API Request-Response Flows Guidelines

GSMA Mobile Money API

This is a Non-binding Permanent Reference of the GSMA

Security Classification: Non-Confidential

Copyright Notice

Disclaimer

Antitrust Notice

Table of Contents

1 Introduction

1.1 Overview

The purpose of this document is to detail guidelines for selecting and implementing a GSMA Mobile Money API request-response flow. The guidelines apply to GSMA Mobile Money API Releases 1.1 and above.

2 Background

2.1 Supported Request-Response Flows

The Mobile Money API is an initiative developed through collaboration between the mobile money industry and the GSMA, and provides a harmonized API Specification for mobile money use cases which is both easy to use and secure. Using best practices from the technology industry in API design and security, it aims to simplify and accelerate integration with mobile money platforms and stimulate the growth of the ecosystem.
The key design principles of the API include the use of REST architectural principles, the use JSON, and the provision of a set of well-defined objects that are abstracted from the underlying object representations held in different mobile money systems.

A benefit of the GSMA Mobile Money API is the flexibility that allows mobile money providers to choose the specific request-response flow to align with the capabilities of their mobile money platform. There are three request-response flows that are supported for requests that involve resource creation or update:

- **Asynchronous with Call-back**. This flow allows a calling API client to issue a request via the GSMA Mobile Money API with the result returned via a call-back to a URL specified by the API client.
- **Asynchronous with Polling**. This flow allows a calling API client to issue a request via the GSMA Mobile Money API. The client is then required to 'poll' the API provider to determine the result.
- **Synchronous**. This flow allows a calling API client to issue a request via the GSMA Mobile Money API. The client will in turn receive the result in the corresponding response to the request.

Note that data retrieval (read) requests always use the synchronous request-response flow.

2.2 Why Multiple Request-Response Flows?

Mobile Money Providers use a wide range of technology to provide mobile money services to their customers, agents and merchants. Some providers use 'off-the-shelf' mobile money systems whilst other providers have developed their own systems. Older systems will typically support only the synchronous request-response flow whereas more modern systems will typically support one or both of the asynchronous request-response flows.

Whilst there are many benefits to be realised by using a single request-response flow to harmonise all implementations, the costs to Mobile Money Providers to modify systems architectures to align to a single flow is in many cases significant. This would therefore serve as a barrier to adoption of the GSMA Mobile Money API. As such, a strategic decision was taken to allow the GSMA Mobile Money API to support as many providers as possible using the three flows.

2.3 Why Request-Response Flow Guidelines?

With three supported request-response flows, there can be confusion as to which flow to use, what options are available and how to implement according to best practice. The aim of these guidelines is to produce clear and consistent implementation instructions for adopters of the GSMA Mobile Money API. This will allow a consistent implementation approach for each flow and reduce the costs/complexities of adoption.

2.4 Choosing a Request-Response Flow

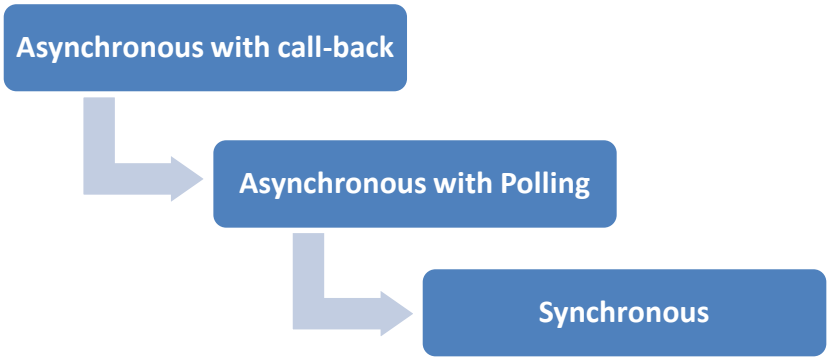
There are a several variables to consider when choosing a request-response flow for the GSMA Mobile Money API. This includes:

- Expected volumes of transactions to be processed.
- Capabilities and architecture of the adopter's mobile money platform.

The **Asynchronous with call-back** request-response flow is recommended for most implementations. From an architectural perspective, it allows the client and API provider to free up system resources by processing the request in an asynchronous manner. It also provides robust error handling, allowing clients determine the outcome of a request in all circumstances. The API also provides flexibility in allowing the client to determine the destination of the call-back.

Where a mobile money platform is unable to implement call-backs, the **Asynchronous with Polling** request-response flow should be considered. This does add additional overhead for the API client due to the need to repeatedly poll determine the result of the request.

The **Synchronous** request-response flow should only be used by adopters that expect low transaction volumes or have a platform that cannot support either of the asynchronous flows.



