

Change Request Form



Mobile Money API 1.0.0 Specification Description

Document Summary

Official Document Number, Document Title and Version Number	Mobile Money API Definition v1.0.0 (Current)
Official Document Type	Non-binding Permanent Reference Document
Change Request Security Classification	Non-confidential
Is this a new document or a Major or Minor Change?	New Document
Will this Change Request result in a Major or Minor version update?	Major Version
This document is for	Approval
Additional Contributors	GSMA
Issuing Group/Project	DCMM-PLG
Approving Group/Project	PSMC
Change Request Creation Date	11/08/2016
What are the reasons for and benefits of creating this new document or Change Request?	Accessible APIs are a core requirement to ecosystem growth. However, currently, the API landscape is scattered, inconsistent and fragmented making it hard for operators and 3rd parties to interconnect efficiently – an issue a harmonised API can solve.

© GSMA © 2019. 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.

Mobile Money API 1.0.0 Specification Description

Review Log (to be completed by GSMA Support Staff)			
Workflow Step	Document Review Comments	GSMA Support Staff Name	Comments Date
Step 1: Change Request Creation (no comments required)			
Step 2: Document Quality and/or Legal Review			
Document Quality Team	INSERT COMMENTS HERE Please enter details for the Quality Review Confirm Document Quality Team feedback Record any issues, actions and key decisions	GSMA Support Staff Name	DD/MM/YY
Legal Review	INSERT COMMENTS HERE Please enter details for the Legal Review Confirm Legal feedback Record any issues, actions and key decisions	GSMA Support Staff Name	DD/MM/YY
Step 3: Formal Review			
Group(s)/Project(s) Review(s) Comments and Feedback	INSERT COMMENTS HERE Please enter details for the Group(s)/Project(s) Review(s) Record any issues, actions and key decisions Confirm outcome of Formal Review	GSMA Support Staff Name	DD/MM/YY
Step 4: Formal Approval(s)			
Group(s)/Project(s) Approval(s) Comments and Feedback	INSERT COMMENTS HERE Please enter details for the Group(s)/Project(s) Approval(s) Record any issues, actions and key decisions Confirm outcome of Formal Approval	GSMA Support Staff Name	DD/MM/YY

Table of Contents

1	Introduction	5
1.1	Definitions of Terms	5
2	API Quick Start Guide	7
3	API Fundamentals	9
3.1	URI	9
3.2	Operations	9
3.3	Patch Specifics	9
3.4	Resource Naming	9
3.5	Identifying the Resource	10
3.6	Client Correlation ID	10
3.7	Use Case Flow Patterns	11
3.8	Case Sensitivity	11
3.9	HTTP Header Information	11
3.9.1	HTTP Request Headers	11
3.9.2	HTTP Response Headers	12
3.9.3	Custom Headers	12
3.10	API Versioning	12
4	API Service Definition	14
4.1	Transactions API	15
4.2	Reversals API	17
4.3	Batch Transactions API	20
4.3.1	Batch Transactions API	20
4.3.2	Batch Rejections API	22
4.3.3	Batch Completions API	23
4.4	Accounts APIs	24
4.4.1	Identifying a Target Account	24
4.4.2	Supported Account Operations	24
4.4.3	Returning Transactions for an Account	25
4.4.4	Accounts Status API	25
4.4.5	Account Balances API	26
4.4.6	Account Name API	27
4.4.7	Statement Entries API	27
4.5	Bills API	29
4.6	Bills Payments API	30
4.7	Debit Mandates API	31
4.7.1	Debit Mandates Object	31
4.8	Links API	32
4.8.1	Link Object	32
4.9	Quotations API	33
5	Supporting Objects	36
5.1	International Transfer Information Object	36
5.2	KYC Information Object	37

Mobile Money API 1.0.0 Specification Description

5.3	Name Object	38
5.4	ID Document Object	38
5.5	Address Object	39
5.6	Account Identifiers Object	39
5.7	Quotes Object	40
5.8	Metadata Object	41
5.9	Supplementary Bill References Object	41
6	Enumerations	42
6.1	ISO Currency Codes	42
6.2	Transaction Types	42
6.3	ID Types	42
6.4	Account Identifiers	43
6.5	ISO Country Codes	44
6.6	Delivery Method Type	44
6.7	Frequency Type	45
7	API Behaviour & Error Handling	46
7.1	Request States	46
7.2	API Behavioural Model	47
7.2.1	Overview	47
7.2.2	Request State Object	47
7.2.3	Sequence Flows	48
7.2.4	API Flow Patterns	48
7.3	HTTP Status Responses and Error Categories	55
7.3.1	Client Error Categories	55
7.3.2	Server Error Categories	55
7.4	Error Codes Definition	56
7.5	Errors Object Definition	57
7.6	API Heartbeat	57
7.7	Missing Response Retrieval	58
7.8	Harmonised Error Codes	59
Annex A	Document Management	62
A.1	Document History	62
A.2	Other Information	62

Mobile Money API 1.0.0 Specification Description

1 Introduction

The purpose of this document is to detail the design principles, objects, behaviours and error handling for the Mobile Money API.

The overriding goal of the API is to enable all parties to implement mobile money API's in a flexible, yet consistent manner. This has been achieved by implementing the following principles:

- Use of REST architectural principles.
- Providing a set of well-defined objects that are abstracted from the underlying object representations held in the various mobile money systems. This allows an API client to construct an API message without requiring specific knowledge of the target server implementation.
- Creation of a standard set of transaction types and other key enumerations, removing the need for developers to map for each and every API implementation.
- Use of ISO international standards for enumerators such as currency and country codes
- Use of supplementary metadata and sub-types to enable use case and/or mobile money provider-specific properties to be conveyed where necessary.
- Recognising that no common mobile money account identifier exists, use of a flexible construct to enable the target account(s) and transaction parties to be identified using one or multiple identifier types.

This documentation contains the following sections:

- [Quick Start Guide](#). Provides a basic reference point to enable a developer to identify which objects and URI's to use for their target use cases.
- [API Fundamentals](#). The core principles and constructs that underpin the API.
- [API Service Definition](#). Details the available [Request State Object](#) API Services.
- [API Supporting Object Definition](#). Details the properties for the supporting objects that constitute the harmonised API.
- [API Enumerations](#). Describes all of the list of values that are applied to specific object properties. Where international standards are used, appropriate references are supplied.
- [API State and Error Handling](#). Describes behavioural aspects of the API and details error handling including error code definition, Heartbeat object definition and polling/callback standards on retries and timeouts.

1.1 Definitions of Terms

Term	Description
ATM	Automated Teller Machine. Enables customer to be able to perform financial transactions without the need for a human cashier. For mobile money, the ATM machine can be considered as an automated agent, allowing withdrawals and some case deposits from and to the customer's wallet.

Mobile Money API 1.0.0 Specification Description

Agent	A person or business that is contracted to facilitate transactions for users. The most important of these are cash-in and cash-out (i.e. loading value into the mobile money system, and then converting it back to cash again); in many instances, agents register new customers too.
API Service	An API Service comprises of a URI which when combined with a HTTP Operation (GET, PATCH, POST) enables a specified operation to be conducted on a specific resource or set of resources.
Application Programming Interfaces (APIs)	In computer programming, an application programming interface (API) is a set of routines, protocols, and tools for building software applications. An API expresses a software component in terms of its operations, inputs, outputs, and underlying types.
Authentication	Confirms the identity of the entity that is requesting the service by verifying supplied credential(s). In this document, the requesting entity is typically referred to as the initiator or sender.
Authorisation	Authorisation provides an authenticated identity with access to the system resources/services to which they are entitled. In the context of this document, the requesting entity (initiator) is provided with a defined set of services and functions that they can perform over API.
Business	Describes an entity such as a public limited or limited company that uses mobile money as a service. This includes taking bill payments, making bill payments and disbursing salaries
Client	Describes the calling system/party that initiates and interacts with the API service. The Client may or may not be the debit or credit party
GUID	Globally Unique Identifier. GUIDs are usually stored as 128-bit values and are used for as the datatype for correlation IDs in the mobile money API.
HTTP	Acronym for Hyper Text Transfer Protocol. HTTP is the foundation of data communication for the World Wide Web. HTTP is the protocol to exchange or transfer hypertext.
ISO	International Organisation for Standardisation. A number of ISO standards are used with the mobile money API.
ITU	International Telecommunications Agency. ITU MSISDN format standards have been adopted for the mobile money API.
JSON	Acronym for JavaScript Object Notation. Used to transmit data objects containing attribute/value pairs. One of two protocols (the other being XML) considered as the basis for a Data Exchange format for mobile money API harmonisation
Merchant	A retail merchant that accepts e-Money from mobile money customers in exchange for goods and/or services.
MSISDN	A number uniquely identifying a subscription in a GSM or a UMTS mobile network. Analogous with a mobile phone number.
Object	An object holds a set of properties. Objects can be combined or used in isolation to provide a representation of a resource.
Organisation	Describes a non-business entity such as a charity or government department that uses mobile money as a service. This includes taking bill payments, making bill payments and disbursing salaries
Provider	Hosts the Mobile Money API services (or subset of).
Resource	A resource is the target for an API operation as identified in the API request. Resources are conceptually different from the representations that are returned to the client.

Mobile Money API 1.0.0 Specification Description

REST-based API	<p>In computing, Representational State Transfer (REST) is the software architectural style adopted by many organisations that offer APIs. REST stipulates that interacting systems exhibit the following characteristics:</p> <ul style="list-style-type: none"> • Client-Server. Responsibilities between clients and servers are explicitly defined. • Stateless. No context is stored between client and server during communication. • Cacheable. As on the World Wide Web, clients and intermediaries can cache responses • Layered System. A client cannot ordinarily tell whether it is connected directly to the end server, or to an intermediary along the way. <p>Uniform Interface. A uniform interface simplifies and decouples the architecture, which enables each part to evolve independently.</p>
URI	Acronym for Uniform Resource Identifier. The URI is a string of characters used to identify a resource (such as an account, for example). In APIs a URI is represented in the form of a URL which is generally used to identify the service to invoke.
Wallet	Term used in this document to describe the store of value that identities (customers, agents, merchants and businesses) use to store e-Money. In general, the document uses the term Account in place of Wallet.

2 API Quick Start Guide

I want to	URI Format	Refer to these sections
Make a Bill Payment	/accounts/bills/payments	Bill Payments API
Create or view a Disbursement	/transactions	Transactions APIs
Create or view a Deposit	/transactions	Transactions APIs
Create or view an International transfer	/transactions	Transactions APIs
Create or view a Merchant Payment	/transactions	Transactions APIs
Create or view an Adjustment (including refunds)	/transactions	Reversals API
Create or view a Reversal	/transactions	Reversals API
Create or view a Domestic Transfer	/transactions	Transactions APIs
Create or view a Withdrawal	/transactions	Transactions APIs
View an Account Status	/accounts/status	Accounts APIs Accounts Status API
View Balance(s) for an account	/accounts/balance	Accounts APIs Balance API
View an Account Holders Name (Name Lookup)	/accounts/accountname	Accounts APIs Account Name API
View Account Statement Entries for an account	/accounts/statemententries	Statement Entries API

Mobile Money API 1.0.0 Specification Description

View a specific statement Entry	/statemententries	Statement Entries API
View Bills for presentment	/accounts/bills	Bills API
Create or update Debit Mandates	/accounts/debitmandates	Debit Mandates API
Create or view an International Transfer Quotation	/quotations	Quotations API
Create, view or update an account to account Link	/accounts/links	Links API
Create a batch of transactions	/batchtransactions	Batch Transactions API
Retrieve batch transaction errors	/batchtransactions/rejections	Batch Transactions API
Retrieve batch transaction completions	/batchtransactions/completions	Batch Transactions API

3 API Fundamentals

3.1 URI

All services exposed by the harmonised Mobile Money API use the following URI format:

{...}/{version}/mm/{Resource}

Where:

- ... is defined upon implementation of the API by the API provider.
- **version** is as per standards defined in the [API Versioning](#) section.
- **mm** is literal for 'Mobile Money'
- **resource** identifies the object and resource that is the subject of the API.

3.2 Operations

The API supports three types of operations, each represented by HTTP Verb as shown below:

- **POST**. Used to create a resource for a given resource type. Transactions, Quotations and Debit Mandates can be created using the API.
- **PATCH**. Used to update a resource for a given resource type. Debit Mandates and Links can be updated using the API. But note that only specific properties can be updated – refer to the relevant API service for more information.
- **GET**. Used to return a representation(s) of a resource(s) or collection of resources. All resources support GET operations.

3.3 Patch Specifics

Updates to resources are accomplished by use of the HTTP PATCH operation. The proposed PATCH format is based upon [IETF RFC 6902](#). Two specific operations are supported:

Replace. This operation replaces the value of the target property with the supplied value. An example of a **replace** operation is `[{"op": "replace", "path": "/XYZ", "value": test}]` where XYZ is the target property.

3.4 Resource Naming

The format of the resource part of the URI identifies the type of resource and if applicable, the specific **resource** for which an operation is to be performed. Resources are generally reflected in plural and by use of nouns. Primary URI constructs for the API are as follows:

/transactions. Identifies a transaction resource.

