



Mobile Money API Specification 1.2.0 Account Linking

Document Summary

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

© GSMA © 2021. The GSM Association (“Association”) makes no representation, warranty or undertaking (express or implied) with respect to and does not accept any responsibility for, and disclaims liability for the accuracy or completeness or timeliness of the information contained in this document. The information contained in this document may be subject to change without prior notice. This document has been classified according to the GSMA Document Confidentiality Policy. GSMA meetings are conducted in full compliance with the GSMA Antitrust Policy.

Document History

Document Version	API Release Version	Date	Brief Description of Change	Editor / Company
0.1	1.2.0-beta	Aug 2020	<ul style="list-style-type: none"> Initial draft of document 	GSMA
0.2	1.2.0-beta	Sep 2020	<ul style="list-style-type: none"> Implemented RFC 2020-3, RFC 2020-10, RFC 2020-13, 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	4
1.1	Intended Audience	4
2	API Endpoints	5
2.1	Transactions API	6
2.1.1	Transaction UML Class Diagram	6
2.1.2	Transaction Object Definition	6
2.2	Reversals API	10
2.2.1	Reversal UML Class Diagram	10
2.2.2	Reversal Object Definition	10
2.3	Accounts APIs	13
2.3.1	Identifying a Financial Service Provider Account	13
2.4	Retrieving Transactions for an Account	13
2.4.1	Account Transaction UML Class Diagram	14
2.5	Account Balances API	15
2.5.1	Account Balance UML Class Diagram	15
2.6	Links API	17
2.6.1	Link UML Class Diagram	17
2.6.2	Link Object Definition	17
3	Supporting Objects	19
3.1	Account Identifier Object	19
3.2	Metadata Object	19
3.3	Custom Data Object	19
3.4	Fees Object	20
3.5	Requesting Organisation Object	20
4	Enumerations	21
4.1	ISO Currency Codes	21
4.2	Transaction Types	21
4.3	Account Identifiers	21
5	API Sequence Diagrams	24
5.1	Setup an Account Link	24
5.2	Setup an Account Link Failure	24
5.3	Perform a Transfer for a Linked Account	24
5.4	Perform a Transfer for a Linked Account - Failure	25
5.5	Perform a Transfer using an Account Link via the Polling Method	25
5.6	Perform a Transfer Reversal	26
5.7	Obtain a Financial Service Provider Balance	27
5.8	Retrieve Transfers for a Financial Service Provider	27
5.9	Check for Service Availability	27
5.10	Retrieve a Missing API Response	28

1 Introduction

The purpose of this document is to specify the endpoints, fields, objects, and enumerations for Account Linking Mobile Money APIs, which are a subset of the [GSMA Mobile Money API Specification](#). The Account Linking Mobile Money APIs allow financial service providers to link customer accounts to mobile money accounts, thus allowing their customers to push funds to and pull funds from mobile money. Conversely, mobile money providers can use the APIs to link their customers mobile money accounts to financial service providers.

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:

- [API Endpoints](#)
- [Supporting Objects](#)
- [Enumerations](#)
- [API Sequence Diagrams](#)

1.1 Intended Audience

Audience	Usage	Role
FSP - Mobile Money Providers	<ul style="list-style-type: none"> • To understand how to implement the Mobile Money API to accept account linking requests from other financial service providers. 	API Provider
	<ul style="list-style-type: none"> • To understand how to implement the Mobile Money API to request linking of accounts to other financial service providers. 	API Consumer
FSP Banks and Microfinance Institutions	<ul style="list-style-type: none"> • To understand how to implement the Mobile Money API to accept account linking requests mobile money providers. 	API Provider
	<ul style="list-style-type: none"> • To understand how to implement the Mobile Money API to request linking of accounts 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 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

The transaction API can be used by FSPs to make a transfer for a linked account. Prior to making the transfer, a link must have been established between the debit and credit accounts.

