level categories defined under server Errors:

- **Generic server Error** – **20***xx*

See Table 106 for all server errors defined in the API.

| Error Code | Name | Description | /participants | /parties | /transactionRequests | /quotes | /authorizations | /transfers | /transactions | /bulkQuotes | /bulkTransfers |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **2000** | Generic server error | Generic server error to be used for not disclosing information that may be considered private. | X | X | X | X | X | X | X | X | X |
| **2001** | Internal server error | A generic unexpected exception. This usually indicates a bug or unhandled error case. | X | X | X | X | X | X | X | X | X |
| **2002** | Not implemented | Service requested is not supported by the server. | X | X | X | X | X | X | X | X | X |
| **2003** | Service currently unavailable | Service requested is currently unavailable on the server. This could be because maintenance is taking place, or because of a temporary failure. | X | X | X | X | X | X | X | X | X |
| **2004** | Server timed out | A time out has occurred, meaning the next Party in the chain did not send a callback in time. This could be because a timeout is set too low or because something took longer than it should. | X | X | X | X | X | X | X | X | X |
| **2005** | Server busy | Server is rejecting requests due to overloading. Try again later. | X | X | X | X | X | X | X | X | X |

**Table 106 – Server errors – 2xxx**

## Client Errors – 3xxx

All possible errors occurring in the server where the server's opinion is that the client have sent one or more erroneous parameters should use the high-level error code 3 (error codes **3***xxx*). These error codes should indicate that the server could not perform the service according to the request from the client. The server should provide an explanation why the service could not be performed.

Low level categories defined under client Errors:

- **Generic Client Error** – **30***xx*
  - See Table 107 for the generic client errors defined in the API.
- **Validation Error** – **31***xx*
  - See Table 108 for the validation errors defined in the API.
- **Identifier Error** – **32***xx*
  - See Table 109 for the identifier errors defined in the API.
- **Expired Error** – **33***xx*
  - See Table 110 for the expired errors defined in the API.

| Error Code | Name | Description | /participants | /parties | /transactionRequests | /quotes | /authorizations | /transfers | /transactions | /bulkQuotes | /bulkTransfers |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **3000** | Generic client error | Generic client error, to be used for not disclosing information that may be considered private. | X | X | X | X | X | X | X | X | X |
| **3001** | Unacceptable version requested | Client requested to use a protocol version which is not supported by the server. | X | X | X | X | X | X | X | X | X |
| **3002** | Unknown URI | The provided URI was unknown by the server. | | | | | | | | | |
| **3003** | Add Party information error | An error occurred while adding or updating information regarding a Party. | X | | | | | | | | |

**Table 107 – Generic client errors – 30xx**

| Error Code | Name | Description | /participants | /parties | /transactionRequests | /quotes | /authorizations | /transfers | /transactions | /bulkQuotes | /bulkTransfers |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **3100** | Generic validation error | Generic validation error to be used for not disclosing information that may be considered private. | X | X | X | X | X | X | X | X | X |
| **3101** | Malformed syntax | The format of the parameter is not valid. For example, amount set to **5.ABC**. The error description field should specify which element is erroneous. | X | X | X | X | X | X | X | X | X |
| **3102** | Missing mandatory element | A mandatory element in the data model was missing. | X | X | X | X | X | X | X | X | X |
| **3103** | Too many elements | The number of elements of an array exceeds the maximum number allowed. | X | X | X | X | X | X | X | X | X |
| **3104** | Too large payload | The size of the payload exceeds the maximum size. | X | X | X | X | X | X | X | X | X |
| **3105** | Invalid signature | Some parameters have changed in the message, making the signature invalid. This may indicate that the message may have been modified maliciously. | X | X | X | X | X | X | X | X | X |
| **3106** | Modified request | A request with the same ID has previously been processed where the parameters are not the same. | | | X | X | | X | | X | X |
| **3107** | Missing mandatory extension parameter | A scheme mandatory extension parameter was missing. | | | X | X | X | X | X | X | X |

**Table 108 – Validation Errors – 31xx**

| Error Code | Name | Description | /participants | /parties | /transactionRequests | /quotes | /authorizations | /transfers | /transactions | /bulkQuotes | /bulkTransfers |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **3200** | Generic ID not found | Generic ID error provided by the client. | X | X | X | X | X | X | X | X | X |
| **3201** | Destination FSP Error | The Destination FSP does not exist or cannot be found. | X | X | X | X | X | X | X | X | X |
| **3202** | Payer FSP ID not found | The provided Payer FSP ID not found. | | | | | | X | | X | X |
| **3203** | Payee FSP ID not found | The provided Payer FSP ID not found. | | | | | | X | | X | X |
| **3204** | Party not found | The Party with the provided identifier, identifier type, and optional sub id or type was not found. | X | X | X | X | | | | | |
| **3205** | Quote ID not found | The provided Quote ID was not found in the server. | | | | X | | X | | | |
| **3206** | Transaction request ID not found | The provided Transaction Request ID was not found in the server. | | | X | | | X | | | |
| **3207** | Transaction ID not found | The provided Transaction ID was not found in the server. | | | | | | | X | | |
| **3208** | Transfer ID not found | The provided Transfer ID was not found in the server. | | | | | | X | | | |
| **3209** | Bulk quote ID not found | The provided Bulk Quote ID was not found in the server. | | | | | | | | X | X |
| **3210** | Bulk transfer ID not found | The provided Bulk Transfer ID was not found in the server. | | | | | | | | | X |