- **/accounts**. Identifies the financial resource that holds the funds. Examples include bank accounts and mobile wallets.
- **/quotations**. The resource that holds the quote or quotes for international remittance quotations.

Mobile Money API 1.0.0 Specification Description

- **/statemententries**. The resource that returns an account statement representation. This is typically qualified by /accounts but can be used in isolation to return a specific statement entry if the transaction reference is known.
- **/accounts/debitmandates**. The resource that holds the debit mandate instructions, i.e. mandates that allow a Payee to deduct funds from a Payers account.
- **/accounts/links**. Represents an account to account link between two systems.
- **/accounts/bills**. Represents a bill that is due to be paid or has been paid.
- **/accounts/accountname**. Represents the primary account holders name.
- **/accounts/balance**. Represents the balance(s) associated with an account.
- **/accounts/status**. Represents the ability of the account to receive/send funds.
- **/batchtransactions**. Represents a collection of transactions and related batch metadata.

A more exhaustive list of permitted resources and permitted resource hierarchies can be found in the detailed sections of this document.

3.5 Identifying the Resource

The harmonised Mobile Money API has adopted a RESTful approach to identifying the resource on which an operation is to be performed. Where possible, the identifier is specified following the resource name in the URI. In some cases, more than one identifier is required to identify the resource or there is a choice on which identifier to use. Therefore, flexibility has been built into the URI to cater for these cases. This is illustrated in the following table.

Resource	Operation	Identifier	Identifier Placement
Transactions	GET	Transaction Reference	URI
Accounts	GET	Various Account Identifiers (see Accounts for details)	URI
Mandates	PATCH and GET	Mandate Reference	URI
Quotations	PATCH and GET	Quotation Reference	URI
StatementEntries	GET	None if qualified by Accounts. If individual entry, then use Transaction Reference	URI
Links	PATCH and GET	Link Reference	URI
Bills	GET	Account Identifiers	URI
Bill Payments	POST	Bill Payment Reference and Account Identifiers	URI
BatchTransactions	PATCH and GET	Batch Reference	URI

3.6 Client Correlation ID

A Client Correlation ID is to be supplied by the API client on HTTP POST and PATCH requests. Put simply, the Client Correlation ID is a unique identifier that enables the client to

Mobile Money API 1.0.0 Specification Description

correlate the API request with the resource created/updated by the provider. The client correlation ID is captured in the [HTTP Header](#).

The format for the correlation ID is a GUID. Critically, the correlation ID supports safe operations. A POST request that is submitted with a correlation that has already been supplied will be rejected as unsafe, thus avoiding transaction duplication.

3.7 Use Case Flow Patterns

All Use Cases supported by the API are built on standard flow patterns. Flow patterns exist for viewing, creating and updating records (resources). Flow patterns can be combined in a single use case. For example, a domestic transfer may involve a name check on the recipient platform prior to a request to create a transaction. There are some simple rules to follow when using flow patterns. These are illustrated below:

- **Viewing.** Only synchronous flows are supported for viewing resource(s). For example, a HTTP GET request to view balances must result in a HTTP response yielding the representation of the resource or representation of the error.
- **Creating/Updating.** Synchronous and asynchronous flows are supported. A synchronous flow involves a definitive result (resource representation or error representation returned) being provided in the HTTP POST response. An asynchronous flow is supported by two mechanisms – Callback and Polling. These are described in more detail in the [Request State Object](#) section.

3.8 Case Sensitivity

All enumeration values referenced within the API use lower case notation – this includes acronyms and abbreviations. The only exceptions are for ISO Codes (country and currency) – the API uses these codes as defined per ISO.

In this document properties are split into words with the first character capitalised for readability. However, the API properties in the actual API (Swagger definition) use camelCase format.

3.9 HTTP Header Information

The following header information can be supplied for the mobile money API.

3.9.1 HTTP Request Headers

Header	Value	Optionality	Notes
Accept	application/json	Mandatory	
Accept-Charset	Utf8		
Authorization	Authorization: Basic {base64Encode(concatenated client's username and password)}	See notes	Mandatory unless OAuth2 scheme is used
Content-Length	{length of request content in 8-bit bytes}	Mandatory	

Mobile Money API 1.0.0 Specification Description

Content-Type	application/json	Mandatory	
Date	{The date and time that the message was originated in HTTP-date format - RFC 7231}	Mandatory	

3.9.2 HTTP Response Headers

Header	Value	Optionality	Notes
Content-Length	{length of response content in 8-bit bytes}	Conditional	Applicable only if the HTTP response contains JSON body
Content-Type	application/json; charset=utf-8	Conditional	Applicable only if the HTTP response contains JSON body
Date	{The date and time that the message was sent in HTTP-date format - RFC 7231}	Mandatory	Mandatory unless OAuth2 scheme is used

3.9.3 Custom Headers

Header	Value	Optionality	Notes
X-API-Key*	Used to pass pre-shared client's API key to the server	Mandatory	
X-Client-ID*	Used to pass pre-shared client's identifier to the server	Mandatory	
X-Content-Hash*	SHA-256 hex digest of the request content (encrypted or plain)	Conditional	Applicable only if the HTTP request contains JSON body
X-Correlation-ID	GUID	Mandatory	See Client Correlation ID
X-Records-Available-Count	Integer containing number of records that are available to be returned	Conditional	Used in the event of pagination
X-Records-Returned-Count	Integer containing the number of records that have been returned	Conditional	Used in the event of pagination

*Please refer to the Mobile Money API Design for further information.

3.10 API Versioning

When changes are made to the Mobile Money API, a new version is released. There are two types of API versions – minor (backwards compatible) and major (backwards-incompatible) versions.

Mobile Money API 1.0.0 Specification Description

The following types of changes are considered to be backwards compatible and hence minor:

- Addition of new API Services.
- Adding optional request properties and/or optional input parameters such as query strings to existing objects.
- Addition of new properties to existing API responses.
- Changing the order of properties within a request or response object.
- New error codes.

The following types of changes are considered to be backwards incompatible and hence major:

- Introducing mandatory properties.
- Changing datatypes on properties.
- Changes to API URIs.

The version that a client intends to use is indicated URI. Format is 'X.Y' where 'X' is the major version and 'Y' is the minor version. Both versions are sequentially numbered. Note that when a major version is incremented, the minor version is reset to zero. The initial released version of the API is 1.0.

There are many different mobile money and third party providers and hence clients connecting to multiple providers are likely to interact with different versions of the API. Client code developed against an older minor version on one client will work on all newer minor versions existing on other clients. However, client code developed against a new minor version will not necessarily work against older minor versions.

4 API Service Definition

All API Services that form the harmonised mobile money API and related objects are detailed within the API Definition and Supporting Object sections. The relationships between objects is illustrated in the figure below.

Note: That green represents objects that are accessible via an API service and blue represents objects that are referenced.

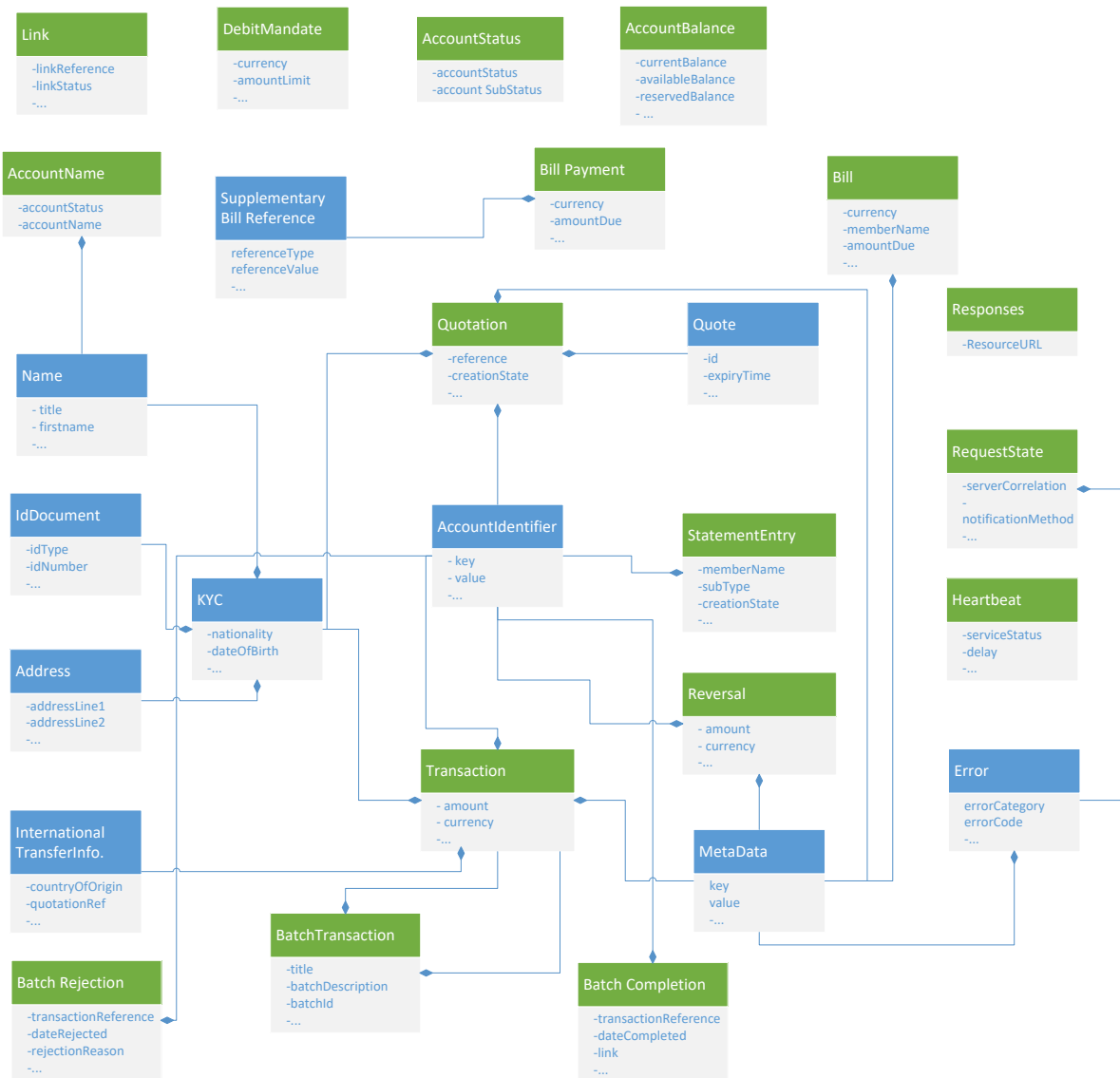


Figure 1. API Object Relationships

Object properties are detailed within tables that contain the following information:

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

Mobile Money API 1.0.0 Specification Description

- Request optional
- ← Response optional
- O Property is optional
- M Property is mandatory
- NA Property does not need to be supplied. If supplied, it will be ignored.

Reference where the property is a collection/array and is defined by another object.

Validation applied to the property, including enumeration, property length and use of regular expressions to validate format.

Note: All String properties have a default maximum length of 256 characters unless specified otherwise.

4.1 Transactions API

The Transactions API is used for all operations involving mobile money financial transactions. This currently covers:

- Creating a Transaction (**POST**)
- Returning a representation of one or more transactions (**GET**)

Transactions are used for a wide range of use cases including Merchant Payments, International Transfers, Domestic Transfers and agent cash-in/cash-out. Reversals and Adjustments are also treated as Transactions.

URI is consistent for all transactions and format is **/transactions**.