The following paths are permitted:

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

2.1.1 Transaction UML Class Diagram

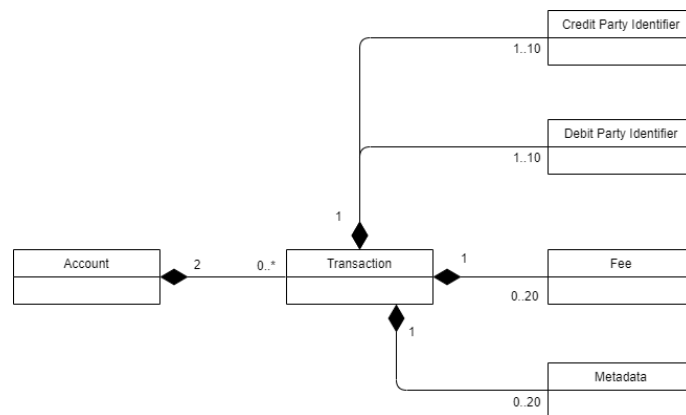


Figure 2-1 Transaction UML Class Diagram

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		
requestingOrganisation	object	The originating organisation of the request.	→O ←O	Requesting Organisation	
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	

Note: To make a transfer against a link, use *linkref* as the account identifier key.

2.2 Reversals API

The Reversals API is used to reverse, adjust or refund a financial transaction. The originating transaction reference must be provided in the path in order to identify the transfer to be reversed. For a partial reversal, the amount needs to be supplied. It should be noted that some API Providers do not support partial reversals and will return an error if a partial amount is supplied.

For viewing reversals, the [Transactions API](#) should be used. For performing a refund without the original transaction reference, use the [Transactions API](#).

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

2.2.1 Reversal UML Class Diagram

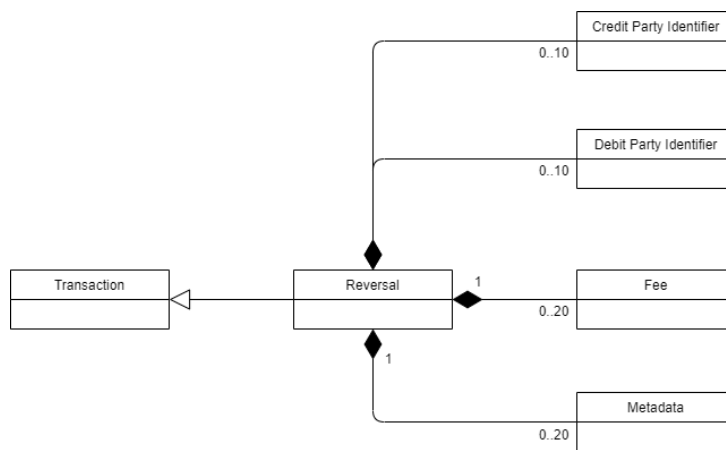


Figure 2-2 Reversal UML Class Diagram

2.2.2 Reversal Object Definition

Reversal Object					
Name	Type	Description		Reference	Validation
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	→NA ←M		

		transaction which is the subject of the reversal.			
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	
servicingId entity	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 Accounts APIs

Using the mobile money APIs, financial service providers can:

- View transfers for their account.
- View their account balance.

2.3.1 Identifying a Financial Service Provider Account

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

2.3.1.1 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}`.

2.3.1.2 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.

2.4 Retrieving Transactions for an Account

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. This can be 'merchantpay', 'reversal' or 'adjustment'

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).