3 Request-Response Flows - Detailed Guidelines

3.1 Comparison of Flows

Characteristic	Flow		
	Asynchronous with call-back	Asynchronous with Polling	Synchronous
Call-back to receive result	Yes	No	No
Poll to determine result	No	Yes	No
Mandated use of Client Correlation Id	Yes	No	Yes
Use of Server Correlation Id	No	Yes	No
Retrieval of Missing Response	Via /responses	Via /requeststates	Via /responses

Table 1 Comparison of Request-Response Flows

As per table 1, each request-response flow can be differentiated as follows:

- The **asynchronous with call-back** flow is the only flow where the client is notified of the result of the request via call-back.
- The **asynchronous with polling** flow is the only flow where the client is required to poll the API provider to determine the result of the call-back.
- The **asynchronous with call-back** and **synchronous** flows mandate use of a client-generated correlation id as the basis for retrieving the result if the API provider did not initially return it, for example, due to a network issue.
- The **asynchronous with polling** flow uses an API provider (server) generated correlation id as the basis for retrieving the result if the API provider did not initially return it, for example, due to a network issue.
- The **asynchronous with call-back** and **synchronous** flows use the /responses API to retrieve a missing result, passing the client correlation Id in the request.
- The **asynchronous with polling** flow calls the /requeststates API to retrieve a missing result, passing the API provider (server) correlation Id in the request.

3.2 Asynchronous Request-Response Flow with Call-back

3.2.1 Asynchronous Request-Response Flow with Call-back - Guidelines for Successful Requests

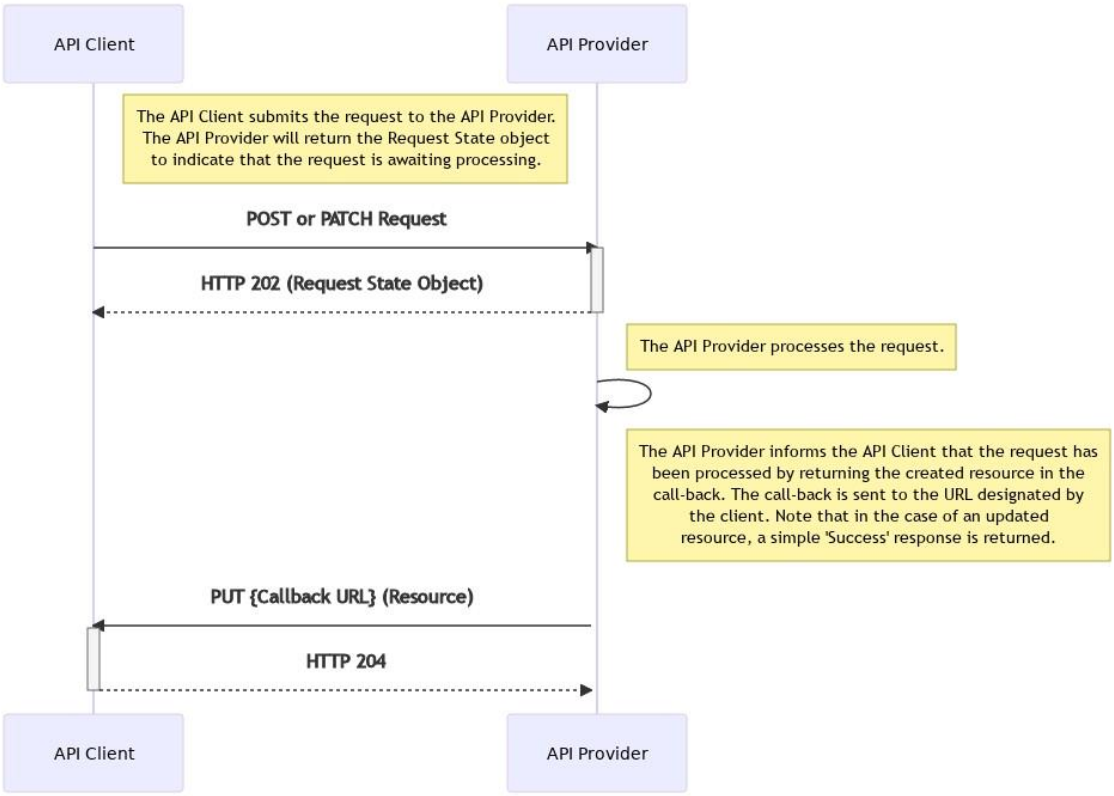


Figure 1 Asynchronous Request-Response Flow with Call-back Overview

Figure 1 describes a successful request using the **asynchronous with call-back** request-response flow. The approach can be used for creation and update requests. When using this flow, the following must be implemented:

- The client must generate a client correlation id and pass this via the *X-CorrelationID* header in the initial request.
- The *X-CorrelationID* header must be stored by the API provider so that the client can use it to retrieve the result should the client not receive the call-back.
- The client must provide a call-back URL and pass this via the *X-Callback-URL* header in the initial request.
- The API provider must return the details of the created resource in the call-back. The details are identical to those that would be returned in a synchronous response. In the case of an updated resource, the result will simply contain a 'Success' notification.

3.2.2 Asynchronous Request-Response Flow with Call-back - Guidelines for Returning Errors

There will be circumstances where an API request will fail. For example, the format of the request may not be valid or the request may fail to process due to incorrect identification of an account. The GSMA Mobile Money API allows for errors to be communicated in the initial acknowledgement step or the final call-back step as illustrated in figures 2 and 3.

