Mobile Money API Specification 1.2.0
P2P Transfers

Document Summary

Official Document Number, Document Title and Version Number	Mobile Money API Specification 1.2.0 – P2P Transfers
Official Document Type	Non-binding Permanent Reference Document
Change Request Security Classification	Non-confidential

Document History

Document Version	API Release Version	Date	Brief Description of Change	Editor / Company
0.1	1.2.0-beta	Aug 2020	• Initial draft of document	GSMA
0.2	1.2.0	Sep 2020	• Implemented RFC 2020-3, RFC 2020-10, RFC 2020-13, RFC 2020-16, RFC 2020-17, RFC 2020-18	GSMA

Other Information

Type	Description
Document Owner	Mobile Money API Working Group
Editor / Company	GSMA

Table of Contents

1 Introduction

The purpose of this document is to specify the endpoints, fields, objects, and enumerations for Person to Person (P2P) Transfer Mobile Money APIs, which are a subset of the GSMA Mobile Money API Specification. The P2P Transfer Mobile Money APIs allow financial service providers (FSPs) to transfer funds from an account holding individual to another account holding individual or to a non-account holding individual (known as an unregistered customer). The API supports a wide number of financial service providers including mobile money providers, banks, and micro-finance institutions. The API supports 'on-us' P2P transfers (both accounts held within one FSP) and 'off-us' P2P transfers (accounts held in different FSPs). Bilateral and Switch-based transfers can be supported.

For further reading, please refer to the following documents:

- **Mobile Money API Introduction**. Introduces the use and benefits of the Mobile Money API. Also provides a glossary of terms used by the Mobile Money API specifications.
- **Mobile Money API Fundamentals**. Specifies the design principles, behaviours, and error handling of the Mobile Money API.
- **Mobile Money API Master Specification**. Documents all Mobile Money API endpoints, fields, objects, and enumerations.

All documentation can be found on the GSMA Mobile Money API Developer Portal.

This document contains the following sections:

1.1 Intended Audience

Audience	Usage	Role
FSP - Mobile Money Providers	• To understand how to implement the Mobile Money API to process P2P transfer requests from Payment Initiation Service Providers.	API Provider
	• To understand how to implement the Mobile Money API to receive P2P transfer requests from other FSPs.	API Provider
	• To understand how to implement the Mobile Money API to send P2P transfer requests to other FSPs.	API Consumer

FSP - Banks / Microfinance Institutions	• To understand how to implement the Mobile Money API to receive P2P transfers from Mobile Money Providers.	API Provider
	• To understand how to implement the Mobile Money API to send P2P transfers to Mobile Money Providers.	API Consumer

2 API Endpoints

API endpoint fields are described in this specification as follows:

- The field **name**.
- The field **type**.
- **Description** of the field.
- **Optionality** of the field, i.e. whether the field must be supplied. Optionality is identified as per follows:

  → Request optionality

  ← Response optionality

  O Field is optional

  M Field is mandatory

  C Field is conditional

  NA Field does not need to be supplied. If supplied, it will be ignored.

- **Reference** where the field is a an array and/or is defined by another object.
- **Validation** applied to the field, including enumeration, field length and use of regular expressions to validate format.

Please note that string fields have a default maximum length of 256 characters unless specified otherwise.

2.1 Transactions API

P2P Transfers can be created, viewed and updated using transactions APIs.

The following paths are permitted:

Operation	Path	Description
Create	*POST /transactions/type/{transactiontype}*	To be used for transaction creation when the provider's API Gateway requires that the transaction *type* be identified in the URL.
View	*GET /transactions/{transactionReference}*	To view a transaction.
Update	*PATCH /transactions/{transactionReference}*	To update the *transactionStatus* of a transaction.

2.1.1 Transaction UML Class Diagram

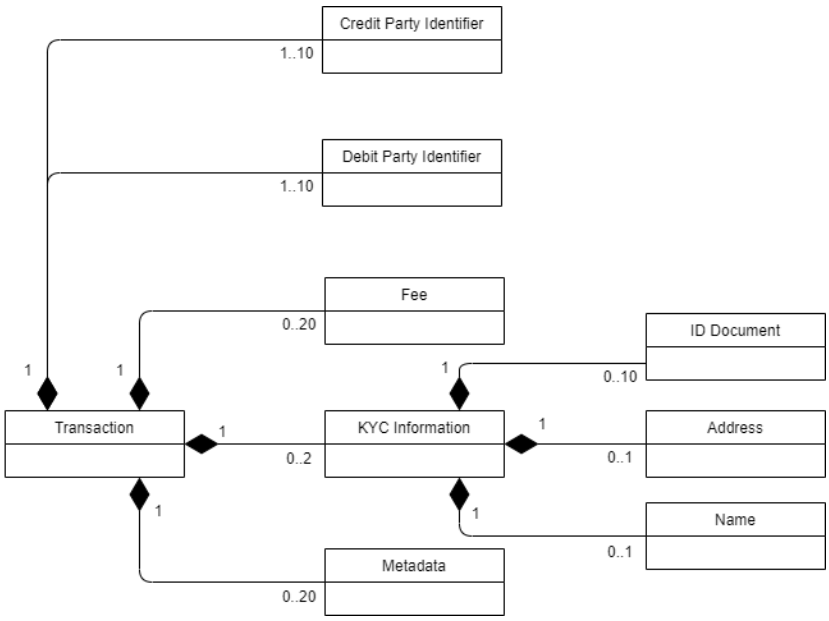

Figure 2-1 Transaction UML Class Diagram

2.1.2 Transaction Object Definition

Transaction Object					
Name	**Type**	**Description**		**Reference**	**Validation**
transactionReference	string	Unique reference for the transaction. This is returned in the response by API provider.	→NA ←M		

requestingOrganisationTransactionReference	string	A reference provided by the requesting organisation that is to be associated with the transaction.	→O ←O		
originalTransactionReference	string	For reversals and refunds, this field indicates the transaction which is the subject of the reversal.	→O ←O		
creditParty	array	A series of key/value pairs that enable the credit party to be identified. Keys include MSISDN and Wallet Identifier.	→C ←C	Account Identifiers	creditParty must be supplied if debitParty is omitted. If debitParty is supplied, then creditParty is optional.
debitParty	array	A collection of key/value pairs that enable the debit party to be identified. Keys include MSISDN and Wallet Identifier.	→C ←C	Account Identifiers	debitParty must be supplied if creditParty is omitted. If creditParty is supplied, then debitParty is optional.
type	string	The harmonised Transaction Type (not required if passed in the URL).	→M ←M		Enumeration = Transaction Types
subType	string	A non-harmonised sub-classification of the type of transaction. Values are not fixed, and usage will vary according to Provider.	→O ←O		
transactionStatus	string	Indicates the status of the transaction as stored by the API provider.	→NA ←M		
amount	string	The transaction amount.	→M ←M		Please refer to API Fundamentals document for amount validation rules.
currency	string	Currency of the transaction amount.	→M ←M		Enumeration = ISO Currency Codes
descriptionText	string	Free format text description of the transaction provided	→O ←O		

		by the client. This can be provided as a reference for the receiver on a notification SMS and on an account statement.			
fees	array	Allows the passing and/or returning of all fees pertaining to the transaction.	→O ←O	Fees Object	
geoCode	string	Indicates the geographic location from where the transaction was initiated.	→O ←O		
oneTimeCode	string	A one-time code that can be supplied in the request or can be generated in the response depending upon the use case. An authorisation code can be supplied in this field for requests that have been pre-authorised.	→O ←O		
recipientKyc	object	A collection of fields detailing the KYC of the transaction recipient.	→O ←O	KYC Information	
senderKyc	object	A collection of fields detailing the KYC of the transaction sender.	→O ←O	KYC Information	
requestingOrganisation	object	The originating organisation of the request.	→O ←O	Requesting Organisation Object	
servicingIdentity	string	The field is used to identify the servicing identity for transactions, e.g. till, POS ID, assistant ID.	→O ←O		
transactionReceipt	string	Transaction receipt number as notified to the parties. This may differ from the Transaction Reference.	→NA ←O		
creationDate	date-time	Date and time when the transaction was created by the API Provider.	→NA ←O		

modificationDate	date-time	Date and time when the transaction was modified by the API Provider.	→NA ←O		
requestDate	date-time	The date and time of the transaction request as supplied by the client.	→O ←O		
customData	string	A collection of key/value pairs that can be used for provider specific fields.	→O ←O	Custom Data Object	
metadata	array	A collection of key/value pairs. These can be used to populate additional properties that describe administrative information regarding the transaction.	→O ←O	Metadata	