The specific resource can be identified by Transaction Reference as per below:

Operation	Identifier	Identifier Placement
GET	Transaction Reference	The format is /transactions/{Transaction Reference}

The object definition for Transactions as provided below:

Transaction Base Object Properties					
Name	Type	Description		Reference	Validation
Amount	String	Principle Transaction Amount	→M ←M		Regular Expression – please refer to Swagger definition
Currency	String	Currency of the principal transaction amount.	→M ←M		Enumeration = ISO Currency Codes

Mobile Money API 1.0.0 Specification Description

Type	String	The harmonised Transaction Type	→M ←M		Enumeration = Transaction Types
Sub-type	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		
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 the SMS and on the account statement.	→O ←O		
Request Date	DateTime	The creation date and time of the transaction as supplied by the client.	→M ←M		
Date Created	DateTime	Date and time when the transaction was created by the API Provider	→NA ←O		
Date Modified	DateTime	Date and time when the transaction was modified by the API Provider	→NA ←O		
Transaction Reference	String	Unique reference for the transaction. This is returned in the response by API provider.	→NA ←M		
Transaction Receipt	String	Transaction receipt number as notified to the parties. This may differ from the Transaction Reference.	→NA ←O		
Requesting Organisation Transaction Reference	String	A reference provided by the requesting organisation that is to be associated with the transaction.	→O ←O		
One Time Code	String	A one-time code that can be supplied in the request or can be generated in the response depending upon the use case.	→O ←O		
Geo Code	String	Indicates the geographic location from where the transaction was initiated.	→O ←O		
Debit Party Identifier	Reference Array	A collection of key/value pairs that enable the debit	→M ←M	Account Identifiers	

Mobile Money API 1.0.0 Specification Description

		party to be identified. Keys include MSISDN and Wallet Identifier.			
Credit Party Identifier	Reference Array	A series of key/value pairs that enable the credit party to be identified. Keys include MSISDN and Wallet Identifier.	→M ←M	Account Identifiers	
Sender KYC Information	Reference	A collection of properties detailing the KYC of the transaction Sender, typically used for International Transfers.	→O ←O	KYC Information	
Recipient KYC Information	Reference	A collection of properties detailed the KYC of the transaction Recipient, typically used for International Transfers.	→O ←O	KYC Information	
International Transfer Information	Reference	A collection of properties detailed information specifically used for international transfers.	→O ←O	International Transfer Information	
Original Transaction Reference	String	For reversals and refunds, this property indicates the transaction which is the subject of the reversal	→O ←O		
Servicing Identity	String	The property is used to identify the servicing identity for 'present' transactions, e.g. till, POS ID, assistant ID	→O ←O		
Requesting LEI	String	Legal Entity Identifier of the organisation that is requesting the transaction.	→O ←O		Length = 20, Regular Expression (See Swagger Definition)
Receiving LEI	String	Legal Entity Identifier of the organisation that is receiving the transaction.	→O ←O		Length = 20, Regular Expression (See Swagger Definition)
Metadata	Reference Array	A collection of key/value pairs. These can be used to populate additional transaction properties.	→O ←O	Metadata	

4.2 Reversals API

The Reversals API is used to reverse a financial transaction. The originating transaction reference must be provided in the URI in order to identify the transaction to be reversed. For a partial reversal, the amount needs to be supplied. It should be noted however that API

Mobile Money API 1.0.0 Specification Description

Providers may not support partial reversals and will return an error if a partial amount is supplied. This API can also be used for adjustments.

Note: That only reversal creation is supported. For viewing reversals, the Transactions API should be used. Format is ***/transactions/{Original Transaction Reference}/reversals***.

The object definition for Reversals is provided below:

Reversal Base Object Properties					
Name	Type	Description		Reference	Validation
Amount	String	Principle Transaction Amount	→O ←O		Regular Expression – please refer to Swagger definition
Currency	String	Currency of the principal transaction amount.	→O ←O		Enumeration = ISO Currency Codes .
Type	String	The harmonised Transaction Type	→M ←M		Enumeration = Transaction Types Note that only Reversals and Adjustments are supported.
Sub-type	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		
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 the SMS and on the account statement.	→O ←O		
Request Date	DateTime	The creation date and time of the transaction as supplied by the client.	→M ←M		
Date Created	DateTime	Date and time when the transaction was created by the API Provider	→NA ←O		
Date Modified	DateTime	Date and time when the transaction was modified by the API Provider	→NA ←O		

Mobile Money API 1.0.0 Specification Description

Transaction Reference	String	Unique reference for the transaction. This is returned in the response by API provider.	→NA ←M		
Transaction Receipt	String	Transaction receipt number as notified to the parties. This may differ from the Transaction Reference.	→NA ←O		
Geo Code	String	Indicates the geographic location from where the transaction was initiated.	→O ←O		Regular Expression – please refer to Swagger definition
Requesting Organisation Transaction Reference	String	A reference provided by the requesting organisation that is to be associated with the transaction.	→O ←O		
Debit Party Identifier	Reference 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	
Credit Party Identifier	Reference 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	
Original Transaction Reference	String	For reversals and refunds, this property indicates the transaction which is the subject of the reversal	→NA ←M		
Requesting LEI	String	Legal Entity Identifier of the organisation that is requesting the transaction.	→O ←O		Length = 20, Regular Expression (See Swagger Definition)
Receiving LEI	String	Legal Entity Identifier of the organisation that is receiving the transaction.	→O ←O		Length = 20, Regular Expression (See Swagger Definition)
Servicing Identity	String	The property is used to identify the servicing identity for 'present' transactions, e.g. till, POS ID, assistant ID	→O ←O		
Metadata	Reference Array	A collection of key/value pairs. These can be used to populate additional transaction properties.	→O ←O	Metadata	

Mobile Money API 1.0.0 Specification Description

4.3 Batch Transactions API

As the name implies, this API allows clients to submit batches of transactions in a single HTTP request. Batch processing is always asynchronous. Batch processing follows a simple state transition:

1. Client submits the batch for processing via a '**POST /batchtransactions**'.
2. The client will return the [RequestState](#) object indicating whether a callback will be provided or polling is required.
3. The API provider will parse the batch in order to determine whether the transactions are capable of being processed.
4. Once parsing has completed, the API provider will set the batch status in the batchtransactions object to '**created**'.
5. The client will be able to retrieve the batchtransactions object by invoking the URI provided by the [RequestState](#) object.
6. If errors are indicated, i.e. some of the transactions failed parsing, the client is able to retrieve the errors via '**GET /batchtransactions/rejections**'.
7. Depending upon the business process, the client (or another client) can approve the batch for posting by issuing a '**PATCH /batchtransactions**' in order to update the status to '**approved**'.
8. As per step 2, a [RequestState](#) object will be returned indicating whether a callback will be provided or polling is required.
9. The API provider will then post the transactions in the batch taking into account any scheduling considerations.
10. Once posting is completed, the API provider will set the batch status in the batchtransactions object to '**completed**'.
11. The client will be able to retrieve the batchtransactions object by invoking the URI provided by the [RequestState](#) object.
12. the client will also be able to retrieve a list of successful transaction completions '**/batchtransactions/completions**' and transaction failures '**/batchtransactions/rejections**'.

4.3.1 Batch Transactions API

As described above, this API enables clients to submit and approve a batch of transactions. The API allows transactions of multiple types to be include in a single batch. The following operations are supported:

- Submit a batch: 'POST /batchtransactions'
- Approve a batch: 'PATCH /batchtransactions/{Batch ID}'. The Batch Status needs to be set to 'approved'.
- View details regarding batch processing: 'GET /batchtransactions/{batch ID}'

Batch Transaction Object Properties					
Name	Type	Description		Reference	Validation
Batch Title	String	Client-provided title for the batch	→O ←O		

Mobile Money API 1.0.0 Specification Description

Batch Description	String	Client-provided description of the batch.	→O ←O		
Batch ID	String	Identifier for the Batch that is assigned by the API provider. This ID is used by the client on subsequent GET or PATCH operations.	→N/A ←M		
Batch Status	String	Indicates the status of the batch.	→O ←M		Enumeration = created, approved, completed
Processing Flag	Boolean	Indicates whether the batch is currently undergoing processing by the API Provider.	→O ←O		
Scheduled Start Date	Datetime	If the batch has been scheduled, the expected start time is provided here.	→O ←O		
Created Date	Datetime	Indicates when the batch was created as recorded by the API provider.	→O ←M		
Approved Date	Datetime	Indicates when the batch was approved as recorded by the API provider.	→O ←M		
Completed Date	Datetime	Indicates when the batch was completed as recorded by the API provider.	→O ←M		
Rejections Count	Integer	Indicates the number of records that have been rejected, either during parsing or during final processing.	→O ←O		
Parsing Success Count	Integer	Indicates the number of records that have been parsed successfully.	→O ←O		
Completed Count	Integer	Indicates the number of records that have been successful completed.	→O ←O		
Transaction Data	Reference Array	Collection of Transactions that are to be processed. Note that the representation of each completed transaction can be retrieved via the '/completions API.	→M ←N/A	Transactions	

Mobile Money API 1.0.0 Specification Description

4.3.2 Batch Rejections API

As described above, this API enables clients to retrieve information on all transactions that have either failed parsing or have failed to be completed. Only GET is supported. Format is '*batchtransactions/{Batch ID}/rejections*'

In order 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 10 to 60.
fromDateTime	String	DateTime	Indicates the minimum date for which records should be returned.
toDateTime	String	DateTime	Indicates the maximum date for which records should be returned.

Note: That HTTPResponse metadata is returned with each response that is paginated indicating the total number of records available and total number of records returned.

Batch Rejection Object Properties					
Name	Type	Description		Reference	Validation
Transaction Reference	String	Transaction Reference as assigned by the API provider.	→N/A ←O		
Date Rejected	Datetime	Date and time of the rejection.	→N/A ←M		
Debit Party Identifier	Reference Array	The debit party identifiers for the transaction as specific in the batch request.	→N/A ←M	Account Identifiers	
Credit Party Identifier	Reference Array	The credit party identifiers for the transaction as specific in the batch request.	→N/A ←M	Account Identifiers	
Rejection Reason	String	The reason for the transaction request as	→N/A ←M		

Mobile Money API 1.0.0 Specification Description

		indicated by the API provider.			
Requesting Organisation Transaction Reference	String	A reference provider by the requesting organisation that is to be associated with the transactions	→N/A ←O		

4.3.3 Batch Completions API

This API lists all transactions that have successfully completed for a given batch. Only GET is supported. Format is '*batchtransactions/{Batch ID}/completions*'.

In order 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 10 to 60.
fromDateTime	String	DateTime	Indicates the minimum date for which records should be returned.
toDateTime	String	DateTime	Indicates the maximum date for which records should be returned.

Note: That HTTPResponse metadata is returned with each response that is paginated indicating the total number of records available and total number of records returned.

Batch Completion Object Properties					
Name	Type	Description		Reference	Validation
Transaction Reference	String	Transaction Reference as assigned by the API provider.	→N/A ←M		
Date Completed	Datetime	Date and time indicating when the transaction was completed.	→N/A ←M		
Link	String	Provides a URL to the transaction resource.	→N/A ←M		

Mobile Money API 1.0.0 Specification Description

Debit Party Identifier	Reference Array	The debit party identifiers for the transaction as specific in the batch request.	→N/A ←M	Account Identifiers	
Credit Party Identifier	Reference Array	The credit party identifiers for the transaction as specific in the batch request.	→N/A ←M	Account Identifiers	
Requesting Organisation Transaction Reference	String	A reference provider by the requesting organisation that is to be associated with the transactions	→N/A ←O		

4.4 Accounts APIs

The Accounts APIs are used to view properties associated with an account resource. Types of accounts include mobile wallets, financial institution accounts and utility accounts (e.g. for electricity).

4.4.1 Identifying a Target Account

Two methods are providing for identifying an account, the multiple identifiers method and the MSISDN identifier method.

Multiple Identifiers Method

There is no single industry-accepted method of uniquely identifying an account. There are numerous methods of identifying an account and the list of permitted identifiers can be found in the [Account Identifiers](#) section. Every Account API identifies the target account through the URI. As there can be multiple identifiers required to identify the target account, the URI uses a '\$' delimiter to separate each identifier. The format can be expressed as:

/accounts/{accountIdentifier1}@{value1}\${accountIdentifier2}@{value2}\${accountIdentifier3}@{value3}

MSISDN Identifier Method

In the scenario where MSISDN is the only identifier needed to uniquely identify an account, an alternate short URI is available: ***/accounts/msisdn/{value}***

4.4.2 Supported Account Operations

The Accounts object can support various operations. A list of supported account resources is listed below:

- ***/accounts/{Account Identifiers}/status***. Returns the current status for an account. See the [Account Status API](#) for more information.
- ***/accounts/{Account Identifiers}/accountname***. Returns all name properties held for the primary identity that is associated with the account. See the [Account Name API](#) for more information.

Mobile Money API 1.0.0 Specification Description

- **/accounts/{Account Identifiers}/balance.** Returns the balances for the account. See the [Account Balances](#) API for more information.
- **/accounts/{Account Identifiers}/statemententries.** Returns all statement entries for a given account. See the [Statement Entries](#) API for more information.
- **/accounts/{Account Identifiers}/bills.** Returns all outstanding bills for a given account. See section the [Bills](#) API for more information.
- **/accounts/{Account Identifiers}/debitmandates.** Allows the creation, updating and viewing of debit mandates for a given account. See [Debit Mandates](#) API for more information.
- **/accounts/{Account Identifiers}/links.** Allows the creation, updating and viewing of account to account links for a given account. See [Links](#) API for more information.

4.4.3 Returning Transactions for an Account

It is possible to return a range of transactions for an account as per the following format:

/accounts/{Account Identifiers}/transactions.

In order 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 10 to 60.
fromDateTime	String	DateTime	Indicates the minimum date for which records should be returned.
toDateTime	String	DateTime	Indicates the maximum date for which records should be returned.

Note 1: That all transactions will be returned in descending date created order.

Note 2: That metadata is returned with each response that is paginated indicating the total number of records available.

4.4.4 Accounts Status API

The Accounts Status API returns a harmonised status of the account. The status enables the client to determine whether transactions can be subsequently posted against the account. URI format is **'/accounts/{Account Identifiers}/status'**.

Account Status Object Properties				
Name	Type	Description	Reference	Validation

Mobile Money API 1.0.0 Specification Description

Account Status	String	Indicates a simplified representation of the account status. 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 posted with the account identifier(s) will result an unregistered voucher creation.	→NA ←M		Enumeration = available, unavailable, unregistered
Account Sub-Status	String	Property can be used to return a provider-specific status for the account.	→NA ←O		
LEI	String	Indicates the Legal Entity Identifier of the organisation holding the account.	→NA ←O		Length = 20, Regular Expression (See Swagger Definition)

4.4.5 Account Balances API

This API defines specific properties for returning balances associated with an account. URI format is `'/accounts/{Account Identifiers}/balance'`.