2.4.1 Account Transaction UML Class Diagram

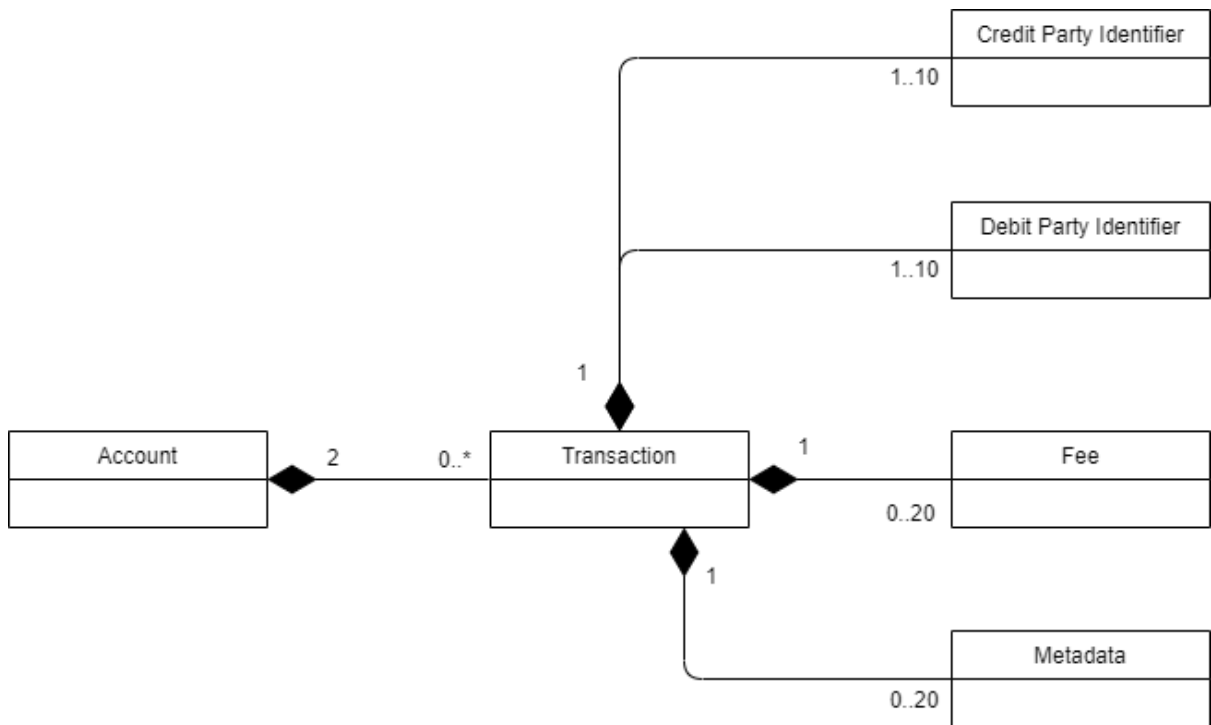


Figure 2-3 Account Transaction UML Class Diagram

2.5 Account Balances API

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

GET /accounts/{identifierType}/{identifier}/transactions – [single identifier method](#)

or *GET /accounts/{Account Identifiers}/transactions* – [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*.

2.5.1 Account Balance UML Class Diagram

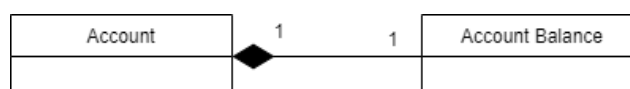


Figure 2-4 Account Balance UML Class Diagram

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 ←0		Please refer to API Fundamentals document for amount validation rules.
currency	string	Currency for all returned balances.	→NA ←0		Enumeration = ISO Currency Codes

2.6 Links API

The Links APIs are used to establish a link between two separate accounts between financial service providers. The API can be used for example to link a mobile wallet account to a microfinance institution account or a bank account. The link object does not mandate the processes to verify and authenticate a link request - this depends upon the use case. A link needs to be associated with a mode of operation:

- **pull.** The link can be used by the client to debit the target account held by the provider.
- **push.** The link can be used by the client to credit the target account held by the provider.
- **both.** The link can be used for Push and Pull requests.

To identify the accounts that are to be linked, the target account is specified in the path whereas the source account is specified in the link object.

The permitted paths are as follows:

- **Creation:** *POST /accounts/{identifierType}/{identifier}/links* or *POST /accounts/{Account Identifiers}/links*.
- **Update** of status and/or mode fields: *PATCH /accounts/{identifierType}/{identifier}/links/{linkReference}* or *PATCH /accounts/{Account Identifiers}/links/{linkReference}*.
- **Read.** *GET /accounts/{identifierType}/{identifier}/links/{linkReference}* or *GET /accounts/{Account Identifiers}/links/{linkReference}*.

Synchronous and asynchronous modes are supported for POST and PATCH methods whereas only synchronous mode is supported for the GET method.