3.2.2.1 Returning an Error on the Acknowledgement Step

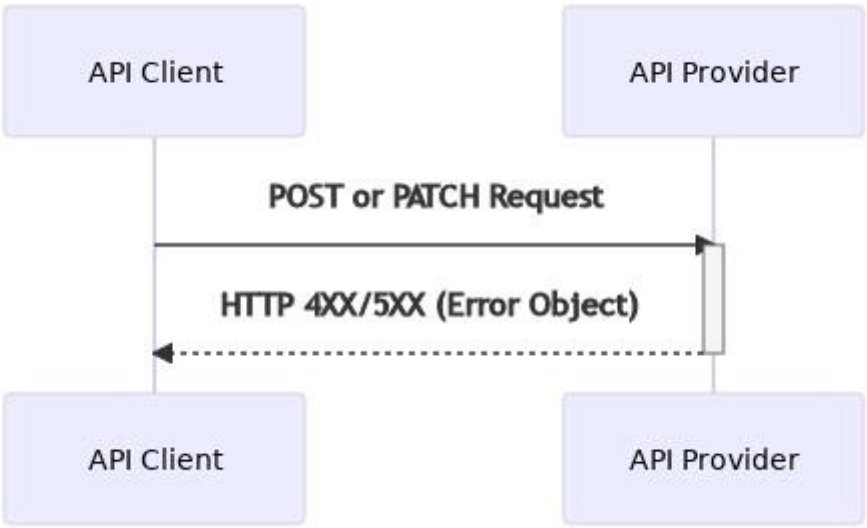


Figure 2 Returning an Error on the Acknowledgement Step

Following submission of the request from the API client, if, for any reason the API provider is unable to submit the request for processing, then the API Provider will reject the request, returning the following:

- The appropriate HTTP response code. This will be a 4xx or 5xx error as documented in the GSMA Mobile Money API specification.
- The *error* object that details the nature of the error. The *error* object is documented in the GSMA Mobile Money API specification.

3.2.2.2 Returning an Error on the Call-back Step

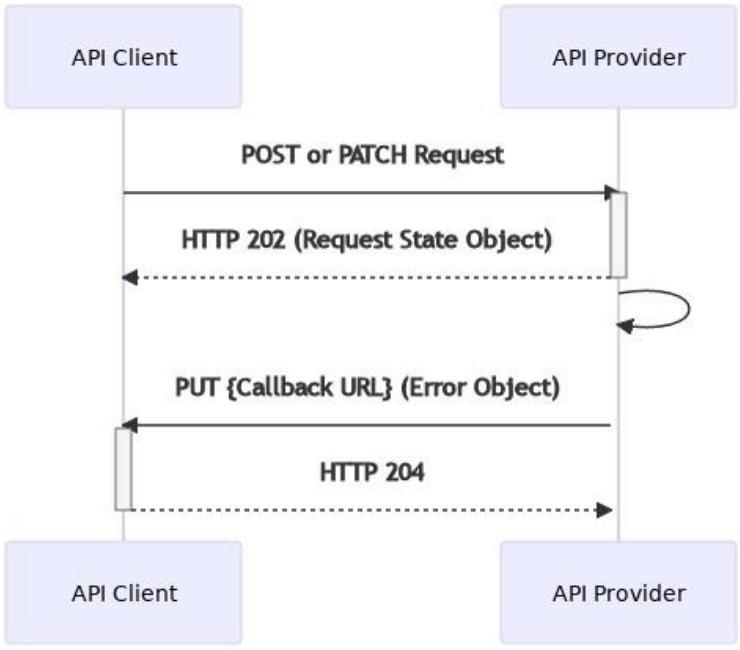


Figure 3 Returning an Error on the Call-back Step

Following submission of the request from the API client, the API provider acknowledges and processes the request in the same manner as defined in Guidelines for Successful Requests. If processing fails, then the API provider must return the *error* object in the call-back request, detailing the nature of the error. The *error* object is documented in the GSMA Mobile Money API specification.

3.2.2.3 Guidance on Returning Errors

Table 2 lists the categories of errors that can occur when using the GSMA Mobile Money API, and provides recommendations as to the step on which the client should be informed of the error.

Error Category	Description	Step to Respond
Schema Validation	Where input provided by the client does not confirm to the API specification.	Acknowledgement
Client Authorisation[1]	Where the client submits invalid credentials resulting in an authentication failure or where the client does not have sufficient permissions to perform the request.	Acknowledgement
End-user Authorisation	Where the API provider requests the end-user to authorise a request who fails to do so.	Call-back
Identification	Where the API provider cannot locate the subject identity or identities of the request.	Call-back
Business Rule	Where the client submits valid input that passes schema validation but the request subsequently fails to process due to a business rule violation, for example, maximum account balance exceeded.	Call-back
Service Unavailable	Where the client can reach the API provider, but the provider's system is not in a state that will permit the request to be accepted through no fault of the client.	Acknowledgement

Table 2 Guidance on Returning Errors

[1] Where an API provider implements an API Gateway, the gateway may reject this request prior to the acknowledgement step.