2.2 Reversals API

The Reversals API is used to reverse a P2P transfer. The originating transaction reference must be provided in the path to identify the payment to be reversed. For a partial reversal, the amount needs to be supplied.

The supported path is *POST /transactions/{originalTransactionReference}/reversals*.

Note that for performing a reversal without the original transaction reference, please use the Transactions API.

For viewing or updating reversals, the Transactions API should be used.

2.2.1 Reversal UML Class Diagram

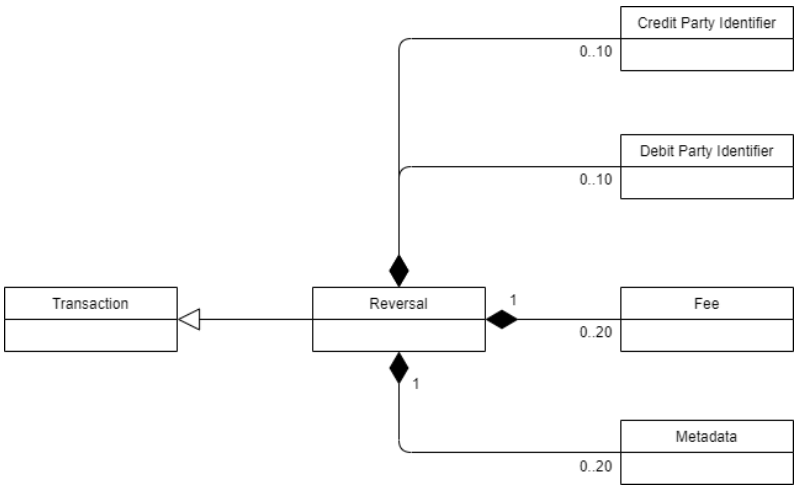

Figure 2-2 Reversal UML Class Diagram

2.2.2 Reversal Object Definition

Reversal Object					
Name	**Type**	**Description**		**Reference**	**Validation**
transaction Reference	string	Unique reference for the transaction. This is returned in the response by API provider.	→NA ←M		
requesting Organisatio nTransactio nReference	string	A reference provided by the requesting organisation that is to be associated with the transaction.	→O ←O		
originalTran sactionRefe rence	string	For reversals and refunds, this field indicates the transaction which is the subject of the reversal.	→NA ←M		
creditParty	array	A series of key/value pairs that enable the credit party to be identified. Keys include MSISDN and Wallet Identifier.	→O ←O	Account Identifiers	
debitParty	array	A collection of key/value pairs that enable the debit party to be identified. Keys include MSISDN and Wallet Identifier.	→O ←O	Account Identifiers	
type	string	The harmonised Transaction Type	→M ←M		Enumeration = Transaction Types Note that only Reversals and Refunds (adjustments) are supported.
subType	string	A non-harmonised sub-classification of the type of transaction. Values are not fixed, and usage will vary according to Provider.	→O ←O		
transaction Status	string	Indicates the status of the transaction as stored by the API provider.	→NA ←M		

amount	string	The transaction Amount.	→O ←O		Please refer to API Fundamentals document for amount validation rules.
currency	string	Currency of the transaction amount.	→O ←O		Enumeration = ISO Currency Codes.
description Text	string	Free format text description of the transaction provided by the client. This can be provided as a reference for the receiver on a notification SMS and on an account statement.	→O ←O		
fees	array	Allows the passing and/or returning of all fees pertaining to the transaction.	→O ←O	Fees Object	
geoCode	string	Indicates the geographic location from where the transaction was initiated.	→O ←O		
requesting Organisation	object	The originating organisation of the request.	→O ←O	Requesting Organisation Object	
servicingIdentity	string	The field is used to identify the servicing identity for transactions, e.g. till, POS ID, assistant ID.	→O ←O		
transaction Receipt	string	Transaction receipt number as notified to the parties. This may differ from the Transaction Reference.	→NA ←O		
creationDate	date-time	Date and time when the transaction was created by the API Provider.	→NA ←O		
modificationDate	date-time	Date and time when the transaction was modified by the API Provider.	→NA ←O		
requestDate	date-time	The date and time of the transaction request as supplied by the client.	→O ←O		

customData	string	A collection of key/value pairs that can be used for provider specific fields.	→O ←O	Custom Data Object	
metadata	array	A collection of key/value pairs. These can be used to populate additional properties that describe administrative information regarding the transaction.	→O ←O	Metadata	

2.3 Quotations API

The Quotations APIs are used to obtain one or multiple quotes for a P2P transfer. The creation of a quote involves returning any fees that will be levied on the sender. A request is made for a quotation by the requesting FSP. The quotation is calculated and returned by the receiving FSP or an intermediate provider such as a Switch. If the sender is satisfied with the quotation, then the sending FSP can confirm and proceed with the P2P transfer using the /transactions API.

The following paths are permitted:

- **Creation** of a quotation: *POST /quotations*
- **View** a quotation: *GET /quotations/{Quotation Reference}*