2.6.1 Link UML Class Diagram

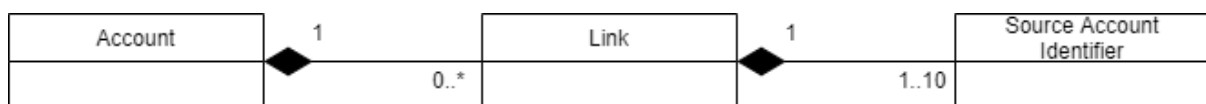


Figure 2-5 Link UML Class Diagram

2.6.2 Link Object Definition

Link Object					
Name	Type	Description		Reference	Validation
linkReference	string	Indicates the Link reference. This enables a linked account to be uniquely identified.	→NA ←M		

sourceAccountIdentifiers	array	A series of key/value pairs that identify the source account. Keys include MSISDN and Wallet Identifier.	→M ←M	Account Identifiers	
mode	string	Indicates the mode of operation for the Link.	→M ←M		Enumeration = push, pull, both
status	string	Indicates the status of the Link.	→M ←M		Enumeration = active, inactive
requestingOrganization	object	The originating organisation of the request.	→O ←O	Requesting Organization	
creationDate	date-time	Indicates when the link was created as recorded by the API provider.	→NA ←O		
modificationDate	date-time	Indicates when the link was modified as recorded by the API provider.	→NA ←O		
requestDate	date-time	The date and time of the link request as supplied by the client.	→O ←O		
customData	string	Contains provider specific fields.	→O ←O	Custom Data Object	

3 Supporting Objects

3.1 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		

3.2 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		

3.3 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		

3.4 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

3.5 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 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.

5 API Sequence Diagrams

The following sequence diagrams illustrate a selection of success and failure flows for account linking 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 Setup an Account Link

This diagram illustrates the setting-up of an account link. The requesting FSP initiates the request which is authorised by the account holding customer. In this diagram, an asynchronous flow is used with a final callback.

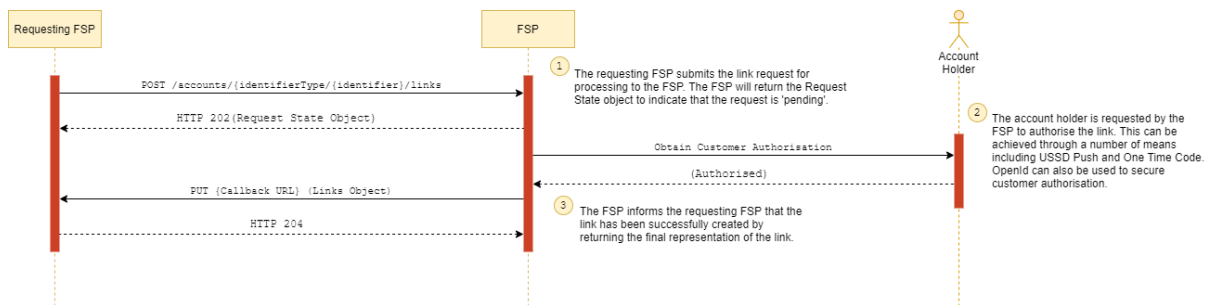


Figure 5-1 Setup an Account Link

5.2 Setup an Account Link Failure

In this diagram, the account holder declines to provide authorisation to setup the account link. The requesting FSP receives a callback containing an error object detailing the reason for failure.

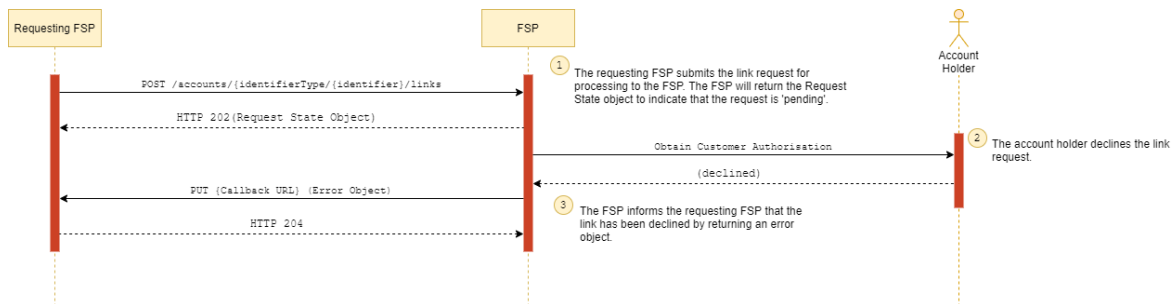


Figure 5-2 Setup an Account Link Failure

5.3 Perform a Transfer for a Linked Account

In this diagram, the requesting FSP initiates a transfer to the FSP using an established link.