3.2.3 Asynchronous Request-Response Flow with Call-back - Guidelines for Non-Responses

There will be circumstances where the client does not receive the call-back containing the final result of the API request. In these circumstances, the client has the ability to retrieve the missing result as detailed in figure 4.

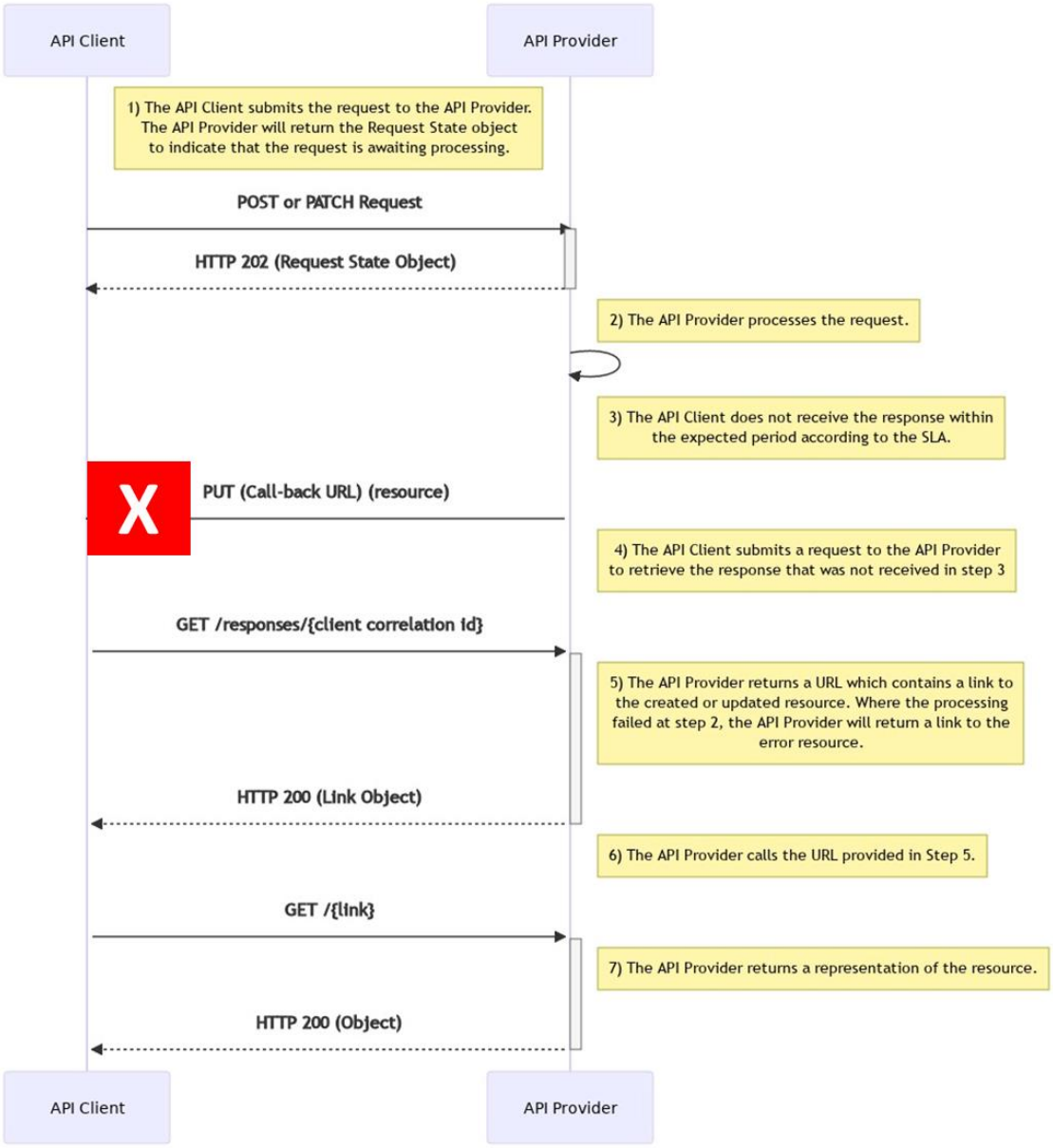


Figure 4 Retrieve a Missing Response

The link returned in step 5 of figure 4 would represent the URL of the created or updated resource in the event that the API request had completed successfully. So for example, for a successful transaction creation, the */responses* API would return a link similar to *{domain}/v1.x/mm/transactions/12345* where '12345' is the transaction reference that was assigned to the created transaction. However there will be situations where the client did not receive a call-back notification for a request that had failed. In this case, the */responses* API will need to return a link to the error record, for example *{domain}/v1.x/mm/errors/45678* where '45678' is the error id. The client can then subsequently call this link which will result in the error details being returned in the form of the *errors* object.

If the API provider is unable to return the link in step 5, then the client can choose to resend the request. When resending a creation request, the client must supply the same Client Correlation Id that was previously used. If the previous request was subsequently processed then the API provider will reject the request due to a duplicate Client Correlation Id. When resending an

update request, use of the Client Correlation Id is not mandatory as all update requests are idempotent.