2.3.1 Quotation UML Class Diagram

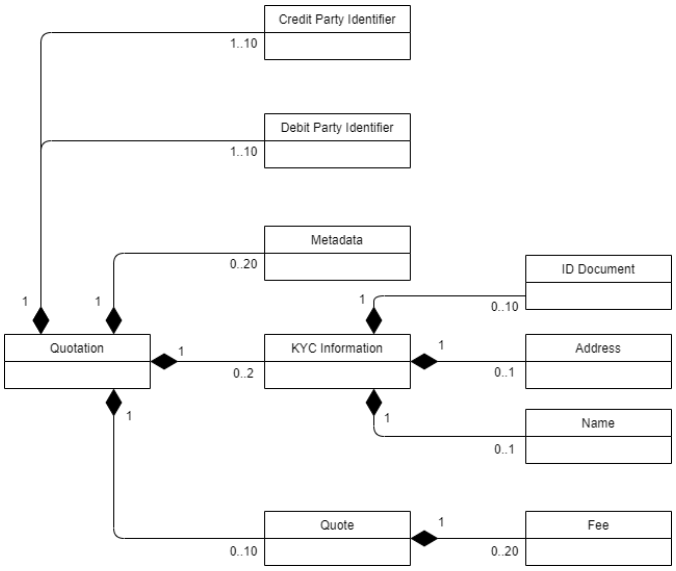

Figure 2-3 Quotation UML Class Diagram

2.3.2 Quotation Object Definition

Quotation Object Definition					
Name	**Type**	**Description**		**Reference**	**Validation**
quotationReference	string	Unique reference for the quotation as provided by the API Provider.	→NA ←M		
creditParty	array	A series of key/value pairs that enable the credit party to be identified. Keys include MSISDN and Wallet Identifier.	→M ←M	Account Identifier	
debitParty	array	A collection of key/value pairs that enable the debit party to be identified. Keys include MSISDN and Wallet Identifier.	→M ←M	Account Identifier	
type	string	The transaction type that the quotation has been requested for.	→O ←O		Enumeration = Transaction Types
subtype	string	The transaction sub-type that the quotation has been requested for.	→O ←O		
quotationStatus	string	Indicates the creation state of the quotation.	→NA ←O		Enumeration = pending, rejected, completed
requestAmount	string	Requested Quotation amount.	→M ←M		Please refer to API Fundamentals document for amount validation rules.
requestCurrency	string	Currency of the requested quotation amount.	→M ←M		Enumeration = ISO Currency Codes
availableDeliveryMethod	string	Delivery Method that is possible for the intended recipient.	→NA ←O		Enumeration = Delivery Method
chosenDeliveryMethod	string	The delivery method chosen by the sending end user as the specific delivery method to be used in the quotes received.	→O ←O		Enumeration = Delivery Method
originCountry	string	The originating country of the quotation request, i.e. the country where the request was initiated.	→O ←O		

receivingCountry	String	Destination country of the quotation request, i.e. the country that the sender wishes to send to.	→O ←O		
quotes	array	A collection of quotes. A quote can be received from a single receiving financial service provider or from multiple providers.	→NA ←O	Quotes	
recipientKyc	object	A collection of fields detailing the KYC of the transaction Recipient, typically used for International Transfers.	→O ←O	KYC Information	
senderKyc	object	A collection of fields detailing the KYC of the transaction Sender, typically used for International Transfers.	→O ←O	KYC Information	
recipientBlocking Reason	string	The reason for blocking the quotation, based on AML checks on the recipient.	→NA ←O		
senderBlockingReason	string	The reason for blocking the quotation, based on AML checks on the sender.	→NA ←O		
requestingOrganisation	object	The originating organisation of the request.	→O ←O	Requesting Organisation Object	
sendingServiceProviderCountry	string	The country of the sending service provider that holds the account of the sender.	→O ←O		
creationDate	date-time	Date and time when the quotation was created by the API Provider.	→NA ←O		
modificationDate	date-time	Date and time when the quotation was modified by the API Provider.	→NA ←O		
requestDate	date-time	The date and time of the quotation request as supplied by the client.	→O ←O		
customData	string	A collection of key/value pairs that can be used for provider specific fields.	→O ←O	Custom Data Object	
metadata	array	A collection of key/value pairs. These can be used to populate	→O ←O	Metadata	

		additional properties that describe administrative information regarding the quotation.			

2.4 Accounts APIs

Using account APIs, an FSP can:

- View transfers for their account with another FSP.
- View their account balance with another FSP.
- Return the name of the intended recipient of a transfer.

2.4.1 Identifying an FSP Account

Two methods are provided for identifying an FSP account, the single identifier method, and the multiple identifiers method.

2.4.1.1 Single Identifier Method

In the scenario where one identifier suffices to uniquely identify an account, the following path is to be used: */accounts/{identifierType}/{identifier}*.

2.4.1.2 Multiple Identifiers Method

Where a single identifier is not sufficient to identify an account, the following path is to be used:
/accounts/{accountIdentifier1}@{value1}${accountIdentifier2}@{value2}${accountIdentifier3}@{value3}.

The path uses a '$' delimiter to separate each identifier, up to a limit of three account identifiers. Each key/value is delimited by '@'.

The list of permitted account identifiers supported by the Mobile Money API can be found in the Account Identifiers section.

2.5 Account Transactions API

An FSP should use this API to return a list of transfers against their account. One of the following paths can be used:

GET /accounts/{identifierType}/{identifier}/transactions – single identifier method

or *GET /accounts/{Account Identifiers}/transactions* – multiple identifiers method

To filter the number of records returned, the following query strings can be used:

Parameter	Type	Format	Description
limit	integer	N/A	Supports pagination. If this is not supplied, then the server will apply a limit of 50 records returned for each request.
offset	integer	N/A	Supports pagination. This value will indicate the cursor position from where to retrieve the set of records. For

			example, a limit of 50 and offset of 10 will return records 11 to 60.
fromDateTime	string	date-time	Indicates the minimum creationDate for which records should be returned.
toDateTime	string	date-time	Indicates the maximum creationDate for which records should be returned.
transactionStatus	string	N/A	Indicates the status of the transactions to be returned.
transactionType	string	N/A	Indicates the type of the transactions to be returned.

Note 1: For a harmonised behavior, API Providers should make sure that the transactions are returned in descending date created order.

Note 2: HTTP response headers are returned with each response indicating the total number of records available (X-Records-Available-Count) and total number of records returned (X-Records-Returned-Count).

2.5.1 Account Transaction UML Class Diagram

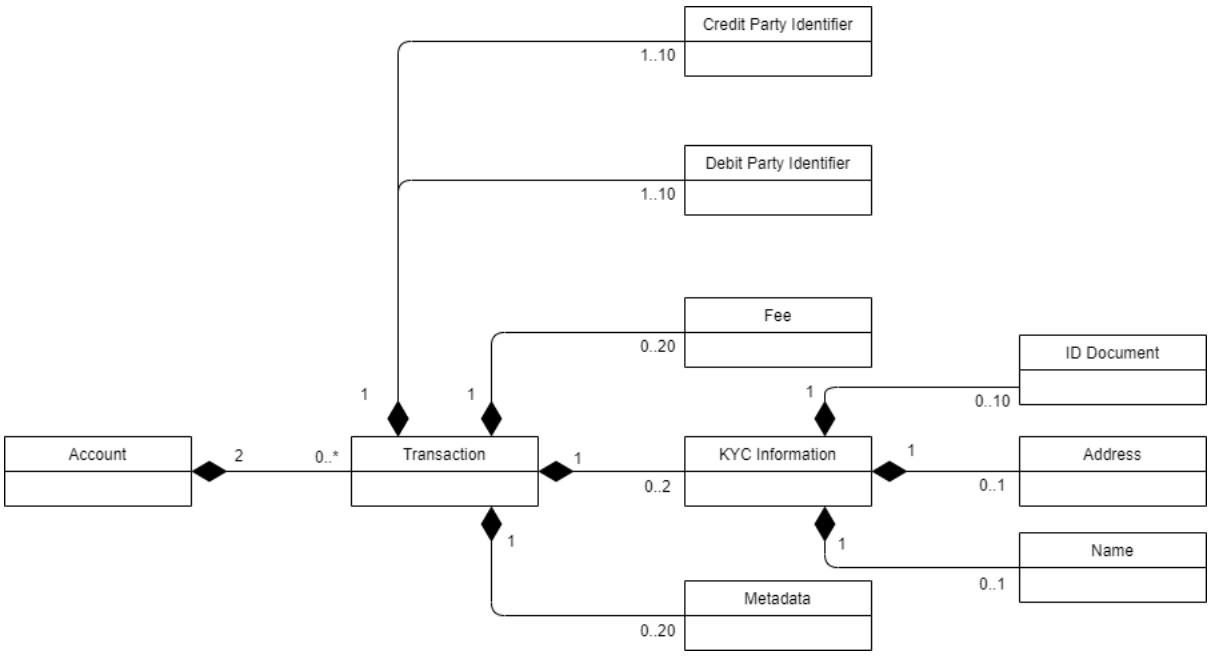

Figure 2-4 Account Transaction UML Class Diagram

2.6 Account Balances API

Using the Account Balances API, an FSP can check their balance. Permitted paths are:

GET /accounts/{identifierType}/{identifier}/balance – single identifier method

or *GET /accounts/{Account Identifiers}/balance* – multiple identifiers method

A 'self' version is also available where the calling API client is the FSP account holder. Path for the 'self' version is */accounts/balance*.