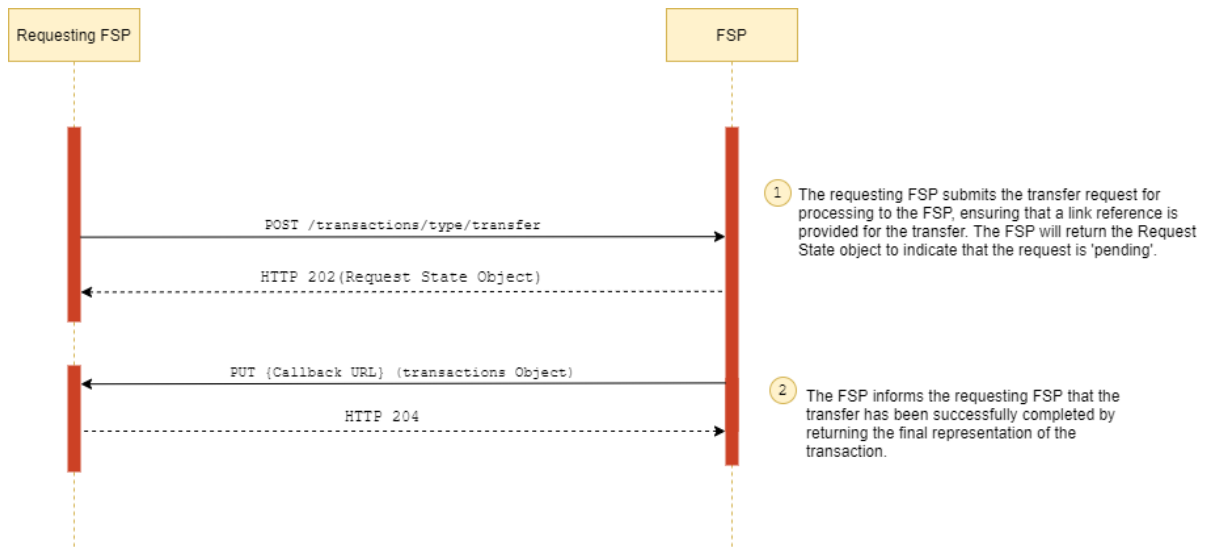


Figure 5-3 Perform a Transfer for a Linked Account

5.4 Perform a Transfer for a Linked Account - Failure

In this diagram, the requesting FSP initiates a transfer to the FSP using an established link. The FSP is unable to process the transfer and returns a callback containing the error object.