3.3 Asynchronous Request-Response Flow with Polling

3.3.1 Asynchronous Request-Response Flow with Polling - Guidelines for Successful Requests

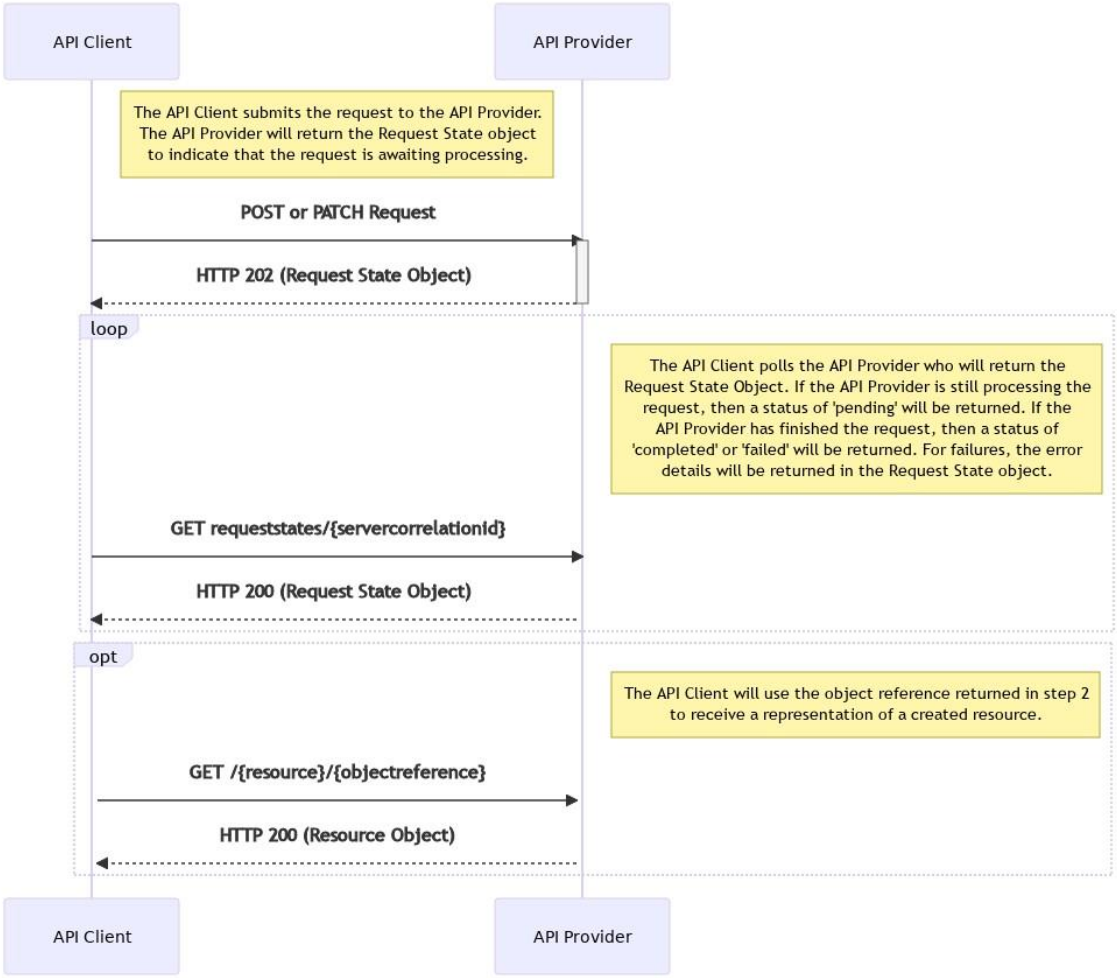


Figure 5 Asynchronous Request-Response Flow with Polling Overview

Figure 5 describes a successful request using the **asynchronous with polling** request-response flow. The approach can be used for creation and update requests. When using this flow, it is recommended that the following guidelines are adopted:

- The API provider must generate and return a Server Correlation Id in step 1 – this will subsequently be used by the client in step 2.

- The object reference will normally only be returned in the Request State object when the request is completed. In some implementations, the API Provider may wish to assign the object a reference in response to the initial request (step 1).
- It is recommended that the API Provider informs the API client of the maximum number of permitted polling attempts by populating the *pollLimit* in step 1. The maximum polling period would need to be agreed between the client and provider and is not defined in the Mobile Money API.

3.3.2 Asynchronous Request-Response Flow with Polling - Guidelines for Returning Errors

There will be circumstances where an API request will fail. For example, the format of the request may not be valid or the request may fail to process due to incorrect identification of an account. The GSMA Mobile Money API allows for errors to be communicated in the initial acknowledgement step or the polling step.