Balance Object Properties					
Name	Type	Description		Reference	Validation
Current Balance	String	The current outstanding balance on the account.	→NA ←O		Regular Expression – please refer to Swagger definition
Available Balance	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		Regular Expression – please refer to Swagger definition
Reserved Balance	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		Regular Expression – please refer to Swagger definition
Uncleared Balance	String	Indicates the sum of uncleared funds in an account, i.e. those that are awaiting a credit confirmation.	→NA ←O		Regular Expression – please refer to Swagger definition
Currency	String	Currency for all returned balances.	→NA ←O		Enumeration = ISO Currency Codes
Account Status	String	Indicates a simplified representation of the account	→NA		Enumeration = available,

Mobile Money API 1.0.0 Specification Description

		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 an unregistered voucher creation.	←O		unavailable, unregistered
--	--	---	----	--	---------------------------

4.4.6 Account Name API

This API defines specific properties for returning account holder name information associated with an account. URI format is ***/accounts/{Account Identifiers}/accountname***.

Account Name Object Properties					
Name	Type	Description		Reference	Validation
Account Name	Reference	A collection of properties detailing the name of the Primary Account Holder.	→NA ←O	Name	
Account Status	String	Indicates a simplified 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 an unregistered voucher creation.	→NA ←M		Enumeration = available, unavailable, unregistered
LEI	String	Indicates the Legal Entity Identifier of the organisation holding the account.	→NA ←O		Length = 20, Regular Expression (See Swagger Definition)

4.4.7 Statement Entries API

The Statement Entries API enables generic representations of transactions to be returned. Typically, the returned representations are used for the purposes of presenting a statement to the account holder. In order to return a statements, an account or a transaction must be specified. The URI format is as follows:

/accounts/{Account Identifiers}/statemententries

Mobile Money API 1.0.0 Specification Description

In order to filter the number of records returned, the following query string parameters 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 10 to 60.
fromDateTime	String	DateTime	Indicates the minimum date for which records should be returned.
toDateTime	String	DateTime	Indicates the maximum date for which records should be returned.

Note 1: That all statement entries will be returned in descending date created order.

Note 2: That metadata is returned with each response that is paginated indicating the total number of records available.

Note 3: That it is also possible to retrieve an individual statement entry as per the following: ***/statemententries/{Transaction Reference}***

Only GET (read) operations are supported for statement entries.

Statement Entries Object Properties					
Name	Type	Description		Reference	Validation
Amount	String	Requested transaction amount.	→NA ←M		Regular Expression – please refer to Swagger definition
Currency	String	Currency of the requested transaction amount.	→NA ←M		Enumeration = ISO Currency Codes
Display Type	String	The transaction type that is to be used for presentation to the account holder as determined by the API provider. This is not necessarily the actual transaction type.	→NA ←O		
Transaction Status	String	Indicates the status of the transaction as represented by the API provider.	→NA ←M		
Description Text	String	Free format text description of the transaction provided by	→NA ←O		

Mobile Money API 1.0.0 Specification Description

		the client. This can be provided as a reference for the receiver on the SMS and on the account statement.			
Request Date	DateTime	The creation date and time of the transaction as supplied by the client.	→NA ←O		
Date Created	DateTime	Date and time when the transaction was created by the API Provider	→NA ←O		
Date Modified	DateTime	Date and time when the transaction modified by the API Provider	→NA ←O		
Transaction Reference	String	Unique reference for the transaction. This is returned in the response by API provider.	→NA ←M		
Transaction Receipt	String	Transaction receipt number as notified to the parties. This may differ from the Transaction Reference.	→NA ←O		
Debit Party Identifier	Reference Array	A collection of key/value pairs that identify the debit. Keys include MSISDN and Wallet Identifier.	→NA ←M	Account Identifiers	
Credit Party Identifier	Reference Array	A series of key/value pairs that identify the credit party. Keys include MSISDN and Wallet Identifier.	→NA ←M	Account Identifiers	

4.5 Bills API

The Bills API is used to return all outstanding bills associated with an account. The main purpose of the object is to support Bill Presentment, i.e. presenting all applicable bills for a payer to view and select for payment. In order to pay a bill, the [Bill Payments API](#) is used. The URI format is as follows – **‘/accounts/{Account Identifiers}/bills’**. In the scenario where MSISDN is the only identifier needed to uniquely identify an account, an alternate short URI is available – **‘/accounts/msisdn/{value}’**

Only GET (read) operations are permitted for the Bills object.

Bill Object Properties					
Name	Type	Description		Reference	Validation
Currency	String	Currency of the bill to be paid.	→NA ←O		Enumeration = ISO Currency Codes
Amount Due	String	Amount outstanding on the bill to be paid.	→NA ←O		Regular Expression – please refer to Swagger definition

Mobile Money API 1.0.0 Specification Description

Due Date	Date	Date on which the Bill is due to be paid.	→NA ←O		
Bill Reference	String	Reference number for the Bill that this payer can use when he/she wishes to pay.	→NA ←O		
Minimum Amount Due	String	The minimum amount that is outstanding on the bill to be paid.	→NA ←O		Regular Expression – please refer to Swagger definition
Bill Description	String	Description of the bill that is to be paid.	→NA ←O		
Metadata	Reference Array	A collection of key/value pairs. These can be used to return additional information regarding the bill.	→NA ←O	Metadata	

4.6 Bills Payments API

The Bills Payments API is used to pay a specific bill associated with an account. There is a choice of URI format as per below:

- **'/accounts/{Account Identifiers}/bills/{Bill Reference}/payments'**. Full method of identifying a bill for which the payment is to be made.
- In the scenario where MSISDN is the only identifier needed to uniquely identify the bill account, use – **'/accounts/msisdn/{value}/bills/{Bill Reference}/payments'**.
- In the scenario where the Bill Payment reference can be used in isolation to uniquely identify the bill, use **'/bills/{Bill Reference}/payments'**.

Only POST is supported for this API.

Bill Payment Object Properties					
Name	Type	Description		Reference	Validation
Currency	String	Currency of the amount that is being paid.	→M ←M		Enumeration = ISO Currency Codes
Paid Amount	String	Amount that is being paid.	→M ←M		
Customer Reference	Date	Textual reference provided by the customer paying the bill	→O ←O		

Mobile Money API 1.0.0 Specification Description

Supplementary Bill Reference Details	Reference Array	In some cases, a single reference is not sufficient to identify a bill. This key-value collection enables further reference information to be supplied.	→O ←O	Supplementary Bill References	
--------------------------------------	-----------------	---	----------	---	--

4.7 Debit Mandates API

The Debit Mandates API is used to enable a mobile money customer to provide prior approval for payments to be taken from their account by the indicated payee. If the amount property is not supplied, the mandate is considered open, i.e. the payer would be able to take any amount. Due to the need to obtain explicit payer approval, requests for mandates are typically asynchronous in nature. Mandates can be created, changed and inactivated. The URI format is as follows:

- Creation: *POST /accounts/{Account Identifiers}/debitmandates*.
- Update: In order to update a debit mandate, a HTTP PATCH is used. Format is: *PATCH /accounts/{Account Identifiers}/debitmandates/{Mandate Reference}*
- Read. *GET /accounts/{Account Identifiers}/debitmandates/{Mandate Reference}*.

4.7.1 Debit Mandates Object

Debit Mandate Object Properties					
Name	Type	Description		Reference	Validation
Currency	String	Currency of the principal transaction amount.	→O ←O		Enumeration = ISO Currency Codes
Amount Limit	String	The maximum amount that can be taken by the Payee on a payment request	→O ←O		
Start Date	Date	Date on which the first payment is to be taken.	→M ←M		
End Date	Date	Date on which the final payment is to be taken.	→O ←O		
Number of Payments	Number	Indicates the number of consecutive payments that are to be taken.	→O ←O		
Frequency Type	String	Indicates the frequency for which payments will be taken from the payers account.	→O ←O		Enumeration = Frequency
Mandate Status	String	Indicates the status of the Mandate as held in the API Provider system	→O ←O		Enumeration = active, inactive

Mobile Money API 1.0.0 Specification Description

Mandate Reference	String	Unique reference provided by the API Provider for the mandate.	→O ←M		
Request Date	DateTime	The creation date and time of the transaction as supplied by the client.	→M ←M		
Date Created	DateTime	Date and time when the debit mandate was created by the API Provider	→NA ←O		
Date Modified	DateTime	Date and time when the debit mandate was modified by the API Provider	→NA ←O		

4.8 Links API

The Links API is used to establish a link between two separate accounts on the client and provider's systems. The API can be used for example to link a mobile wallet account to an MFI 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.

In order to identify the accounts that are to be linked, the target account is specified in the URI whereas the source account is specified in the link object.

The URI format is as follows:

- Creation: *POST /accounts/{Target Account Identifiers}/links.*
- Update: In order to update a Link (status and/or mode), a HTTP PATCH is used. Format is: *PATCH /accounts/{Target Account Identifiers}/links/{Link Reference}.*
- Read. *GET /accounts/{Target Account Identifiers}/links/{Link Reference}.*

4.8.1 Link Object

Link Object Properties					
Name	Type	Description		Reference	Validation
Link Reference	String	Indicates the Link reference. This enables a linked account to be uniquely identified.	→NA ←M		
Link Status	String	Indicates the status of the Link.	→M ←M		Enumeration = active, inactive

Mobile Money API 1.0.0 Specification Description

Link Mode	String	Indicates the mode of operation support for the Link. If not populated, then default value it 'Both'.	→O ←O		Enumeration = push, pull, both
Source Account Identifier	Reference Array	A series of key/value pairs that identify the source account. Keys include MSISDN and Wallet Identifier.	→M ←M	Account Identifiers	

4.9 Quotations API

The quotations API is used to obtain one or multiple quotes for a mobile money customer that wishes to send money internationally. The production of a quote typically involves obtaining the Forex rate and any additional fees that will be levied on the sending customer. Normally a request is made for a quotation by the requesting mobile money system (also known as the Sending Service Provider) in response to a customer request. The quotation is then provided to an IMT hub (also known as a Wholesale Service Provider) which is typically responsible for performing sender (and possibly receiver) AML checks as well as obtaining a Forex rate. The quotation is calculated and returned to the requesting mobile money system. If the customer is satisfied with the quotation, then he will initiate an International Transfer transaction.

The quotations object supports:

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

Quotation Object Properties					
Name	Type	Description		Reference	Validation
Quotation Reference	String	Unique reference for the Quotation as provided by the API Provider	→NA ←M		
Quotation Status	String	Indicates the creation state of the quotation.	→NA ←O		Enumeration = pending, rejected, completed
Request Date	DateTime	The creation date and time of the transaction as supplied by the client.	→M ←M		
Date Created	DateTime	Date and time when the transaction was created by the API Provider	→NA ←O		
Date Modified	DateTime	Date and time when the transaction was modified by the API Provider	→NA ←O		

Mobile Money API 1.0.0 Specification Description

Debit Party Identifier	Reference 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	
Credit Party Identifier	Reference 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	
Sender KYC	Reference	A collection of properties detailing the KYC of the transaction Sender, typically used for International Transfers.	→O ←O	KYC Information	
Recipient KYC	Reference	A collection of properties detailed the KYC of the transaction Recipient, typically used for International Transfers.	→O ←O	KYC Information	
Request Amount	String	Requested quotation amount.	→M ←M		
Request Currency	String	Currency of the requested quotation amount.	→M ←M		Enumeration = ISO Currency Codes
Chosen Delivery Method	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
Available Delivery Methods	String Array	Delivery Methods that are possible for the intended recipient.	→NA ←O		Enumeration = Delivery Method
Quotes	Reference Array	A collection of quotes. A quote can be received from a single receiving payment service provider or from multiple providers.	→O ←O	Quotes	
Sender Blocking Reason	String	The reason for blocking the quotation, based on AML checks on the sender	→O ←O		
Recipient Blocking Reason	String	The reason for blocking the quotation, based on AML checks on the recipient	→O ←O		
Metadata	Reference Array	A collection of key/value pairs. These can be used	→O ←O	Metadata	

Mobile Money API 1.0.0 Specification Description

		to populate additional quotation properties.			
--	--	--	--	--	--

5 Supporting Objects

5.1 International Transfer Information Object

The International Transfer Information object contains details that are specific to international transfers.