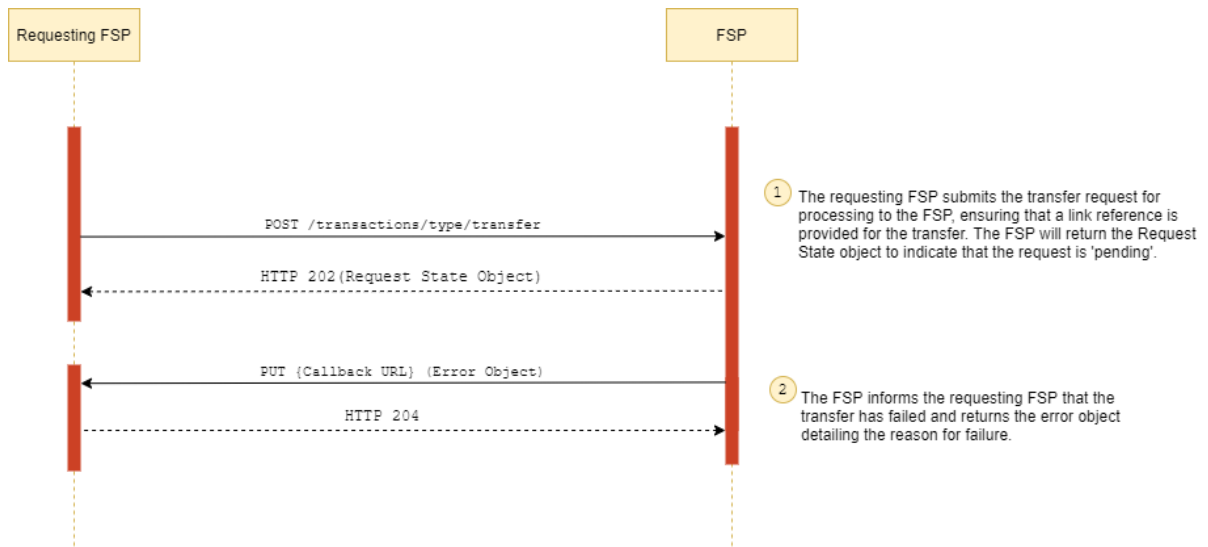


Figure 5-4 Perform a Transfer for a Linked Account - Failure

5.5 Perform a Transfer using an Account Link via the Polling Method

In this example, an asynchronous flow is used with the polling method. The client polls against the request state object to determine the outcome of the transfer request.

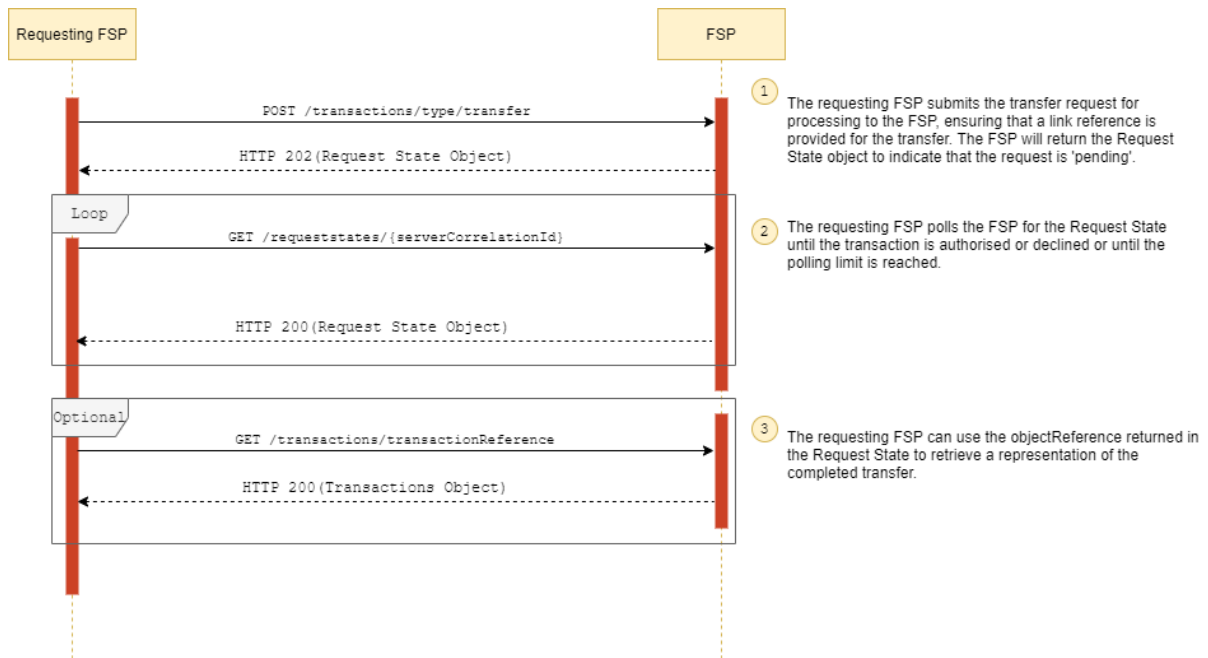


Figure 5-5 Perform a Transfer using an Account Link via the Polling Method

5.6 Perform a Transfer Reversal

In some failure scenarios, a requesting FSP may need to reverse a transaction. This diagram illustrates a reversal with the final result communicated via the callback.