3.3.2.1 Returning an Error on the Acknowledgement Step

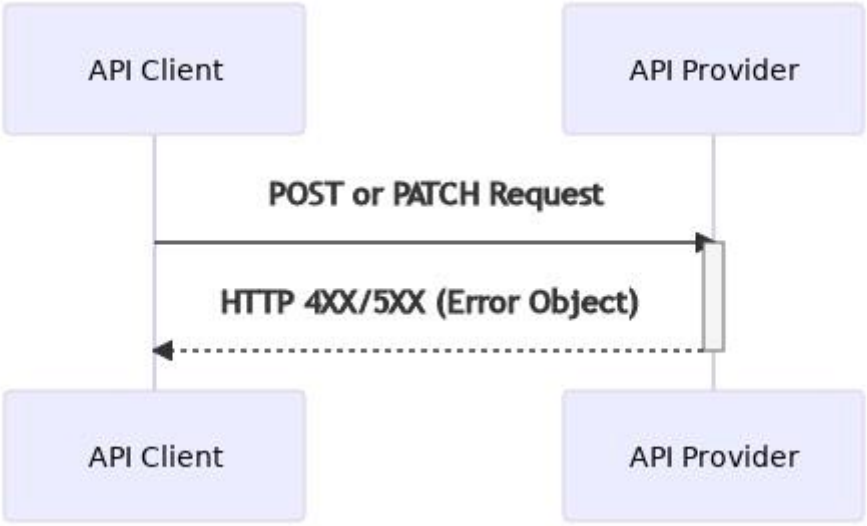


Figure 6 Returning an Error on the Acknowledgement Step

Following submission of the request from the API client, if, for any reason the API provider is unable to submit the request for processing, then the API Provider will reject the request, returning the following:

- The appropriate HTTP response code. This will be a 4xx or 5xx error as documented in the GSMA Mobile Money API specification.
- The *error* object that details the nature of the error. The *error* object is documented in the GSMA Mobile Money API specification.

3.3.2.2 Returning an Error on the Polling Step

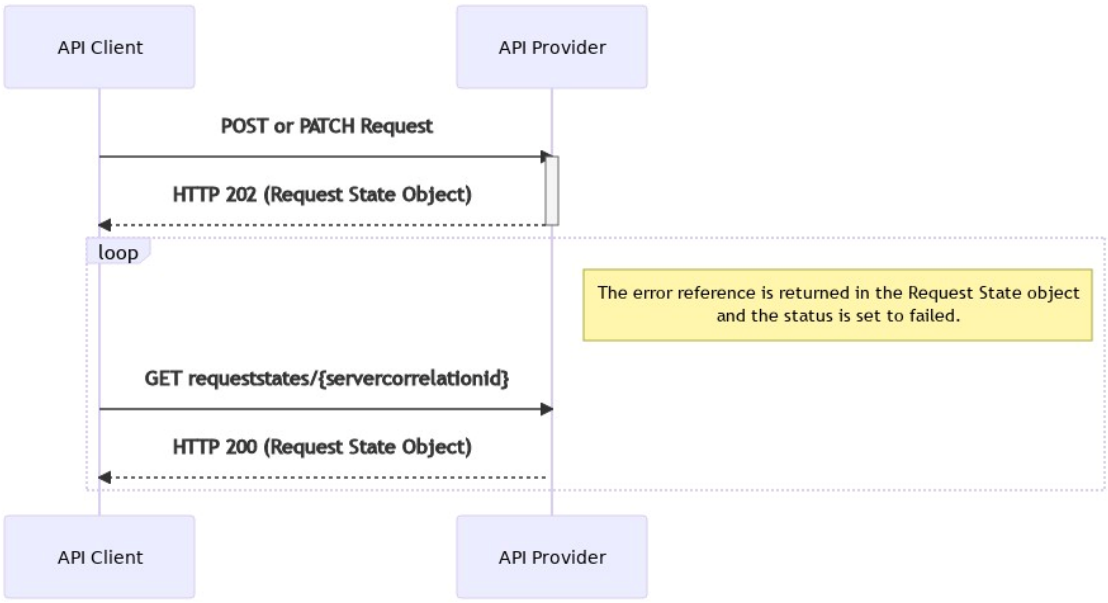


Figure 7 Returning an Error on the Polling Step

Following submission of the request from the API client, the API provider acknowledges and processes the request in the same manner as defined in Guidelines for Successful Requests. If processing fails, then the API client will use the polling flow to retrieve the details of the error. The API provider should follow these recommendations when returning errors on the polling step:

- The *status* in the Request State object must be set to 'failed'.
- The *errorReference* in the Request State object must be used to pass the details of the error. The *errorReference* is documented in the GSMA Mobile Money API specification.

3.3.2.3 Guidance on Returning Errors

Table 3 lists the categories of errors that can occur when using the GSMA Mobile Money API, and provides recommendations as to the step on which the client should be informed of the error.

Error Category	Description	Step to Respond
Schema Validation	Where input provided by the client does not confirm to the API specification.	Acknowledgement
Client Authorisation[1]	Where the client submits invalid credentials resulting in an authentication failure or where the client does not have sufficient permissions to perform the request.	Acknowledgement
End-user Authorisation	Where the API provider requests the end-user to authorise a request who fails to do so.	Polling
Identification	Where the API provider cannot locate the subject identity or identities of the request.	Polling
Business Rule	Where the client submits valid input that passes schema validation but the request subsequently fails to	Polling

	process due to a business rule violation, for example, maximum account balance exceeded.	
Service Unavailable	Where the client can reach the API provider, but the provider's system is not in a state that will permit the request to be accepted through no fault of the client.	Acknowledgement

Table 3 Guidance on Returning Errors

[1] Where an API provider implements an API Gateway, the gateway may reject this request prior to the acknowledgement step.