2.6.1 Account Balance UML Class Diagram

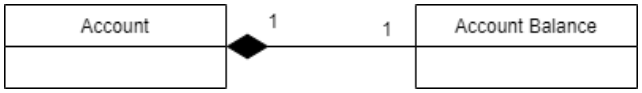

Figure 2-5 Account Balance UML Class Diagram

2.6.2 Account Balance Object Definition

Balance Object					
Name	**Type**	**Description**		**Reference**	**Validation**
accountStatus	string	Indicates a harmonised representation of the account state. This will be shown as 'available' or 'unavailable'. A state of 'unavailable' means that the account is in a state that does not allow posting of transactions. Unregistered indicates that although not available, a transaction created with the account identifier(s) will result in an unregistered voucher creation.	→NA ←O		Enumeration = available, unavailable, unregistered
currentBalance	string	The current outstanding balance on the account.	→NA ←O		Please refer to API Fundamentals document for amount validation rules.
availableBalance	string	Indicates the balance that is able to be debited for an account. This balance is only provided on some API provider systems.	→NA ←O		Please refer to API Fundamentals document for amount validation rules.
reservedBalance	string	Indicates the portion of the balance that is reserved, i.e. intended to be debited. This balance is only provided on some API provider systems.	→NA ←O		Please refer to API Fundamentals document for amount validation rules.

unClearedBalance	string	Indicates the sum of uncleared funds in an account, i.e. those that are awaiting a credit confirmation.	→NA ←O		Please refer to API Fundamentals document for amount validation rules.
currency	string	Currency for all returned balances.	→NA ←O		Enumeration = ISO Currency Codes

2.7 Account Holder Name API

Using the Account Holder Name API, a sending FSP can retrieve the name of the intended recipient of a transfer. This can then be provided to the sending account holder to confirm the recipient name is correct prior to authorising the transfer.

Permitted paths are:

GET /accounts/{identifierType}/{identifier}/accountname – single identifier method

or *GET /accounts/{Account Identifiers}/accountname* – multiple identifiers method

2.7.1 Account Holder Name UML Class Diagram

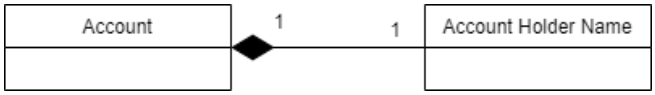

Figure 2-6 Account Holder Name UML Class Diagram

2.7.2 Account Holder Name Object Definition

Account Holder Name Object					
Name	**Type**	**Description**		**Reference**	**Validation**
name	Reference	A collection of fields detailing the name of the primary account holder.	→NA ←O	Name	

3 Supporting Objects

3.1 KYC Information Object

KYC refers to 'Know your Customer'. The KYC object contains a number of fields that enable the identity of subject to be verified. KYC can be provided with transfers for the sending identity and the receiving identity. There are no mandatory KYC object fields.

KYC Information Object					
Name	**Type**	**Description**		**Reference**	**Validation**
birthCountry	string	The country of birth of the KYC subject.	→O ←O		Enumeration = ISO Country Codes
dateOfBirth	date	Birth date of the KYC subject.	→O ←O		
contactPhone	string	Contact phone number (mobile or landline) of the KYC subject.	→O ←O		Must contain between 6 and 15 consecutive digits First character can contain a '+' or digit Can contain spaces.
emailAddress	string	Email address of the KYC subject.	→O ←O		
employerName	string	Employer name of the KYC subject.	→O ←O		
gender	string	Gender of the KYC Object.	→O ←O		Length=1, Enumeration = (m)ale, (f)emale, (u)nspecified
idDocument	array	An array of fields containing the forms of identification that are associated with the subject.	→O ←O	Id Document	
nationality	string	Nationality of the KYC subject.	→O ←O		Enumeration = ISO Country Codes
postalAddress	object	A collection of fields that details the postal address of the KYC subject.	→O ←O	Address	
occupation	string	Occupation of the KYC subject.	→O ←O		
subjectName	object	Refers to the name fields for the KYC subject.	→O ←O	Name	

3.2 Name Object

The name object identifies the name details for the subject identity.

Name Object					
Name	**Type**	**Description**		**Reference**	**Validation**
title	string	The given title of the KYC subject, e.g. Mr, Mrs, Dr.	→O ←O		
firstName	string	First name (also referred to as given name) of the KYC subject.	→O ←O		
middleName	string	Middle Name of the KYC subject.	→O ←O		
lastName	string	Surname (also referred to as last or family name) of the KYC subject.	→O ←O		
fullName	string	The full name of the KYC subject.	→O ←O		
nativeName	string	The full name expressed as in the native language.	→O ←O		

3.3 Id Document Object

As part of KYC information, identification documentation is normally required. The Id Document Object enables documents pertaining to a subject's identity to be described.

Id Document Object					
Name	**Type**	**Description**		**Reference**	**Validation**
idType	string	Indicates the type of identification for the KYC subject, e.g. passport, driving licence etc..	→M ←M		Enumeration = ID Types
idNumber	string	Reference pertaining to the type of identification for the KYC subject.	→O ←O		
issueDate	date	Date of issue for the identification document.	→O ←O		
expiryDate	date	Date of expiry for the identification document.	→O ←O		
issuer	string	Indicates the organisation/government entity that issued the ID document.	→O ←O		
issuerPlace	string	Place of issue for the identification type.	→O		

			←O		
issuerCountry	string	Country where the identification type was issued.	→O ←O		Enumeration = ISO Country Codes
otherIdDescription	string	Where an ID Type of 'otherid' is specified, a description of the type of identification can be provided in this field.	→O ←O		

3.4 Address Object

The address object holds the postal address of the subject. Due to variability of address information in a number of mobile money markets, only country is mandatory.

Address Object					
Name	**Type**	**Description**		**Reference**	**Validation**
addressLine1	string	First line of the address.	→O ←O		
addressLine2	string	Second line of the address.	→O ←O		
addressLine3	string	Third line of the address.	→O ←O		
city	string	City/Town.	→O ←O		
stateProvince	string	State or Province.	→O ←O		
postalCode	string	Postal Code.	→O ←O		
country	string	Country.	→M ←M		Enumeration = ISO Country Codes

3.5 Account Identifier Object

The Account Identifier object enables one or multiple identifiers to be provided to enable the recipient system to resolve the account/party.

Account Identifier Object					
Name	**Type**	**Description**		**Reference**	**Validation**
key	string	Provides the account identifier type.	→M ←M		Enumeration = Account Identifiers

value	string	Provides the account identifier type value.	→M ←M		

3.6 Quote Object

Quotations can consist of multiple quotes. The fields for a quote are defined in the object.