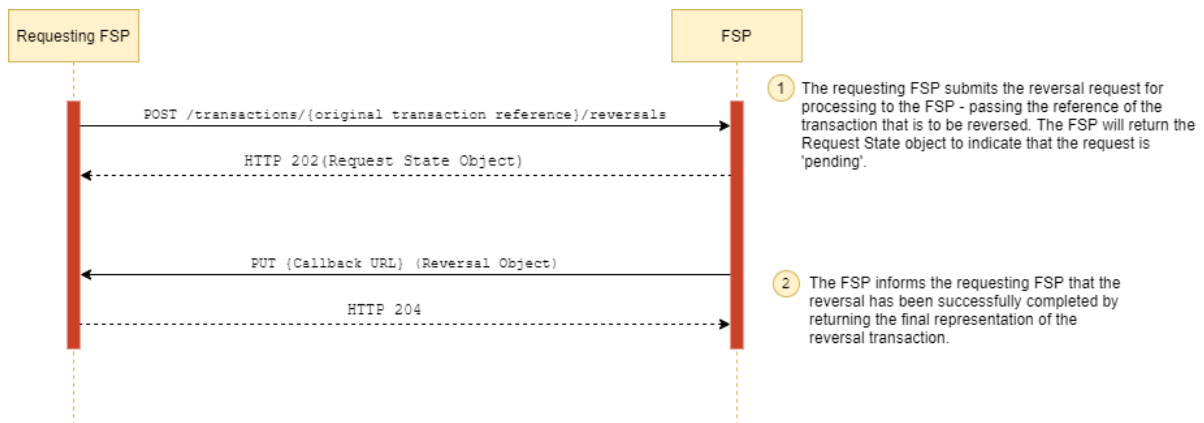


Figure 5-6 Perform a Transfer Reversal

5.7 Obtain a Financial Service Provider Balance

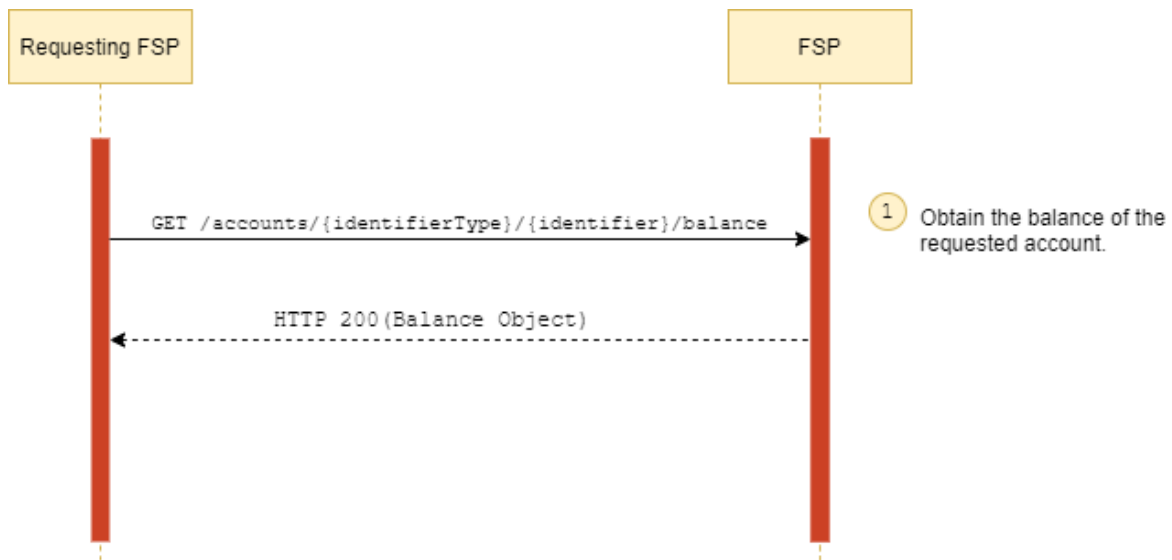


Figure 5-7 Obtain a Financial Service Provider Balance

5.8 Retrieve Transfers for a Financial Service Provider

This diagram illustrates use of a cursor mechanism to retrieve all transfers for a financial service provider via multiple requests.



Figure 5-8 Retrieve Transfers for a Financial Service Provider

5.9 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.