International Transfer Information Object Properties					
Name	Type	Description		Reference	Validation
Country of Origin	String	The originating country of the transaction, i.e. the country where the transaction commenced.	→M ←M		Enumeration = ISO Country Codes
Quotation Reference	String	Reference for the quotation that was provider to the sender. (refer to Quotations API for more information).	→O ←O		
Quote ID	String	The specific quote associated with the quotation (refer to Quotes object for more information).	→O ←O		
Receiving Country	String	Destination Country of international remittance	→O ←O		
Purpose of Remittance	String	Property providing a description of the reason for the international remittance.	→O ←O		
Relationship with Sender	String	Indicates the relationship (if any) between the sender and the receiver	→O ←O		
Delivery Method to Use	String	The recipient delivery method as chosen by the sender	→O ←O		Enumeration = Delivery Method Types
Sender Blocking Reason	String	The reason for blocking the transaction, based on AML checks on the sender	→NA ←O		
Recipient Blocking Reason	String	The reason for blocking the transaction, based on AML checks on the recipient	→NA ←O		

5.2 KYC Information Object

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

KYC Information Object Properties					
Name	Type	Description		Reference	Validation
Nationality	String	Nationality of the KYC subject.	→0 ←0		Enumeration = ISO Country Codes
Date of Birth	Date	Birth date of the KYC subject.	→0 ←0		
Occupation	String	Occupation of the KYC subject.	→0 ←0		
Employer Name	String	Employer Name of the KYC subject.	→0 ←0		
Contact Phone	String	Contact phone number (mobile or landline) of the KYC subject. Phone number to be provided in international format as per ITU E.123.	→0 ←0		Regular Expression to validate against ITU E.123 Refer to Swagger definition for more information.
Gender	String	Gender of the KYC Object.	→0 ←0		Length=1, Enumeration = (m)ale, (f)emale, (u)nspecified
Id Document	Reference Array	An array of properties containing the forms of identification that are associated with the subject.	→0 ←0	Id Document	
Postal Address	Reference	A collection of properties that details the postal address of the KYC subject.	→0 ←0	Address	
KYC Subject Name	Reference	Refers to the name properties for the KYC subject	→0 ←0	Name	
Email Address	String	Email address of the KYC subject	→0 ←0		
Birth Country	String	The country of birth of the KYC subject	→0 ←0		Enumeration = ISO Country Codes

5.3 Name Object

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

Name Object Properties					
Name	Type	Description		Reference	Validation
Title	String	The given title of the KYC subject, e.g. Mr, Mrs, Dr.	→0 ←0		
First Name	String	First name (also referred to as given name) of the KYC subject.	→0 ←0		
Middle Name	String	Middle Name of the KYC subject.	→0 ←0		
Last Name	String	Surname (also referred to as last or family name) of the KYC subject.	→0 ←0		
Full Name	String	The full name of the KYC subject	→0 ←0		
Native Name	String	The full name expressed as in the native language	→0 ←0		

5.4 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 Properties					
Name	Type	Description		Reference	Validation
ID Type	String	Indicates the type of identification for the KYC subject, e.g. passport, driving licence etc..	→M ←M		Enumeration = ID Types
ID Number	String	Reference pertaining to the type of identification for the KYC subject	→0 ←0		
Issue Date	Date	Date of issue for the identification document	→0 ←0		
Expiry Date	Date	Date of expiry for the identification document	→0 ←0		
Issuer	String	Indicates the organisation/government entity that issued the ID document.	→0 ←0		

Mobile Money API 1.0.0 Specification Description

Issuing Place	String	Place of issue for the identification type.	→O ←O		
Issuing Country Code	String	Country where the identification type was issued.	→O ←O		Enumeration = ISO Country Codes
Other ID Description	String	Where an ID Type of 'otherid' is specified, a description of the type of identification can be provided in this property.	→O ←O		

5.5 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 Properties					
Name	Type	Description		Reference	Validation
Address Line 1	String	First line of the address.	→O ←O		
Address Line 2	String	Second line of the address.	→O ←O		
Address Line 3	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

5.6 Account Identifiers Object

In Mobile Money, there is no single and common method for identifying mobile money accounts and/or transaction parties. Identifiers include MSISDN (Mobile Number), Bank Short Code, Account Number, Agent/Merchant Short Code and Wallet Identifier. 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 Properties					
Name	Type	Description		Reference	Validation

Mobile Money API 1.0.0 Specification Description

Key	String	Provides the account identifier type.	→M ←M		Enumeration = Account Identifiers
Value	String	Provides the account identifier type value.	→M ←M		

5.7 Quotes Object

The quotes object defines the properties associated with international remittance quotes.

Quotes Object Properties					
Name	Type	Description		Reference	Validation
QuoteID	String	The unique ID for this quote	→NA ←M		
QuoteExpiryTime	DateTime	The timestamp when the quote will expire	→NA ←O		
Receiving Service Provider	String	The name of the RSP, i.e. the provider that the quote is associated with.	→NA ←O		
Sending Amount	String	Requested quotation amount as supplied by the sender.	→NA ←M		Regular Expression – please refer to Swagger definition
Sending Currency	String	Currency of the requested quotation amount.	→NA ←M		Enumeration = ISO Currency Codes
Receiving Amount	String	The total amount as it will be received by the receiving end user.	→NA ←M		Regular Expression – please refer to Swagger definition
Receiving Currency	String	The currency of the quote.	→NA ←M		Enumeration = ISO Currency Codes
FXRate	String	The conversion rate applicable between the sending and the receiving currency for the requested transaction	→NA ←M		Regular Expression – please refer to Swagger definition
Delivery Method	String	The delivery method that is applicable to the quotation	→NA ←O		Enumeration = Delivery Method Object

Mobile Money API 1.0.0 Specification Description

5.8 Metadata Object

The metadata object allows additional properties to be specified for the parent object in the form of key/value pairs. Additional properties should only be used where no suitable defined property match can be found. The number of key/value pairs is limited to 20.

Identifier Object Properties					
Name	Type	Description		Reference	Validation
Key	String	Identifies the type of additional property.	→M ←M		
Value	String	Identifies the value of the additional property.	→M ←M		

5.9 Supplementary Bill References Object

This object enables additional payment references to be specified for a bill payment in the form of key/value pairs. Additional properties should only be used where no suitable defined property match can be found. The number of key/value pairs is limited to 20.

Identifier Object Properties					
Name	Type	Description		Reference	Validation
Payment Reference Type	String	Identifies the type of the additional payment reference.	→M ←M		
Payment Reference Value	String	Identifies the value of the additional payment reference.	→M ←M		

6 Enumerations

6.1 ISO Currency Codes

The three-character alphabetic code for currency as defined by ISO 4217 is to be used for all currency properties. 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>

6.2 Transaction Types

A small number of types have been defined to classify the nature of a transaction. Use of these types will enable clients to indicate the type of transaction in a manner that is common regardless of the API provider.

Code	Description
billpay	Payment of bill from a business for goods and/or services.
deposit	Exchange of cash in return for e-Money either at a physical agent or via ATM
disbursement	Disbursement of funds (making payments from an organisation (business, NGO, government entity) to a mobile money recipient.
transfer	Transfer of funds between mobile money provider and another provider or financial institution in the same country.
merchantpay	Purchases of goods and/or services from shops (payer present) or online (payer not present).
intrtransfer	Transfer of funds to a recipient in another country, either directly to/from a mobile wallet or via an international money transfer provider.
adjustment	General adjustments to an account via an adjustment transaction (e.g. refunds).
reversal	Reversal of a prior transaction to return funds to the payer.
withdrawal	Exchange of e-Money in return for cash either at a physical agent or via ATM

6.3 ID Types

The ID Types enumeration lists accepted identification types. Due to the wide international variation in accepted types of identification, a catch-all type of 'OtherID' has also been defined.

ID Type	Description
passport	Payment of bill from a business for goods and/or services.
nationalregistration	National Registration Number
otherid	Catch-all for IDs not on the list
drivinglicence	Driving Licence Number
socialsecurity	Social Security Number
alienregistration	Alien Registration ID

Mobile Money API 1.0.0 Specification Description

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

6.4 Account Identifiers

The Account Identifier enumeration lists all possible means to identify a target account and for transactions, the debit and/or credit party. 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 ranks 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	Mobile Number of the account holder. Should conform to ITU E.123 . Refer to Swagger

Mobile Money API 1.0.0 Specification Description

			definition for more information
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 Short 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.

6.5 ISO Country Codes

The two-character alphabetic code for country as defined by ISO 3166 is to be used for all properties 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

6.6 Delivery Method Type

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

Delivery Method	Description
-----------------	-------------

Mobile Money API 1.0.0 Specification 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

6.7 Frequency Type

When requesting a debit mandate, the API client is able to specific the frequency in which the payment should be taken. Valid values are defined in the table below.

Frequency Type	Description
weekly	Payment will be taken weekly
fortnight	Payment will be take every two weeks
monthspecificdate	Payment to be taken on a specific date every month
2months	Payment to be taken every two months
3months	Payment to be taken every three months
4months	Payment to be taken every four months
6months	Payment to be taken every six months
yearly	Payment to be taken yearly
lastdaymonth	Payment to be taken on the last calendar day of the month
lastdaymonthworking	Payment to be taken on the last working day of the month according to working days as per the resident country of the account.
lastmonday	Payment to be taken on the last Monday of the month
lasttuesday	Payment to be taken on the last Tuesday of the month
lastwednesday	Payment to be taken on the last Wednesday of the month
lastthursday	Payment to be taken on the last Thursday of the month
lastfriday	Payment to be taken on the last Friday of the month
lastsaturday	Payment to be taken on the last Saturday of the month
lastsunday	Payment to be taken on the last Sunday of the month
specificdaymonthly	Payment to be taken on a specific day of the month

7 API Behaviour & Error Handling

The ability to manage API state and handle exceptions in a consistent manner is central to the harmonised Mobile Money API. Consistency is achieved via the following:

- **Harmonised API State Model.** For synchronous requests, HTTP response codes are used exclusively to manage request state communication – both during the creation of a resource and updating of a resource. For asynchronous requests, a simple RequestState object is additionally used to manage request state communication.
- **Harmonised HTTP Status Codes.** Standardised HTTP status codes are associated with a harmonised state model for resource creation and update.
- **Harmonised Application-level Error Reason Codes.** Standardised error codes are returned in a standardised error object for ALL client-responsible and server-responsible errors. Error parameters allow more detailed provider-specific diagnostic information to be returned with the error.
- **API Heartbeat** enables API provider service availability to be established by clients on a regular basis.
- **Responses** object enables clients to re-request a missing API response.
- **Error Handling Guidelines** provide a standardised way of handling exceptions, including managing timeouts and retries.

7.1 Request States

States are represented by the use of HTTP response codes and also the RequestState object for asynchronous requests. Figure 8.1 illustrates the state model for synchronous requests. The state models for asynchronous requests is represented by the sequence flows in the [API Behavioural Model](#).

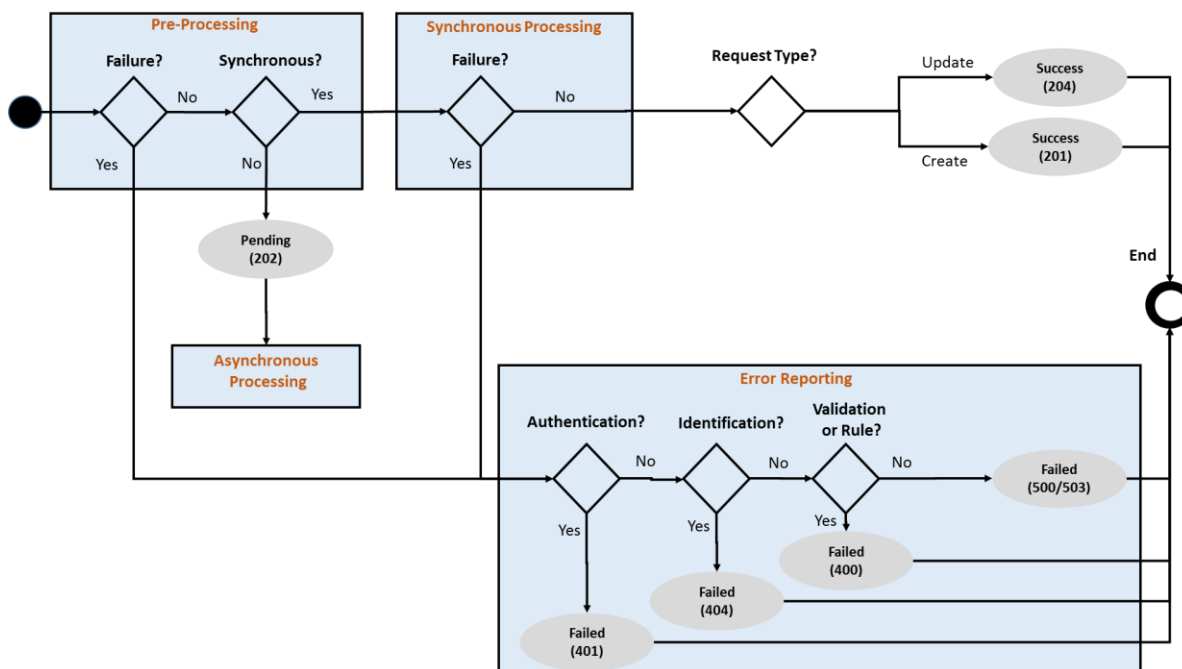


Figure 2: Synchronous Request States**7.2 API Behavioural Model****7.2.1 Overview**

API behaviour is governed by the following factors:

- The resource.
- The type of operation, i.e. create, update or read.
- Whether the provider will process the request synchronously.
- Whether the provider implements callback or polling methods for asynchronous processing.