Quote Object					
Name	**Type**	**Description**		**Reference**	**Validation**
quoteId	string	The unique ID for this quote.	→NA ←M		
receivingAmount	string	The total amount as it will be received by the receiving end user.	→NA ←M		Please refer to API Fundamentals document for amount validation rules.
receivingCurrency	string	The currency of the quote.	→NA ←M		Enumeration = ISO Currency Codes
sendingAmount	string	Requested quotation amount as supplied by the sender.	→NA ←M		Please refer to API Fundamentals document for amount validation rules.
sendingCurrency	string	Currency of the requested quotation amount.	→NA ←M		Enumeration = ISO Currency Codes
deliveryMethod	string	The delivery method that is applicable to the quotation.	→NA ←O		Enumeration = Delivery Method
fees	array	Returns all fees that are applicable to the quote.	→NA ←O	Fees Object	
fxRate	string	The conversion rate applicable between the sending and the receiving currency for the requested transaction.	→NA ←O		Please refer to API Fundamentals document for amount validation rules. Note 10 decimal places supported.
quoteExpiryTime	date-time	The timestamp when the quote will expire.	→NA ←O		
receivingServiceProvider	string	The name of the receiving service provider, i.e. the provider that the quote is associated with.	→NA ←O		

3.7 Metadata Object

The metadata object allows fields to be specified to convey administrative information regarding the associated resource in the form of key/value pairs. Additional fields should only be used where no suitable defined field match can be found. The number of key/value pairs is limited to 20.

Metadata Object					
Name	**Type**	**Description**		**Reference**	**Validation**
key	string	Identifies the type of additional fields.	→M ←M		
value	string	Identifies the value of the additional field.	→M ←M		

3.8 Custom Data Object

The custom data object allows additional fields to be specified for the associated resource in the form of key/value pairs. Additional fields should only be used where no suitable defined field match can be found. The number of key/value pairs is limited to 20.

Custom Data Object					
Name	**Type**	**Description**		**Reference**	**Validation**
key	string	Identifies the type of additional fields.	→M ←M		
value	string	Identifies the value of the additional field.	→M ←M		

3.9 Fees Object

An object that enables fees that are differentiated by type to be provided and/or returned.

Fees Object					
Name	**Type**	**Description**		**Reference**	**Validation**
feeType	string	Defines the type of fee.	→M ←M		
feeAmount	string	Defines the amount of the fee.	→M ←M		Please refer to API Fundamentals document for amount validation rules.
feeCurrency	string	Defines the currency for the given fee.	→M ←M		Enumeration = ISO Currency Codes

3.10 Requesting Organisation Object

An object that details the originating organisation of the request.

Requesting Organisation Object					
Name	**Type**	**Description**		**Reference**	**Validation**
requestingOrganisationIdentifierType	string	Identifies the identifier type of the requesting organisation.	→M ←M		‘swiftbic’, ‘lei’, ‘organisationid’
requestingOrganisationIdentifier	string	Contains the requesting organisation identifier.	→M ←M		

4 Enumerations

4.1 ISO Currency Codes

The three-character alphabetic code for currency as defined by ISO 4217 is to be used for all currency fields. The full list of codes is maintained by Swiss Interbank Clearing on behalf of the International Organisation for Standardisation. This list can be obtained via the following website - http://www.currency-iso.org/en/home/tables/table-a1.html.

4.2 Transaction Types

A transaction type is used to classify the nature of a transaction.

Code	Description
transfer	Transfer of funds between mobile money provider and another provider or financial institution in the same country.
reversal	Reversal of a prior transaction to return funds to the payer.

4.3 ID Types

The ID Types enumeration contains accepted identification types. Due to the wide international variation in accepted types of identification, a catch-all type of ‘otherid’ is included.

ID Type	Description
passport	Payment of bill from a business for goods and/or services.
nationalregistration	National Registration Number.
othered	Catch-all for IDs not on the list.
drivinglicence	Driving Licence Number.
socialsecurity	Social Security Number.
alienregistration	Alien Registration ID.
nationalidcard	National Identity Card.
employer	Employers Identification.
taxid	Tax Identification Number.
seniorcitizenscard	Senior Citizens ID Card.
marriagecertificate	Marriage Certificate.
birthcertificate	Birth Certificate.
healthcard	Health Card.
votersid	Voters Identification.
villageelderletter	Letter of confirmation from village elder.
pancard	Credit/debit card number (Primary Account Number).
officialletter	Official letter confirming identity.

4.4 Account Identifiers

The Account Identifier enumeration lists all possible means to identify a target account. Identifiers can be combined if necessary, to provide a unique identifier for the target account.

Code	Short Description	Type	Description
accountcategory	Account Category	string	Can be used to identify the sources of funds category where there are multiple accounts (wallets) held against an account holder.
bankaccountno	Bank Account Number	string	Financial institution account number that is typically known by the account holder.
accountrank	Account Rank	string	Is used to identify the rank of the source of funds where there are multiple accounts (wallets) held against an account holder.
identityalias	Identity Alias	string	An alias for the identity, e.g. short code for an agent till.
iban	IBAN	string	Internationally agreed system of identifying bank accounts across national borders to facilitate the communication and processing of cross border transactions. Can contain up to 34 alphanumeric characters.
accountid	Account Holder Identity	string	Identifier for the account holder.
msisdn	MSISDN	string	Must contain between 6 and 15 consecutive digits First character can contain a '+' or digit Can contain spaces.
swiftbic	SWIFTBIC	string	A bank identifier code (BIC) is a unique identifier for a specific financial institution. A BIC is composed of a 4-character bank code, a 2-character country code, a 2-character location code and an optional 3-character branch code. BICs are used by financial institutions for letters of credit, payments and securities transactions and other business

			messages between banks. Please refer to ISO 9362 for further information.
sortcode	Bank Sort Code	string	Sort code to identify the financial institution holding the account.
organisationid	Organisation Account Identifier	string	Used to identify the organisation for which a payment is to be made.
username	Username	string	Used to identify target account via an associated username.
walletid	Wallet Identifier	string	A means to identify a mobile money wallet, particularly where multiple wallets can be held against an MSISDN. typically used in conjunction with MSISDN or identity alias to identify a particular wallet.
linkref	Link Reference	string	A means to uniquely identify an account via an account to account link. E.g. wallet account link to bank account.
consumerno	Consumer Number	String	Identifies the consumer associated with the account.
serviceprovider	Service Provider	String	Provides a reference for a Service Provider.
storeid	Store ID	String	Identifies the transacting store / retail outlet.
bankname	Bank Name	String	Name of the bank.
bankaccounttitle	Bank Account Title	String	The title of the bank account.
emailaddress	Email Address	String	emailaddress of the party.
mandatereference	Debit Mandate Reference	String	A means to identify an account via a debit mandate reference.

4.5 ISO Country Codes

The two-character alphabetic code for country as defined by ISO 3166 is to be used for all fields specifying a country or nationality. The full list of codes is maintained by the International Organisation for Standardisation. The list can be obtained via the following website - http://www.iso.org/iso/country_codes.

4.6 Delivery Method Types

When a customer requests a transfer quotation they are able to specify their preferred method of delivery of the transfer to the recipient. Permitted delivery methods are provided below.

Delivery Method	Description
directtoaccount	The transfer is to be delivered into the account (wallet) of the recipient.
agent	The recipient can visit an agent and get the transferred funds.
personaldelivery	a supplementary service where an authorised person can deliver the funds, in hand, to the receiving end user.

5 API Sequence Diagrams

The following sequence diagrams illustrate a selection of success and failure flows for P2P transfers using the Mobile Money API. For further information on API behaviour and error handling, please refer to the Mobile Money API Fundamentals document.

5.1 P2P Transfer via Switch

In this diagram, a switch is used by the sending FSP to (1) confirm the recipient name, (2) request a quotation and (3) perform the transfer with the receiving FSP. A callback is provided by the receiving FSP to return confirmation of the transfer.