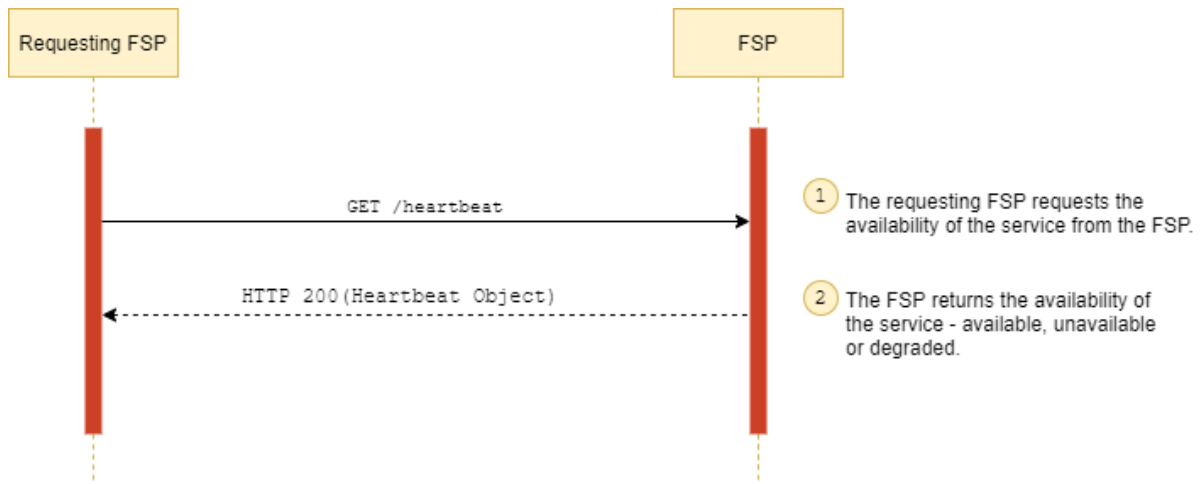


Figure 5-9 Check for Service Availability

5.10 Retrieve a Missing API Response

This API can be used by the requesting 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 FSP.

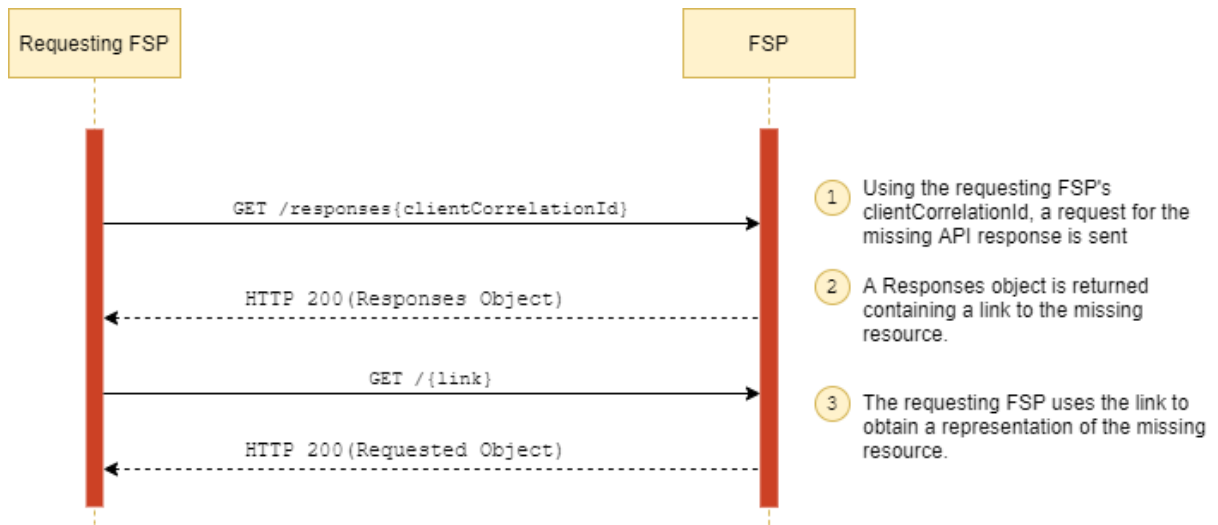


Figure 5-10 Retrieve a Missing API Response

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.mmapl@gsmal.com.