3.3.3 Asynchronous Request-Response Flow with Polling - Guidelines for Non-Responses

With the asynchronous with polling flow, the client will be expected to poll until the final result is returned by the API provider. However, if the API provider defines a poll limit and/or maximum period where polling is allowed, then there can be a situation where the client has exhausted their polling opportunities without receiving a final result. In this case, the client can choose to resend the request. When resending a creation request, the client must supply the same Client Correlation Id that was previously used. If it transpired that the previous request was subsequently processed then the API provider will reject the request due to a duplicate Client Correlation Id. When resending an update request, use of the Client Correlation Id is not mandatory as all update requests are idempotent.

3.4 Synchronous Request-Response Flow

3.4.1 Synchronous Request-Response Flow – Guidelines for Successful Requests

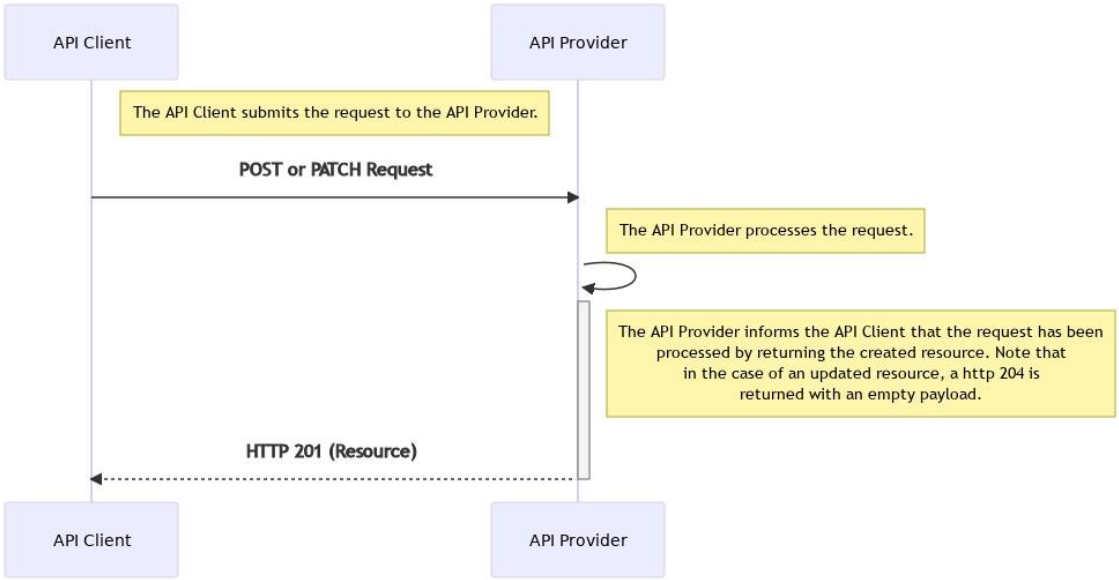


Figure 8 - Synchronous Request-Response Flow

Figure 8 describes a successful request using the **synchronous** request-response flow. The approach can be used for creation and update requests. When using this flow, the following must be implemented:

- The client must generate a client correlation id and pass this via the *X-CorrelationID* header in the initial request.
- The *X-CorrelationID* header must be stored by the API provider so that the client can use it to retrieve the result should the client not receive a response.

3.4.2 Synchronous Request-Response Flow with Call-back - Guidelines for Returning Errors

There will be circumstances where an API request will fail. For example, the format of the request may not be valid or the request may fail to process due to incorrect identification of an account. The GSMA Mobile Money API allows for errors returned for the synchronous request-response flow as illustrated in figure 9.

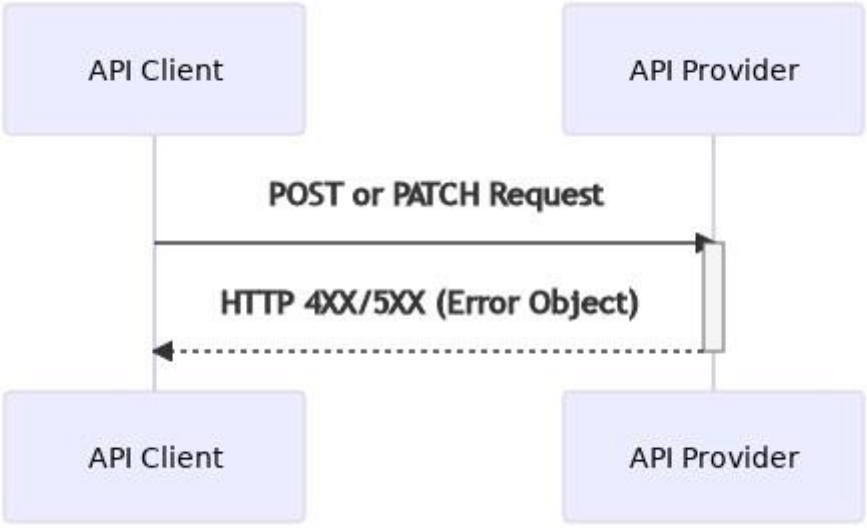


Figure 9 Returning an Error with the Synchronous Request-Response Flow

Following submission of the request from the API client, if, for any reason the API provider is unable to process the request, the following must be returned:

- The appropriate HTTP response code. This will be a 4xx or 5xx error as documented in the GSMA Mobile Money API specification.
- The *error* object that details the nature of the error. The *error* object is documented in the GSMA Mobile Money API specification.