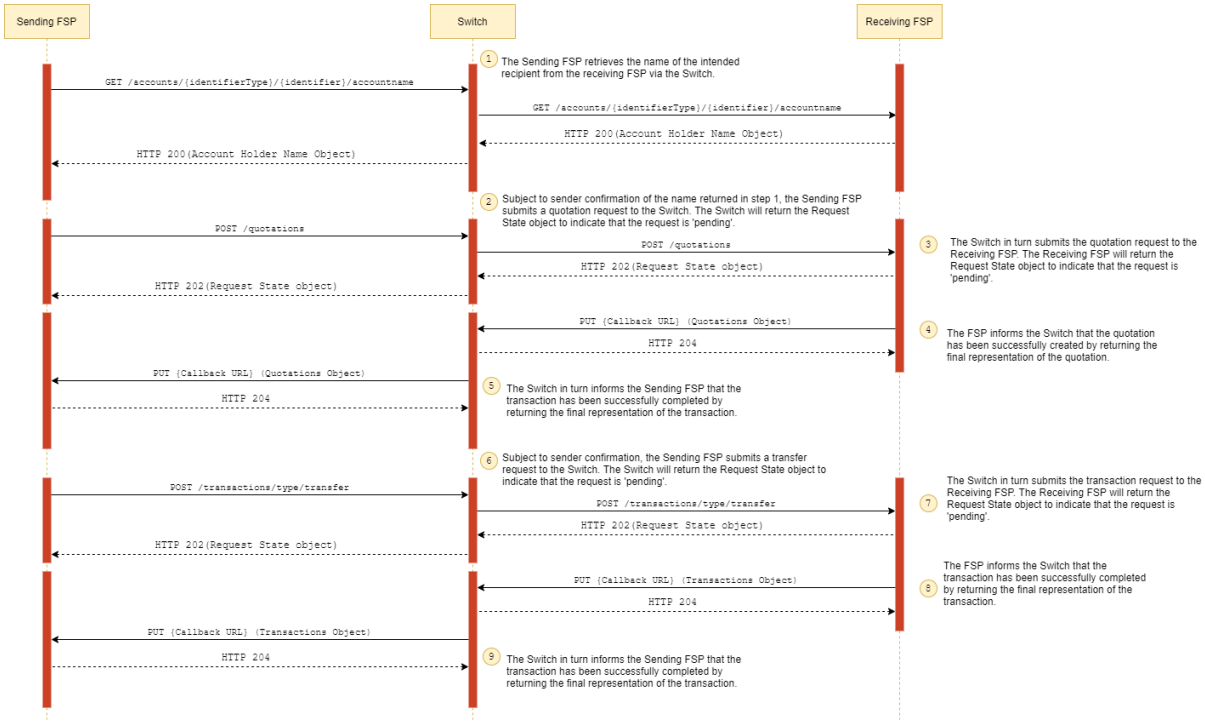

Figure 5-1 P2P Transfer via Switch

5.2 Bilateral P2P Transfer

In this diagram, the sending FSP connects directly with the receiving FSP to confirm the recipient name and to perform the transfer. A callback is provided by the receiving FSP to return confirmation of the transfer. In this example, a quotation is not requested.

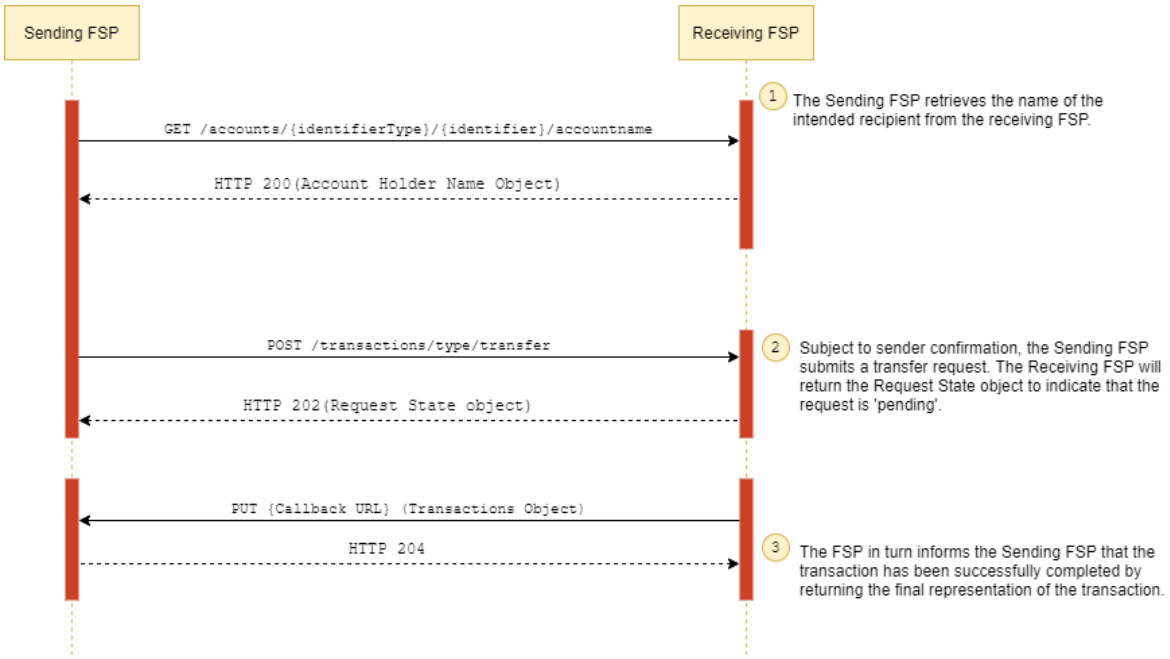

Figure 5-2 Bilateral P2P Transfer

5.3 'On-us' P2P Transfer Initiated by a Third Party Provider

In this diagram, A third party provider enables a sender to transfer money to a recipient in the same FSP. The third party provider (1) confirms the recipient name, (2) requests a quotation and (3) performs the transfer with the FSP. A callback is provided by the FSP to return confirmation of the transfer.