When combined, these factors drive API behaviour from the simplest request (e.g. view an account balance) to a more complex request (e.g. update a debit mandate asynchronously).

7.2.2 Request State Object

Asynchronous flows involve either a callback or a polling mechanism to enable to client to determine the final state of the request. Both mechanisms involve the use of the RequestState object as per below:

- **Callback.** A request is initiated via a HTTP POST or PATCH request with an intermediate response represented by a RequestState object. Once the request has been completed, the provider will initiate a PATCH request to provide the client with the final RequestState.
- **Polling.** Is initiated by a HTTP POST or PATCH request with an intermediate response provided in the form of the RequestState object. A HTTP GET is then issued on the RequestState by the client at regular defined intervals until the final resource state is returned.

The object definition for RequestState is described below.

RequestState Object Properties					
Name	Type	Description		Reference	Validation
Server Correlation ID	String	A unique identifier issued by the provider to enable the client to identify the RequestState resource on subsequent polling requests.	→NA ←M		GUID
Status	String	Indicates the status of the request.	→NA ←M		Enumeration = pending, completed, failed
Pending Reason	String	A textual description that can be provided to	→NA ←O		

Mobile Money API 1.0.0 Specification Description

		describe the reason for a pending status.			
Notification Method	DateTime	Indicates whether a callback will be issued or whether the client will need to poll.	→NA ←M		Enumeration = callback, polling
Object Reference	Reference	Provides a reference to the subject resource, e.g. transaction reference.	→NA ←O		
Expiry Time	DateTime	Indicate the time by which the provider will fail the request if completion criteria have not been met. For an example, a debit party failing to authorise within the allowed time period.	→NA ←O		
Poll Limit	Integer	Indicates the number of poll attempts for the given requeststate resource that will be allowed by the provider.	→NA ←O		
ErrorReference	Reference	If the asynchronous processing failed, details of the error will be returned here	→NA ←O	Errors Object	

7.2.3 Sequence Flows

There are seven sequence flows that can be implemented over the mobile money API. These are reflected in the attached sheet below. The sheet indicates the necessary HTTP operations, request object, response object and expected success and failure response codes for each API service.



Mobile Money API
Request and Responses

7.2.4 API Flow Patterns

Sequence diagrams 8.2 through 8.8 illustrate the standard flow patterns for the mobile money API. Note that Green flows represent a success path and red flows represent a failure path. A solid line indicates a HTTP request and a dotted line indicates a HTTP response.