3.4.3 Synchronous Request-Response Flow - Guidelines for Non-Responses

There will be circumstances where the client does not receive the result from the API provider. In these circumstances, the client has the ability to retrieve the missing result as detailed in figure 10.

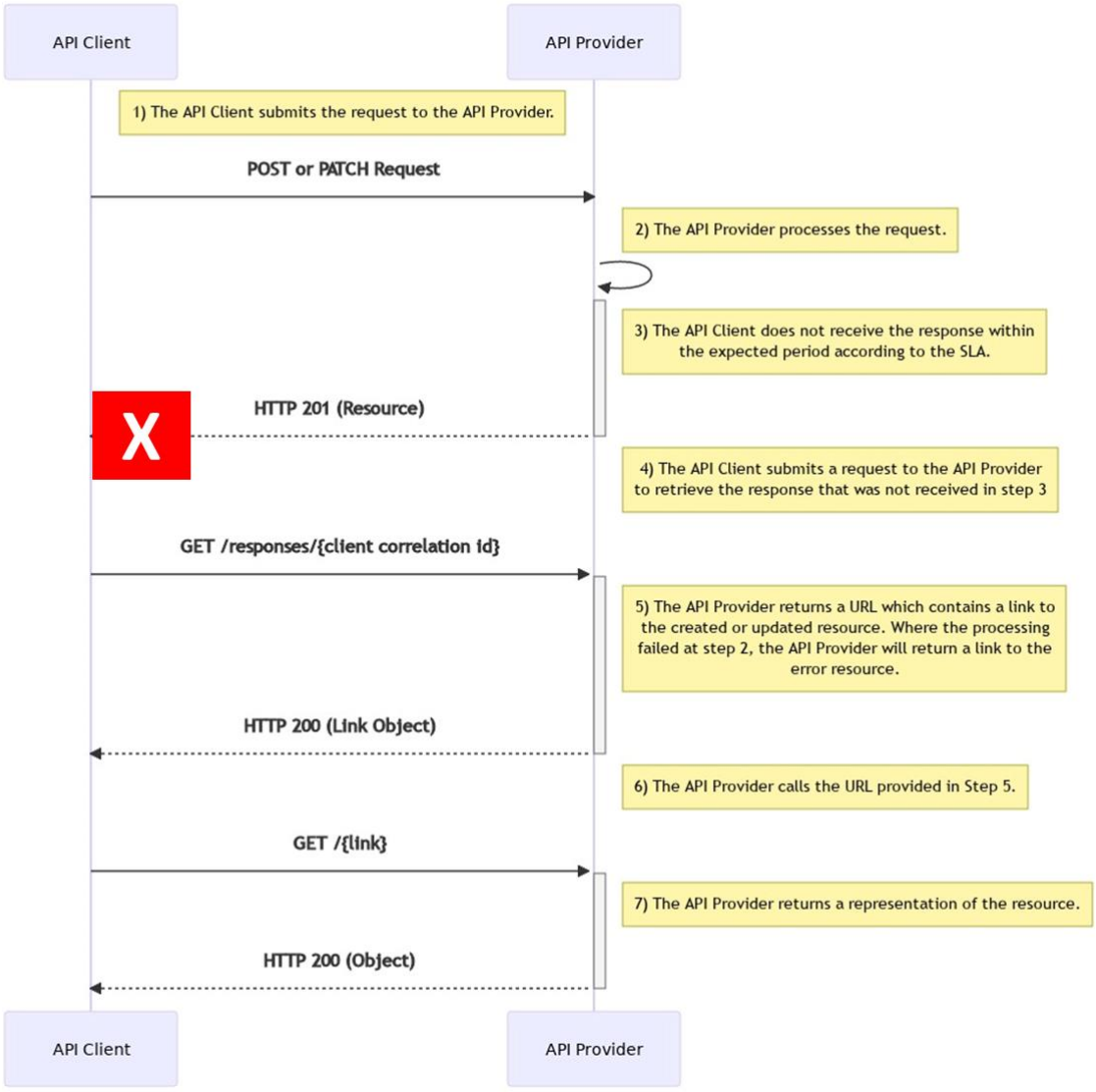


Figure 10 Retrieve a Missing Response

If the API provider is unable to return the link in step 5 above, then the client can choose to resend the request. When resending a creation request, the client must supply the same Client Correlation Id that was previously used. If the previous request was subsequently processed then the API provider will reject the request due to a duplicate Client Correlation Id. When resending an update request, use of the Client Correlation Id is not mandatory as all update requests are idempotent.