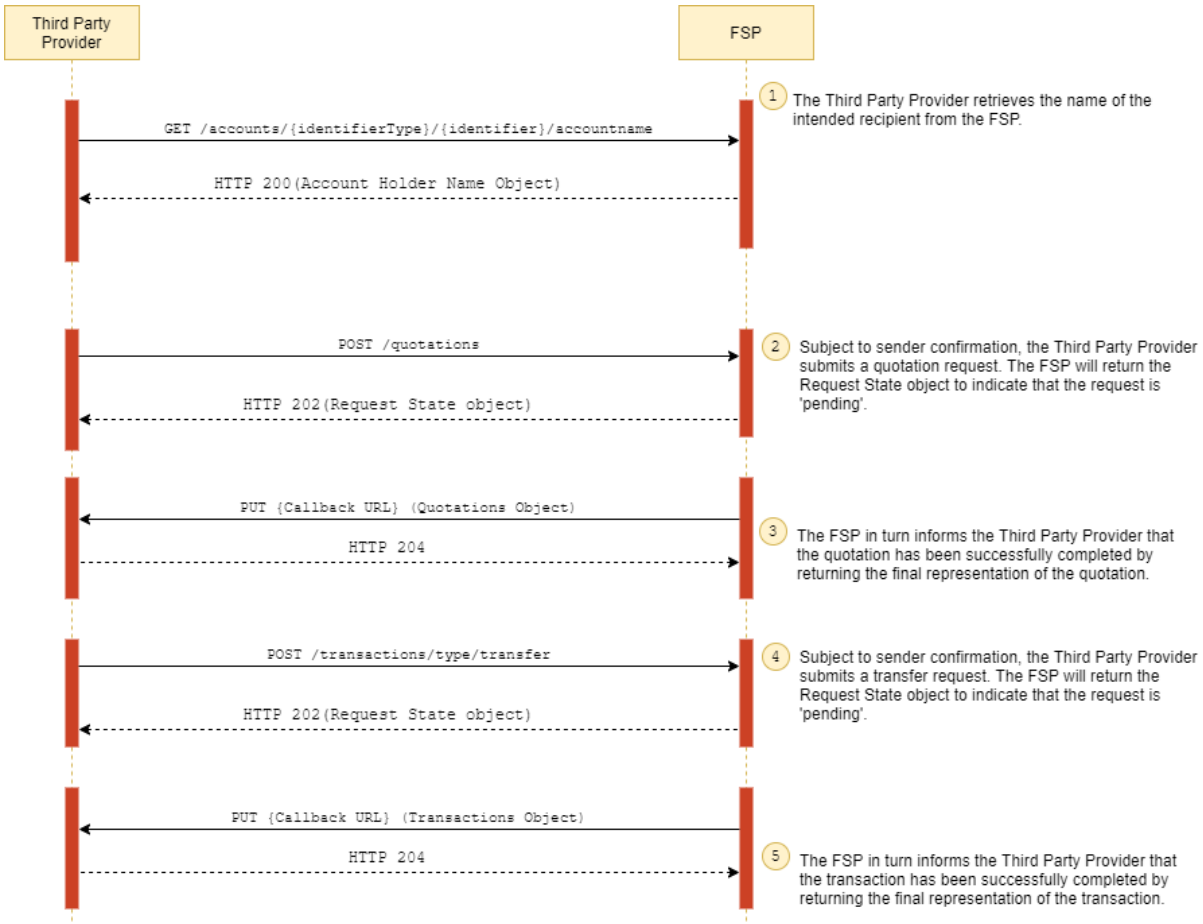

Figure 5-3 Bilateral P2P Transfer

5.4 P2P Transfer Failure

The failure of a transfer is reflected by the return of an error object in the callback from the receiving FSP.

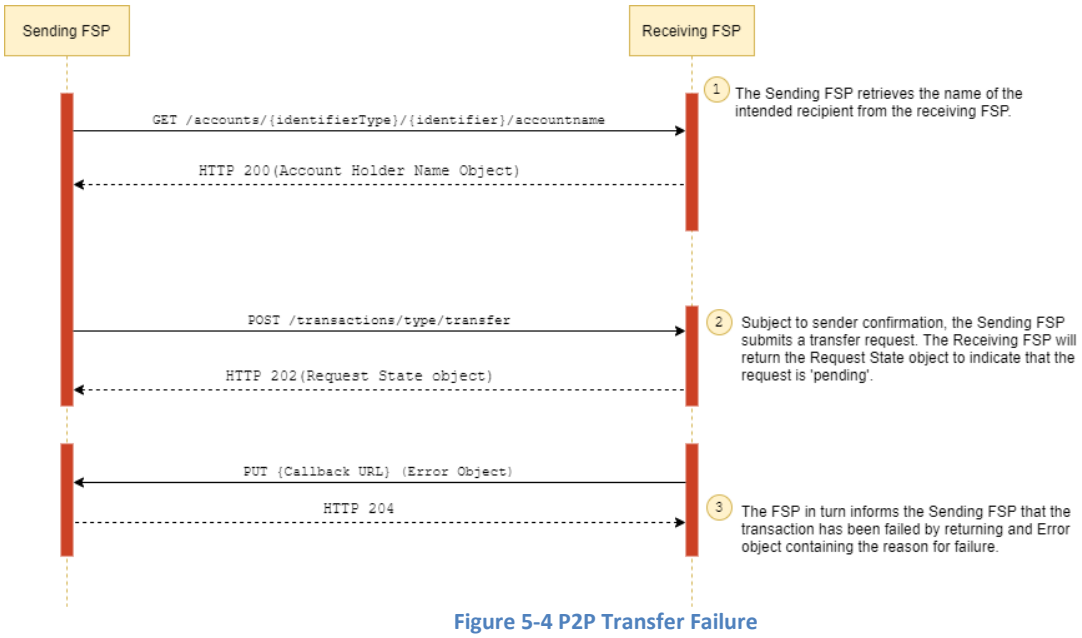

Figure 5-4 P2P Transfer Failure

5.5 P2P Transfer Reversal

In some failure scenarios, a transfer may need to be reversed. This diagram illustrates an reversal with the final result communicated via the callback.

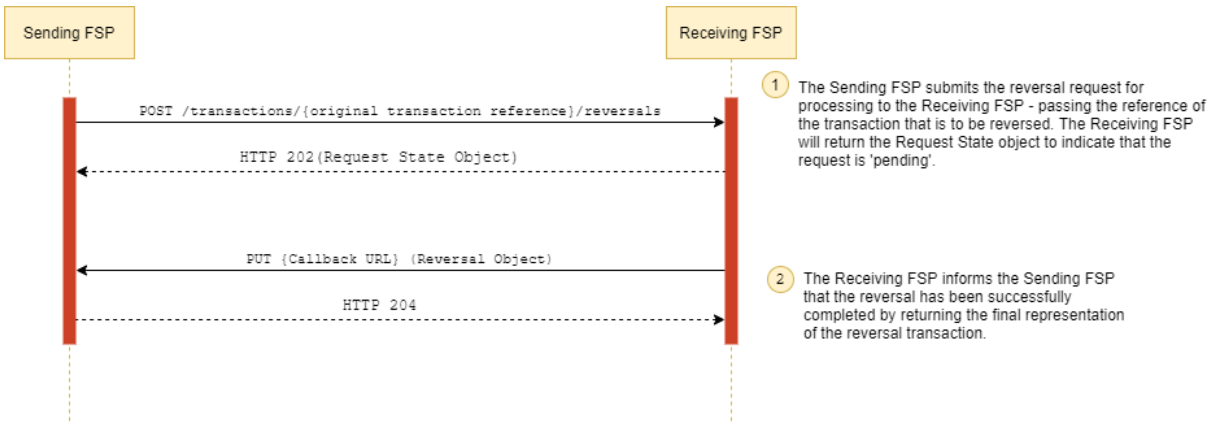

Figure 5-5 P2P Transfer Reversal

5.6 Obtain an FSP Balance

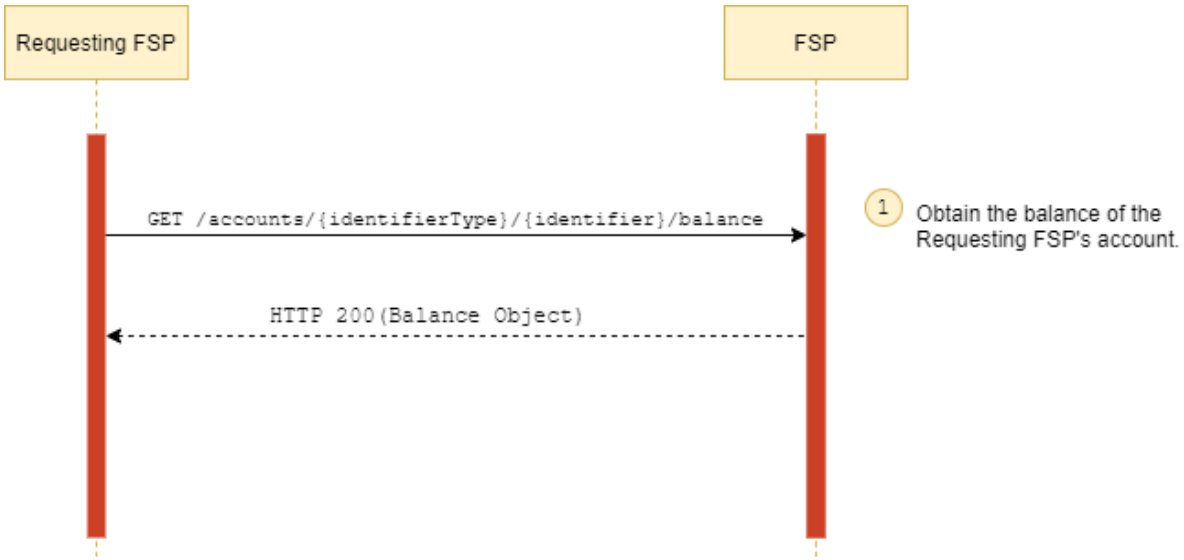

Figure 5-6 Obtain an FSP Balance

5.7 Retrieve Transactions for an FSP

This diagram illustrates use of a cursor mechanism to retrieve all transactions for a requesting FSP via multiple requests.