Mobile Money API 1.0.0 Specification Description

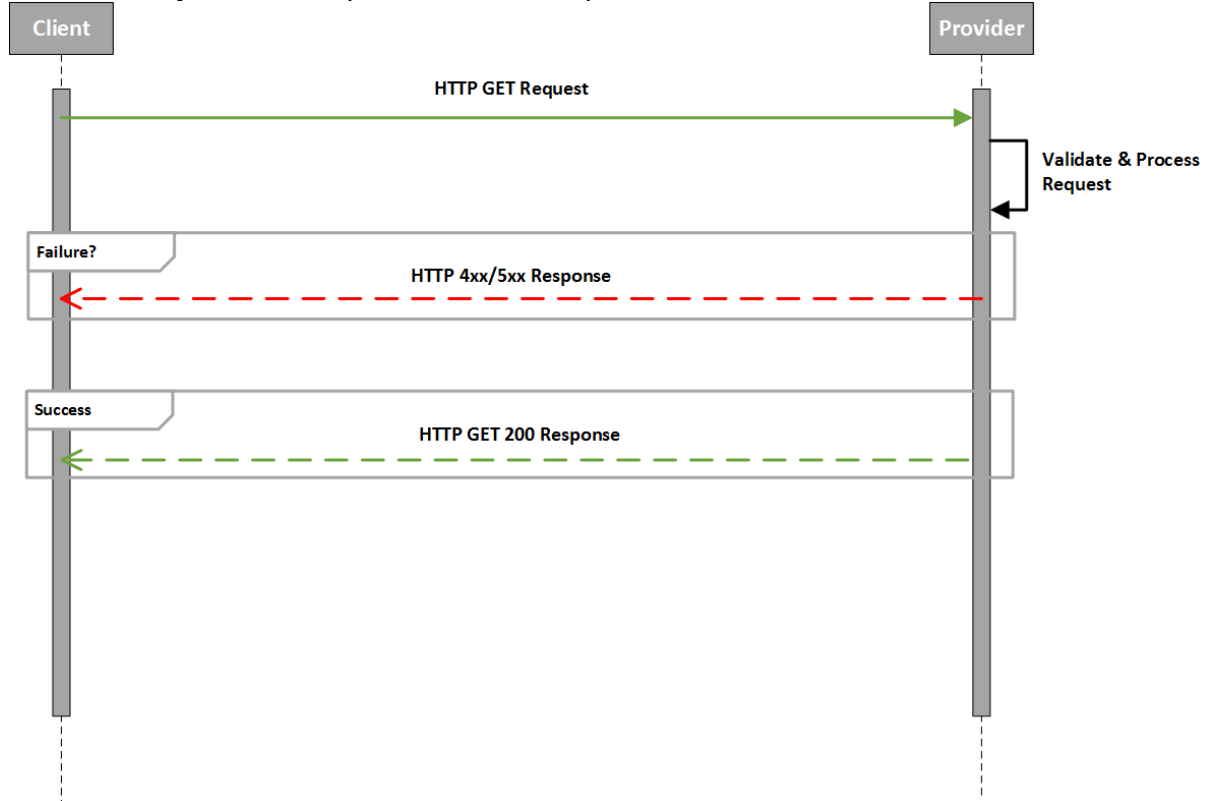


Figure 3: Read (GET) Flow Pattern

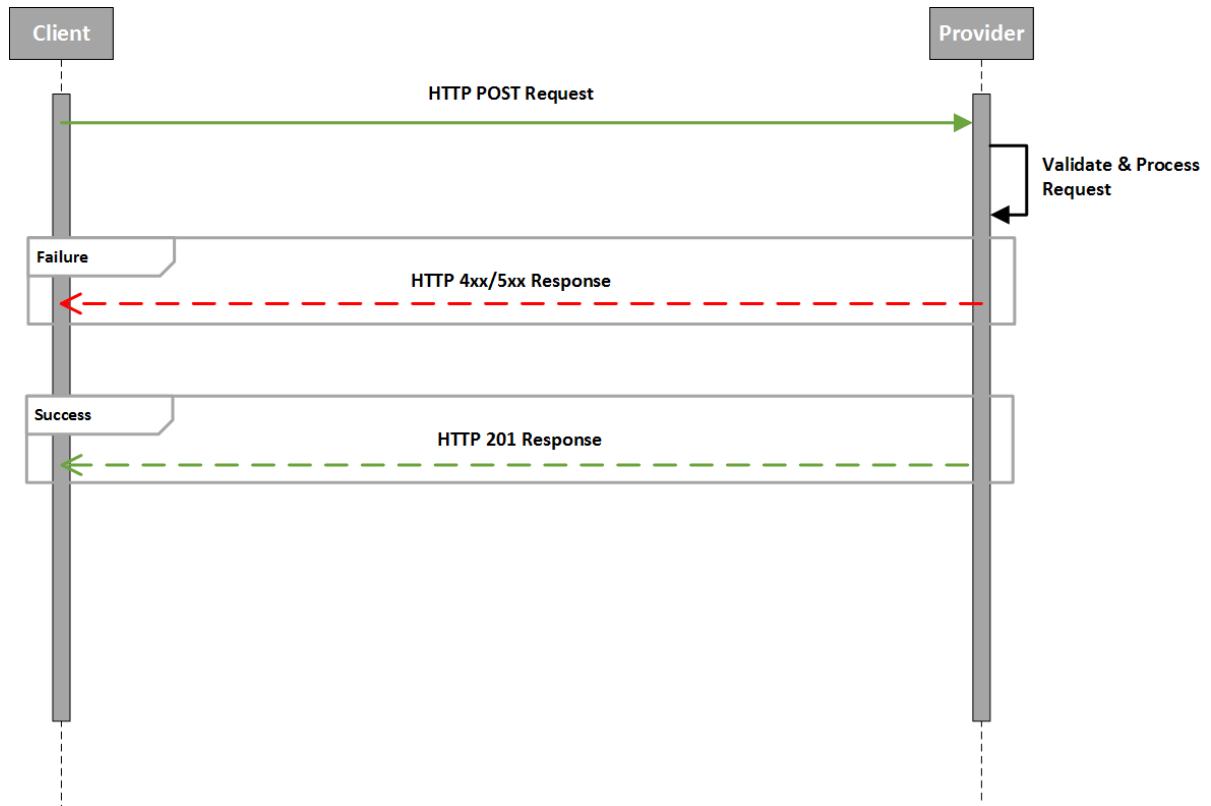


Figure 4: Create (POST) Synchronous Flow Pattern

Mobile Money API 1.0.0 Specification Description

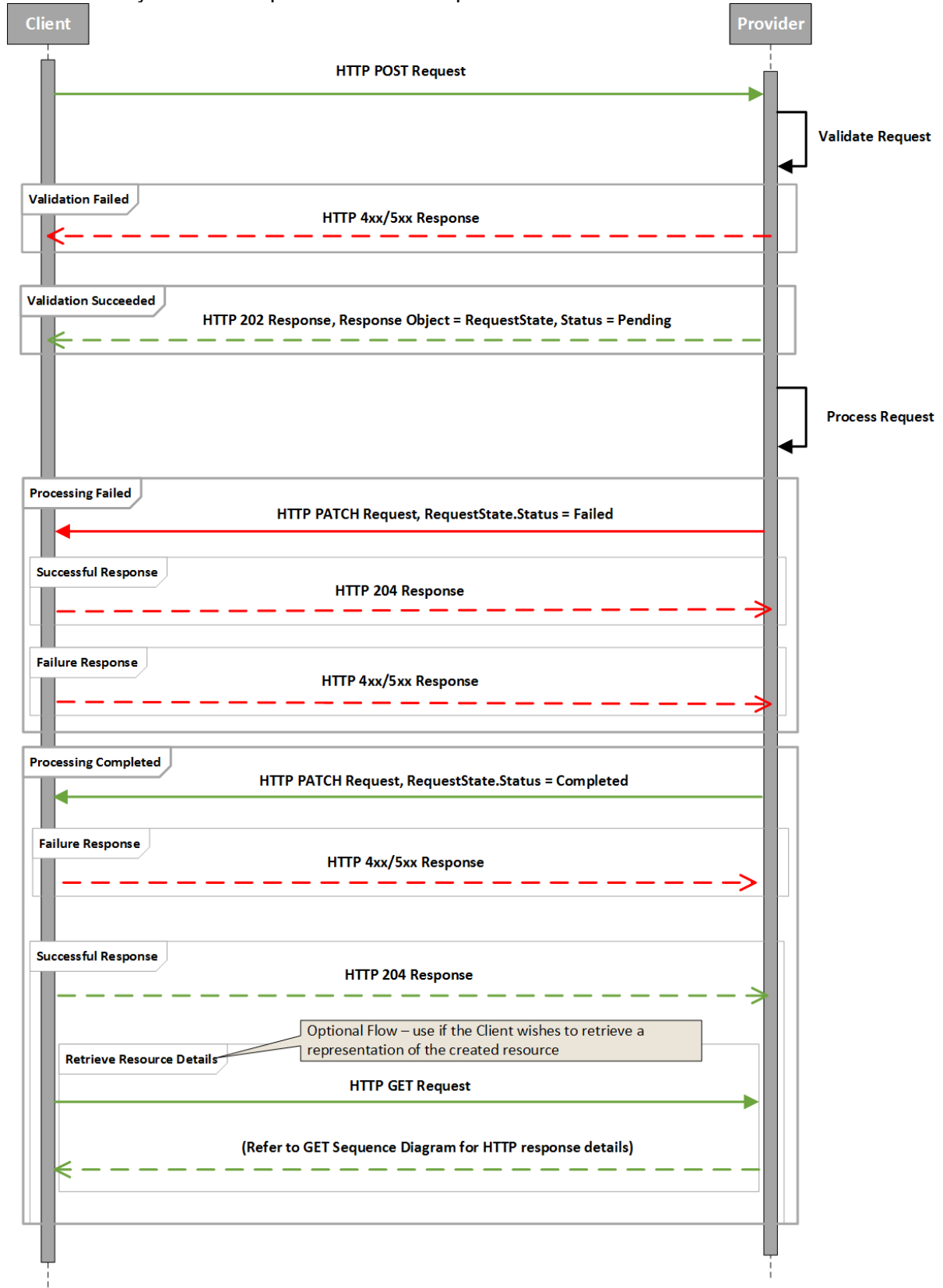


Figure 5: Create (POST) Callback Asynchronous Flow Pattern

Mobile Money API 1.0.0 Specification Description

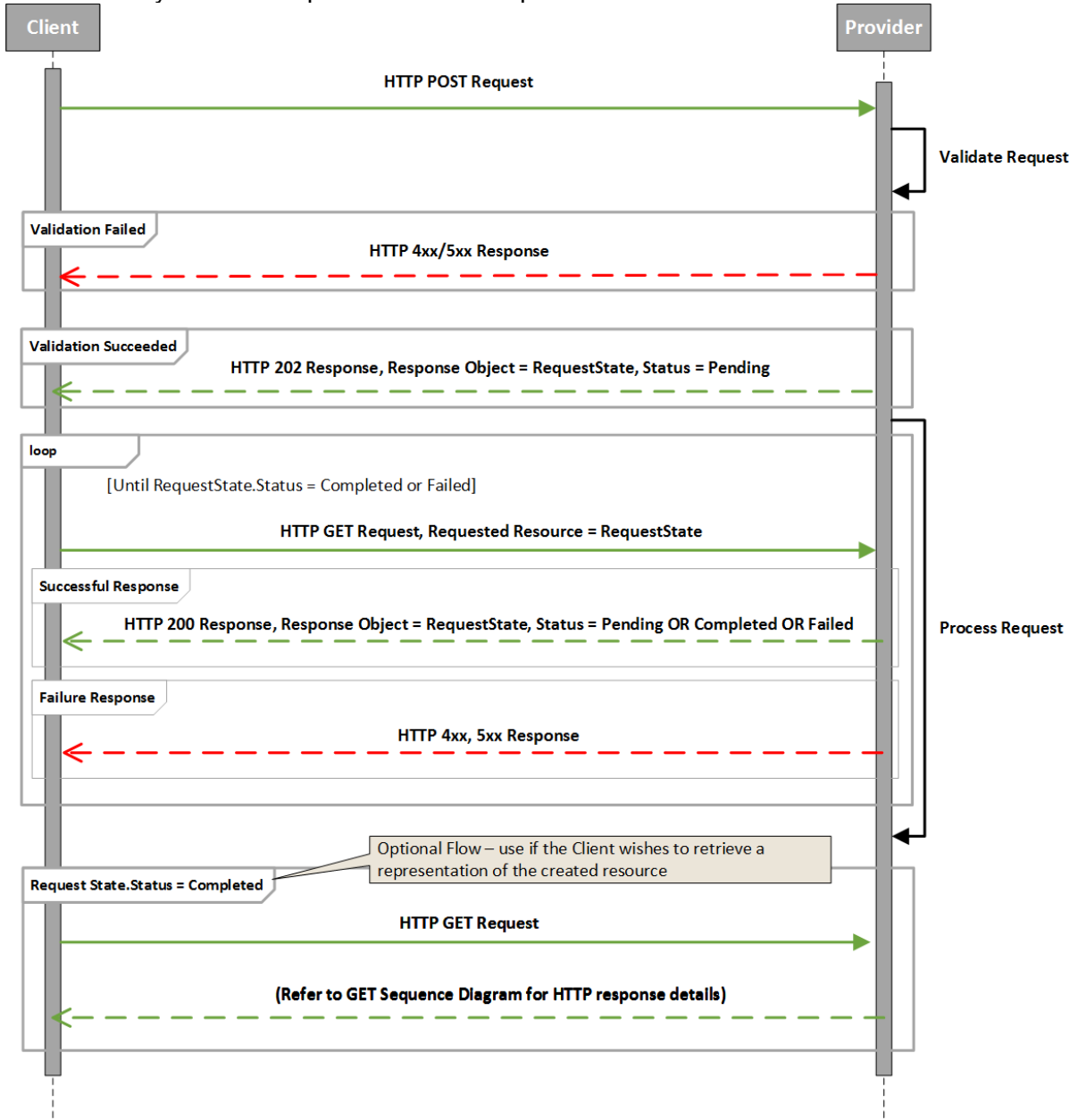


Figure 6: Create (POST) Polling Asynchronous Flow Pattern

Mobile Money API 1.0.0 Specification Description

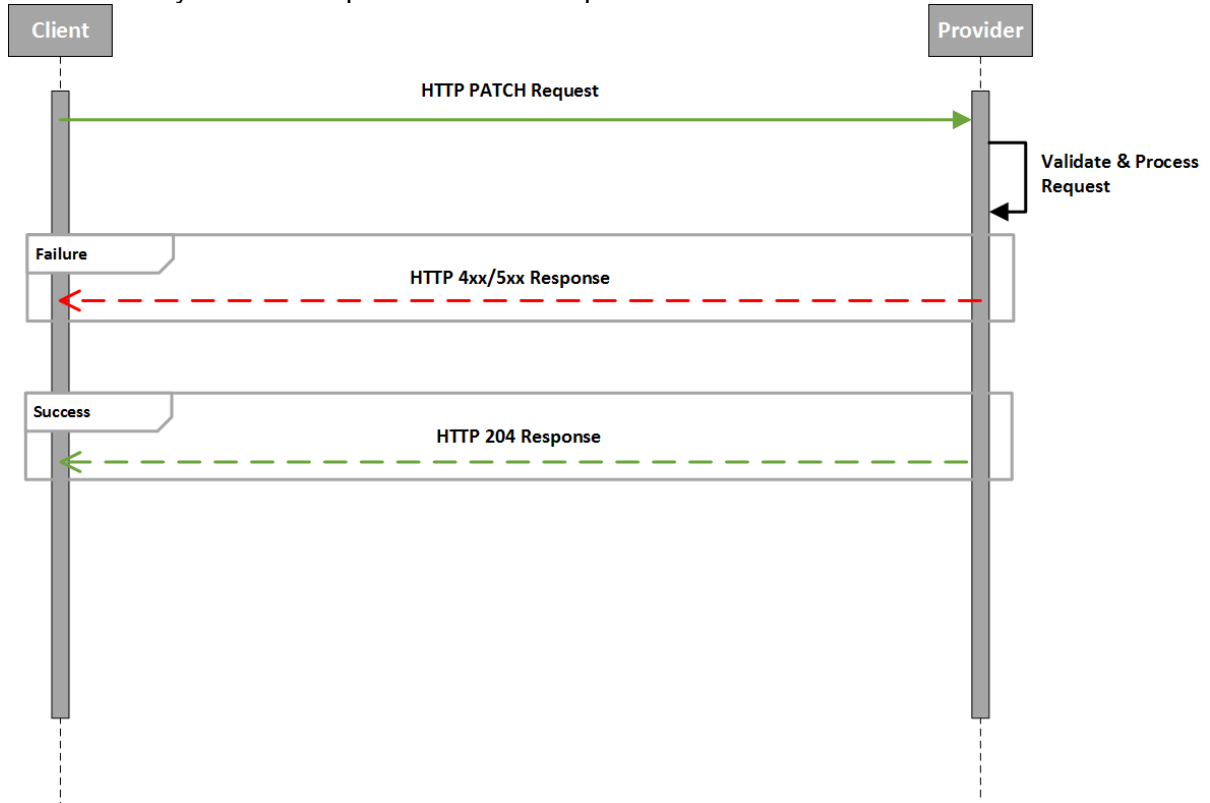


Figure 7: Update (PATCH) Synchronous Flow Pattern

Mobile Money API 1.0.0 Specification Description

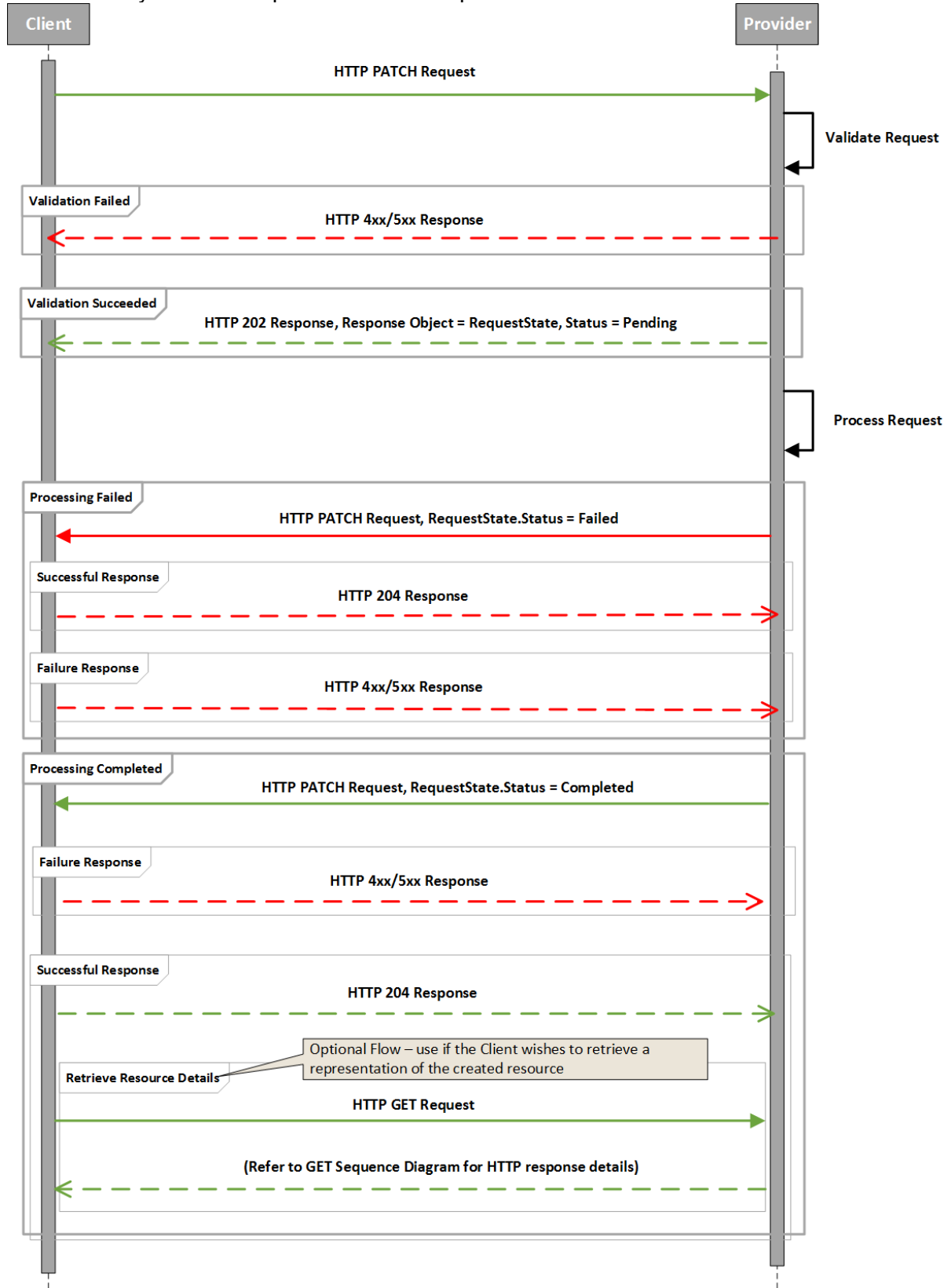


Figure 8: Update (PATCH) Callback Asynchronous Flow Pattern

Mobile Money API 1.0.0 Specification Description

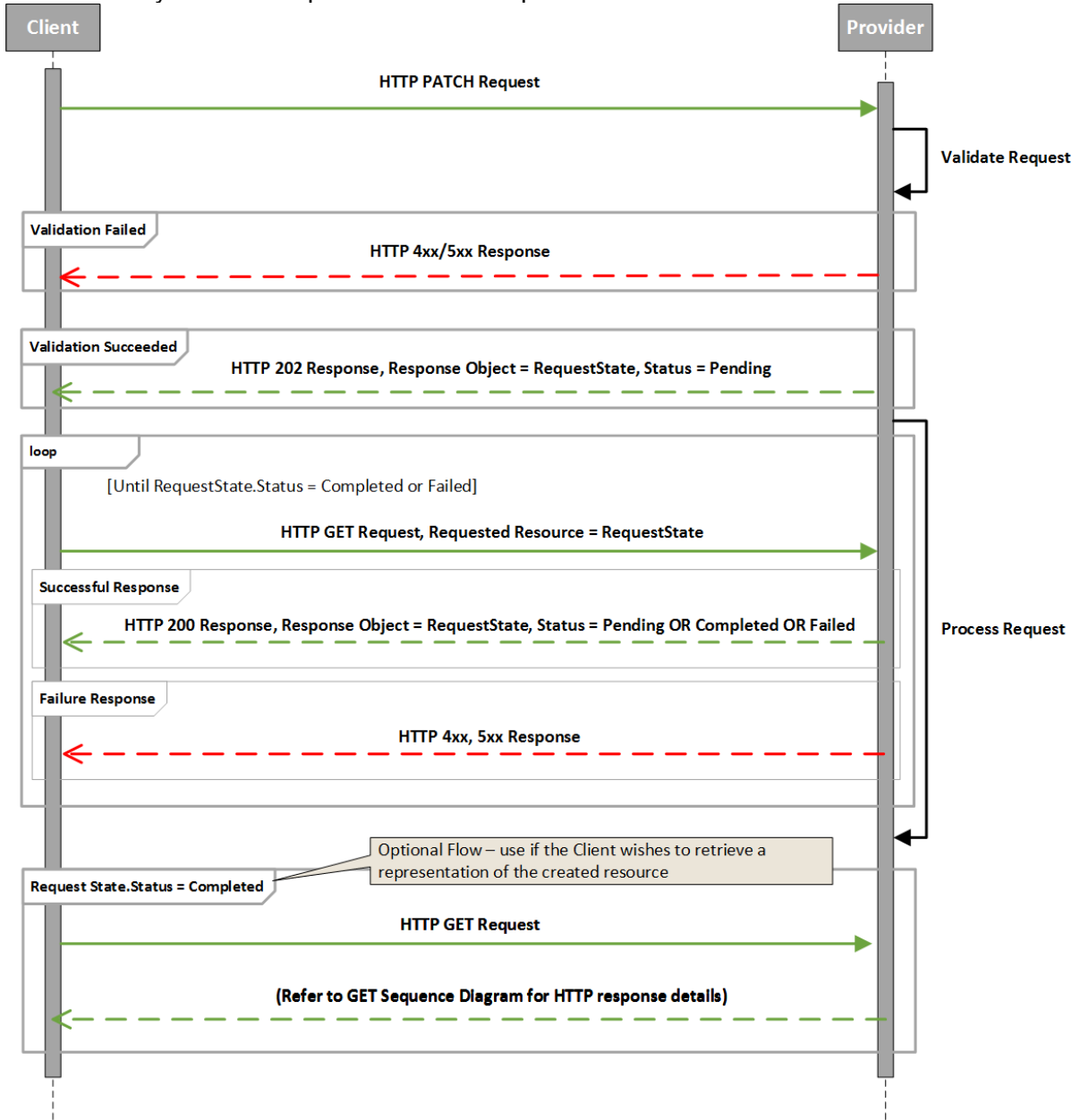


Figure 9: Update (PATCH) Polling Asynchronous Flow Pattern

Mobile Money API 1.0.0 Specification Description

7.3 HTTP Status Responses and Error Categories

At a high-level, the following HTTP status codes are returned for operations:

Operation	Success	Intermediate*	Client Error	Server Error
GET	200	N/A	400, 401, 404	500, 503
PATCH	204	202	400, 401, 404	500, 503
POST	201	202	400, 401, 404	500, 503

*Note that 'intermediate' in the table above relates to the HTTP response for the first leg of an asynchronous request.

7.3.1 Client Error Categories

Client errors fall into one of four categories as shown below:

Error Category	Description	HTTP Response Code
BusinessRule	The resource could not be successfully completed due a violation of a business rule. Business rules include financial limit violations, duplicate requests and invalid states.	400
Validation	Violation of a constraint that will prevent the resource from being processed. Examples include: <ul style="list-style-type: none"> - Invalid property length - Failure to resolve regular expression 	400
Authorisation	It was not possible to authenticate or authorise the client or other party to perform the service	401
Identification	The requested resource could not be matched on the system with the supplied identifier(s).	404

Additional non-harmonised 4xx HTTP response codes may be returned to the client by web or proxy servers.

Where the need for prevention of information leakage is paramount and this outweighs the benefits of error granularity, the provider may wish to use a 400 response code with no error body.

7.3.2 Server Error Categories

Server Errors are captured in two simple categories as per the following table:

Error Category	Description	HTTP Response Code
Internal	The request could not be completed due to a non-client related issues that do not constitute complete system unavailability. Examples include software licence issues, unavailability of system configuration information.	500

Mobile Money API 1.0.0 Specification Description

Service Unavailable	The service is not currently available. This could be due to network issues, issues with individual components or complete systems outages. Regardless of the cause, the result means that the request cannot be performed.	503
---------------------	---	-----

Additional non-harmonised 5xx HTTP response codes may be returned to the client by web or proxy servers.

7.4 Error Codes Definition

The mobile money API uses harmonised error codes in order to provide consistent error reporting back to API clients. Error codes are structured as per follows

- Error Category. All error codes must be associated with a [client](#) or [server](#) error category. This provides context and uniqueness to the error code.
- Error Code. Provides the reason for the request failure. A human-readable description of the error is also available. Error codes are generally granular, i.e. they provide a specific reason for failure. In some cases, granularity is neither possible or desirable. Where this is the case, a 'Generic' code has been defined. Cases include:
 - The API Provider has generated an error that does not map to existing Rule error codes.
 - The API Provider wishes to avoid disclosure of confidential information regarding the resource or parties to the resource. For example, the fact that a customer has breached their monthly transaction limit may not be disclosed to specific clients.
- Error Parameters. Provides a construct to communicate supplementary information regarding the error in key/value pairs. The supplementary information is currently non-harmonised and can include:
 - The API provider-specific Error Code and Description.
 - Additional identification of the error subject, e.g. account identifiers, invalid properties etc...
 - Diagnostic information, e.g. affected subsystem, licence failure type etc...

Note that although error parameter contents are non-harmonised, care should be taken regarding confidentiality of disclosed information. Confidential parameter information should only be disclosed to trusted clients.

A full list of harmonised Error Codes is provided in Appendix A.

7.5 Errors Object Definition

All errors generated by the API provider will contain an Errors object in the response body. At a minimum, the Errors object must contain the Error Reason and Error Code.

Errors Object Properties					
Name	Type	Description		Reference	Validation
Error Category	String	The category grouping for the error.	→M ←M		Enumeration = Error Categories
Error Code	String	The harmonised error code identifying the reason for error.	→M ←M		Enumeration = Error Reasons
Error Description	String	A textual Description of the error.	→O ←O		
Error DateTime	DateTime	The timestamp indicating when the error occurred.	→O ←O		
Error Parameters	Reference	Diagnostic information in the form of key/value pairs relating to the error.	→O ←O	Metadata	

7.6 API Heartbeat

The Heartbeat API is used for monitoring purposes and establishes whether an API provider's system is in a state that enables a client to submit a request for processing within established SLAs. There are three states that can be returned by the API provider in response to a heartbeat request:

- Available. The system is available and can receive and complete requests within SLAs.
- Degraded. The system can receive and complete requests but not within SLAs, i.e. delay in transaction processing is anticipated. When known, the expected processing delay time can be returned by the provider.
- Unavailable. The system cannot receive and process requests. Any submitted requests will fail whilst the system is in this state.

The Heartbeat can be requested using the following format:

- ***/heartbeat***. Issues heartbeat to determine availability of the transactions service, for example International Transfers.

Only synchronous API Heartbeat requests are supported. Note that the HTTP Request does not contain a request body. The HTTP response contains the following properties.

Mobile Money API 1.0.0 Specification Description

Heartbeat Response Properties					
Name	Type	Description		Reference	Validation
Service Status	String	Provides the status of the requested service.	→NA ←M		Enumeration = available, unavailable, degraded
Delay	Number	The anticipated processing delay in milliseconds	→NA ←O		Service Status must be set to degraded.
Planned Restoration Time	Datetime	Where the planned restoration time is known (e.g. scheduled maintenance), it can be provided in this property	→NA ←O		

7.7 Missing Response Retrieval

In some circumstances, the client may not have received the final representation of the resource for which it attempted to create. For example, a proxy server issue may have resulted in a HTTP 5xx response but the provider may have actually successfully completed the request. The /Responses API allows a client to identify and retrieve the final representation of the resource assuming that the resource was created. In order to get a representation, the client issues a GET /Responses/{Client Correlation ID}. The provider will then match the client correlation ID to the appropriate resource and return a link to that resource. If the resource is not found for the given correlation ID then a HTTP 404 will be returned.

The response object for */responses* is detailed below.

Responses Properties					
Name	Type	Description		Reference	Validation
Link	String	Provides a URL to the resource associated with the given correlation ID	→NA ←M		

7.8 Harmonised Error Codes

The error codes defined in this section are considered a basic initial set and will be expanded over time.

Error Category	Error Code	Error Code Description
BusinessRule	GenericError	A generic Error Code for the Rule Error Category. This is used in two scenarios: 1. The API Provider has generated an error that does not map to existing Rule error codes. 2. The API Provider wishes to avoid disclosure of confidential information regarding the resource or parties to the resource.
BusinessRule	DailyVolumeLimitExceeded	The party has exceeded their daily transacting volume limit - This can be a service limit or a limit that is specific to the party.
BusinessRule	DailyValueLimitExceeded	The party has exceeded their daily transacting value limit - This can be a service limit or a limit that is specific to the party.
BusinessRule	WeeklyVolumeLimitExceeded	The party has exceeded their weekly transacting volume limit - This can be a service limit or a limit that is specific to the party.
BusinessRule	WeeklyValueLimitExceeded	The party has exceeded their weekly transacting value limit - This can be a service limit or a limit that is specific to the party.
BusinessRule	MonthlyVolumeLimitExceeded	The party has exceeded their monthly transacting volume limit - This can be a service limit or a limit that is specific to the party.
BusinessRule	MonthlyValueLimitExceeded	The party has exceeded their monthly transacting value limit - This can be a service limit or a limit that is specific to the party.
BusinessRule	AccountMaxTotalValueExceeded	The party has exceeded their cumulative transacting value limit defined for the account.
BusinessRule	AccountMaxTotalVolumeExceeded	The party has exceeded their cumulative transacting volume limit defined for the account.
BusinessRule	LessThanTransactionMinValue	The amount specified for the transaction is less than the defined minimum for the service.
BusinessRule	GreaterThanTransactionMaxValue	The amount specified for the transaction is greater than the defined maximum for the service.
BusinessRule	MaxBalanceExceeded	The amount specified will cause the balance of the credit parties account to exceed the rule limit.
BusinessRule	SamePartiesError	The debit and credit parties are the same.
BusinessRule	DuplicateRequest	The request has previously been processed, i.e. this request is a duplicate and hence has been rejected.

Mobile Money API 1.0.0 Specification Description

BusinessRule	InsufficientFunds	Available funds are not sufficient to enable the party to be debited for the requested transaction.
BusinessRule	IncorrectState	The account is in a state that does not permit the requested service.
BusinessRule	UnderPaymentNotAllowed	The requested amount is less than the amount that needs to be supplied for this transaction
BusinessRule	OverPaymentNotAllowed	The requested amount is greater than the amount that needs to be supplied for this transaction
BusinessRule	RateLimitError	The client has submitted too many requests within a period of time.
BusinessRule	TransactionTypeError	The harmonised transaction type is not supported by the API Provider for the given request.
BusinessRule	NoMandateAuthority	The transaction request is not supported by the debit mandate.
BusinessRule	LinkViolation	The request has violated an account to account link for one of the following reasons: <ul style="list-style-type: none"> - Pull request not supported by the Link Mode. - Push request not supported by the Link Mode.
BusinessRule	CountryofOriginNotPermitted	The API Provider does not allow transactions to be accepted from the specified country of origin.
BusinessRule	NationalityNotPermitted	The API Provider does not allow transactions to be accepted from individuals with the specified nationality.
BusinessRule	IdDocumentNotSupported	The API Provider does not allow the specified ID Document.
BusinessRule	IssuingCountryNotSupported	The API Provider does not allow the issuing country of the specified ID document.
QuoteExpiry	QuoteHasExpired	A transaction with a quote reference cannot be processed at the quote has expired.
Identification	IdentifierError	The requested resource could not be matched on the system with the supplied identifier(s).
Validation	GenericError	A generic Error Code for the Validation Error Category. This is used in where the API Provider has generated an error that does not map to existing Validation error codes.
Validation	LengthError	The specified property contents are greater than the maximum allowed length or less than the minimum allowed length.
Validation	FormatError	The specified property contents do not conform to the format required for this Property.
Validation	NegativeValue	The amount supplied is negative and this is not allowed for the given service.
Validation	CurrencyNotSupported	The currency supplied is not supported by the API Provider.
Validation	MandatoryValueNotSupplied	A mandatory value has not been provided in the header and/or JSON body.

Mobile Money API 1.0.0 Specification Description

Validation	InvalidOffset	The pagination offset provided is not valid. This is typically because the offset is greater than the maximum number of records.
Authorisation	ClientAuthorisationError	General Client Authentication failure. No further details provided to prevent leakage of security information.
Authorisation	RequestDeclined	The debit party did not approve the request.
Authorisation	ServicingPartyAuthorisationError	The servicing party (e.g. agent) credentials or permissions are not valid.
Authorisation	RequestingPartyAuthorisationError	The party requesting the service has not provided the right credentials and/or does not have permission to perform this service.
Internal	GenericError	The request could not be completed due to a non-client related issues that do not constitute complete system unavailability. Examples include software licence issues, unavailability of system configuration information.
Service Unavailable	GenericError	The service is not currently available. This could be due to network issues, issues with individual components or complete systems outages. Regardless of the cause, the result means that the request cannot be performed.

Annex A Document Management

A.1 Document History

Version	Date	Brief Description of Change	Approval Authority	Editor / Company
1.0		New PRD (WG Doc nn/nnn).	WG #nn EMC #nn	
1.1		Minor CR nnn (WG Doc nn/nnn).<description of change><reason for change>	WG #nn	
2.0		Major CR nnn (WG Doc nn/nnn).<description of change><reason for change>	eVote EMC #nn	

A.2 Other Information

Type	Description
Document Owner	<Working Group/Project>
Editor / Company	

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@gsm.a.com.

Your comments or suggestions & questions are always welcome.