**Table 109 – Identifier Errors – 32xx**

| Error Code | Name | Description | /participants | /parties | /transactionRequests | /quotes | /authorizations | /transfers | /transactions | /bulkQuotes | /bulkTransfers |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **3300** | Generic expired error | Generic expired object error, to be used for not disclosing information that may be considered private. | X | X | X | X | X | X | X | X | X |
| **3301** | Transaction request expired | Client requested to use a transaction request that has already expired. | | | | X | | | | | |
| **3302** | Quote expired | Client requested to use a quote that has already expired. | | | | | X | X | | | X |
| **3303** | Transfer expired | Client requested to use a transfer that has already expired. | X | X | X | X | X | X | X | X | X |

**Table 110 – Expired Errors – 33xx**

### 4.6.3 Payer Errors – 4xxx

All possible errors occurring in the server when the Payer or the Payer FSP is the cause of an error should use the high-level error code 4 (error codes **4***xxx*). These error codes should indicate that there was no error in the server or in the request from the client, but the request failed for some reason due to the Payer or the Payer FSP. The server should provide an explanation why the service could not be performed.

Low level categories defined under Payer Errors:

- **Generic Payer Error** – **40***xx*
- **Payer Rejection Error** – **41***xx*
- **Payer Limit Error** – **42***xx*
- **Payer Permission Error** – **43***xx*
- **Payer Blocked Error** – **44***xx*

See Table 111 for all Payer errors defined in the API.

| Error Code | Name | Description | /participants | /parties | /transactionRequests | /quotes | /authorizations | /transfers | /transactions | /bulkQuotes | /bulkTransfers |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **4000** | Generic Payer error | Generic error due to the Payer or Payer FSP, to be used for not disclosing information that may be considered private. | | | X | X | X | X | X | X | X |
| **4001** | Payer FSP insufficient liquidity | The Payer FSP has insufficient liquidity to perform the transfer. | | | | | | X | | | |
| **4100** | Generic Payer rejection | Payer or Payer FSP rejected the request. | | | X | X | X | X | X | X | X |
| **4101** | Payer rejected transaction request | Payer rejected the transaction request from the Payee. | | | X | | | | | | |
| **4102** | Payer FSP unsupported transaction type | The Payer FSP does not support or rejected the requested transaction type | | | X | | | | | | |
| **4103** | Payer unsupported currency | Payer doesn't have an account which supports the requested currency. | | | X | | | | | | |
| **4200** | Payer limit error | Generic limit error, for example, the Payer is making more payments per day or per month than they are allowed to, or is making a payment that is larger than the allowed maximum per transaction. | | | X | X | | X | | X | X |
| **4300** | Payer permission Error | Generic permission error, the Payer or Payer FSP doesn't have the access rights to perform the service. | | | X | X | X | X | X | X | X |
| **4400** | Generic Payer blocked error | Generic Payer blocked error, the Payer is blocked or has failed regulatory screenings. | | | X | X | X | X | X | X | X |

**Table 111 – Payer errors – 4xxx**

### 4.6.4 Payee Errors – 5xxx

All possible errors occurring in the server when the Payee or the Payee FSP is the cause of an error should use the high-level error code 5 (error codes **5***xxx*). These error codes should indicate that there was no error in the server or in the request from the client, but the request failed for some reason due to the Payee or the Payee FSP. The server should provide an explanation for why the service could not be performed.

Low level categories defined under Payee Errors:

- **Generic Payee Error** – **50***xx*
- **Payee Rejection Error** – **51***xx*
- **Payee Limit Error** – **52***xx*
- **Payee Permission Error** – **53***xx*
- **Payee Blocked Error** – **54***xx*

See Table 112 for all Payee errors defined in the API.

| Error Code | Name | Description | /participants | /parties | /transactionRequests | /quotes | /authorizations | /transfers | /transactions | /bulkQuotes | /bulkTransfers |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **5000** | Generic Payee error | Generic error due to the Payer or Payer FSP, to be used for not disclosing information that may be considered private. | | | X | X | X | X | X | X | X |
| **5001** | Payee FSP insufficient liquidity | The Payee FSP has insufficient liquidity to perform the transfer. | | | | | | X | | | |
| **5100** | Generic Payee rejection | Payee or Payee FSP rejected the request. | | | X | X | X | X | X | X | X |
| **5101** | Payee rejected quote | Payee does not want to proceed with the financial transaction after receiving a quote. | | | | X | | | | X | |
| **5102** | Payee FSP unsupported transaction type | The Payee FSP either does not support or rejected the requested transaction type. | | | | X | | | | X | |
| **5103** | Payee FSP rejected quote | Payee doesn't want to proceed with the financial transaction after receiving a quote. | | | | X | | | | X | |
| **5104** | Payee rejected transaction | Payee rejected the financial transaction. | | | | | | X | | | X |
| **5105** | Payee FSP rejected transaction | Payee FSP rejected the financial transaction. | | | | | | X | | | X |
| **5106** | Payee unsupported currency | Payee doesn't have an account which supports the requested currency. | | | | X | | X | | X | X |
| **5200** | Payee limit error | Generic limit error, for example, the Payee is receiving more payments per day or per month than they are allowed to, or is receiving a payment which is larger than the allowed maximum per transaction. | | | X | X | | X | | X | X |
| **5300** | Payee permission error | Generic permission error; the Payee or Payee FSP does not have the access rights to perform the service. | | | X | X | X | X | X | X | X |
| **5400** | Generic Payee blocked error | Generic Payee Blocked error, the Payee is blocked or has failed regulatory screenings. | | | X | X | X | X | X | X | X |

**Table 112 – Payee errors – 5xxx**