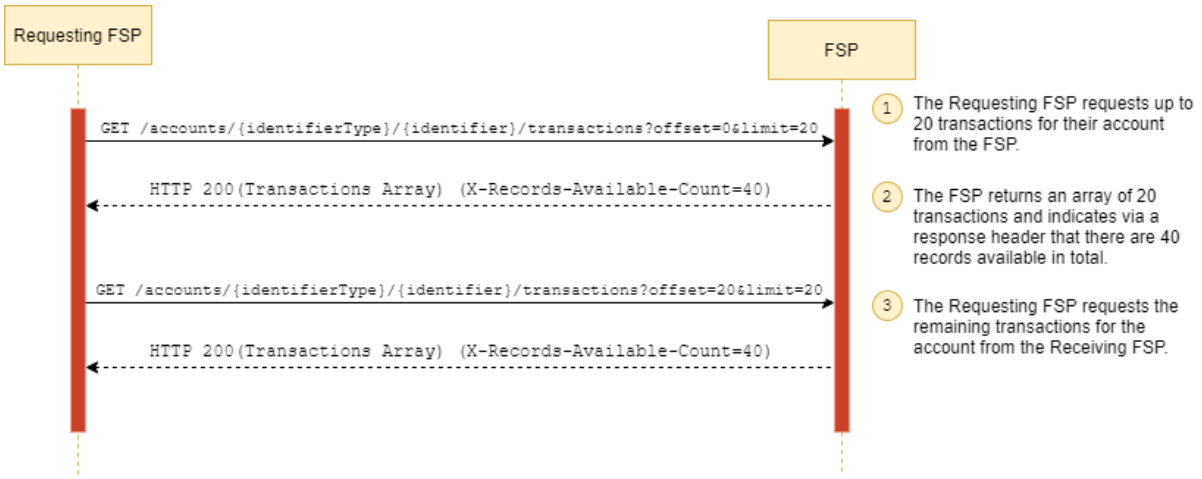

Figure 5-7 Retrieve Transactions for an FSP

5.8 Check for Service Availability

The Heartbeat API is used for monitoring purposes and establishes whether the FSP is in a state that enables a client to submit a request for processing.

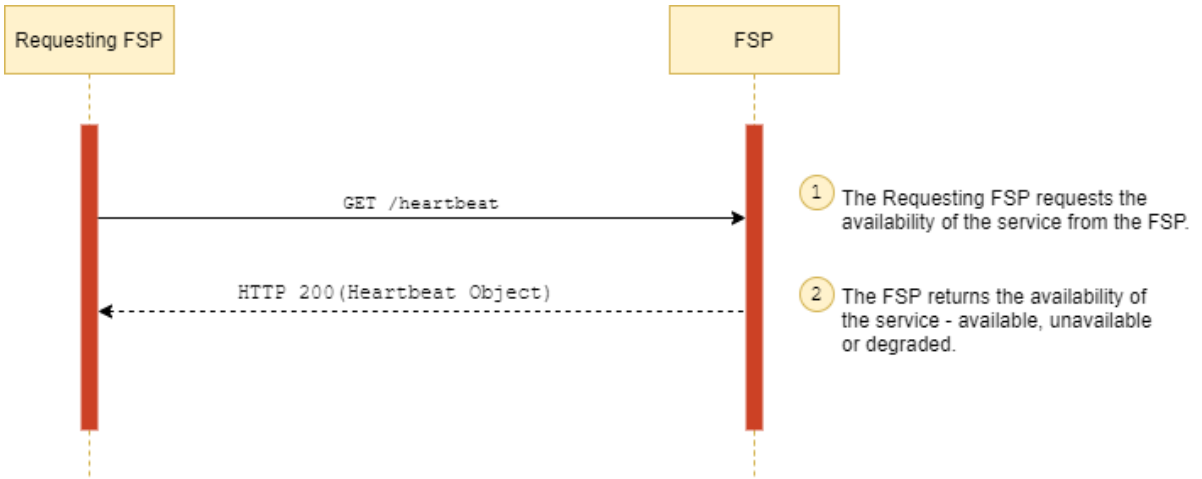

Figure 5-8 Check for Service Availability

5.9 Retrieve a Missing API Response

This API can be used by the sending FSP to retrieve a link to the final representation of the resource for which it attempted to create. Use this API when a callback is not received from the receiving FSP.

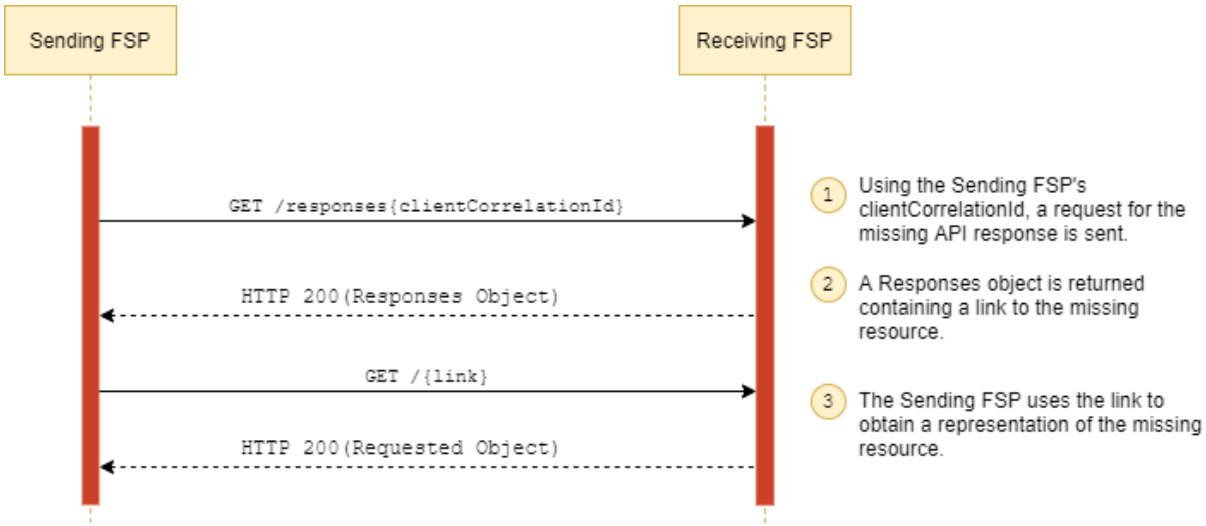

Figure 5-9 Retrieve a Missing API Response

This document is produced by the GSMA with input from the GSMA Mobile Money API Working Group. It is our intention to provide a quality product for your use. If you find any errors or omissions, please contact us with your comments. You may notify us at support.mmapi@gsma.com.