SIBS Gateway V2 Integration Manual

Version: 2025.03

Date: March 2025

Status: Done

Classification: Restricted

Document Info

Reference:	
Document Title:	SIBS Gateway V2 Integration Manual
Version:	2025.03
Status:	Draft
Classification:	Restricted
Document Type:	Manual

Distribution List

Name
SIBS Gateway 02.00 Integrators

Version History

Version	Date	Description	Author
01.00	30/06/2020	Document creation	Miguel Frederico
01.01	05/02/2021	Add Webhook examples	Fábio Júlio
01.02	16/03/2021	Add Static QR Code info on webhook section	Fábio Júlio
01.03	02/11/2021	Add MBWAY Mandates	Filipe Alemão
01.04	21/01/2022	Add MBWAY Mandates SPG Error Messages; Add TerminalId to Create Mandate API	Filipe Alemão
01.05	09/02/2022	Included new mandateStatus “EXPR”. Removed mandateStatus “INTT”. In chapter Notifications, the indication of a limited number of endpoints per Merchant removed. Expression “MB WAY Mandate” replaced by “Authorised Payment”.	Filipe Alemão

		In API and attribute names, the “Mandate” and “MBWAY Mandate” expressions were kept and refer to “Authorised Payment”. Create Authorised Payment Server to Server URI updated.	
01.06	28/04/2022	-New chapter for “Create Authorised Payment Server”. -Backoffice Operations table reviewed. -Headers Content-Type and X-IBM-Client-Id included in API’s description. -Attribute paymentMethod included in SIBS Gateway V2 Form Integration API Request. - paymentMethodList values reviewed. - threeDSecureOptions inserted in SIBS Gateway V2 Form Integration API Request . - Attribute name “systemManufacturer” replaced by “deviceManufacturer” in accordance with implementation. - Attribute “termsAndConditions” include SIBS Gateway V2 Form Integration in response. - MB WAY QR Code service presented as suspended.	Filipe Alemão
01.07	09/06/2022	- Chapter Notifications reviewed. - Description added to MandateStatus values. - amountLimit added to Get Authorised Payment API Response. - Instalment Plan added to Card API. - browserJavascriptEnabled attribute added to deviceInfo. - Data Dictionary: • Payment Method List and Payment Type attributes update with values description. • Instalment Plan added to dictionary. - Instalment Plan added to Payment Method Card Complex Type. - Instalment Plan added to Merchant Notification Request.	Filipe Alemão
2023.01	January 2023	- Version evolution to ‘v2’. - API Recurring discontinued. - API Merchant Initiated Transaction created. - ChapterNotifiactions reviewed: New ComplexType merchantInitiatedTransaction;	Filipe Alemão

		-New ComplexType customer – Added examples: Create Authorised Payment Creation Example; Cardholder Initiated Transaction with Type Recurring Example; Merchant Initiated Transaction with Type Recurring Example; Cardholder Initiated Transaction with Type UCOF Example; Merchant Initiated Transaction with Type UCOF Example; -Document Hyperlinks revision. -Document Endpoints revision.	
2023.04	April 2023	Edit Chapter 6.4.1 – Authorisation/Purchase Added the Fallback feature in Authorised Payments 7.1.1 SIBS Gateway V2 Form Integration Added a note about the header “Bearer” Added a note about the header “clientId” Added MB WAY In-App Purchase in MB WAY ID Chapter Fields added in API Chapter and Data Dictionary Field added to Complex Type – Merchant Notification Request	Hugo Mateus
2023.05	May 2023	Added “Authorised Payment Inquiry Financial Data” in Authorised Payment Chapter Added Get Authorised Payment Financial Data API in Chapter “Authorised Payment API” Added “Amount Available” and “Amount Limit” in Data Dictionary Update of SIBS Gateway V2 Error Message with Authorised Payment error messages Added in the Glossary the term “MULTIBANCO” Reference “MB Payment Reference” replaced by “MULTIBANCO Payment Reference” MB WAY QR Code service removed	Hugo Mateus Filipe Alemão

2023.09	September 2023	Added the fields “HMAC” and “Signing String” to the Transactional API’s and Backoffice Operations: Transactional API’s: • MBWAY ID • Authorised Payment – Purchase • MULTIBANCO. MB Service Reference Backoffice Operations: • Merchant initiated Transaction API • Capture API • Cancellation API • Refund API • Cancel Authorised Payment API Added the conditions of the fields for the API described above. In “Inputs and Outputs” the field “customerPhone” was erased in Authorise Payment and others fields were added The types of the fields of each API were detailed in order to define maximum size limits. Added “TransactionDateTime” to “MerchantNotificationRequest”.	Hugo Mateus Filipe Alemão
2023.10	October 2023	WIP	Filipe Alemão
		Added the fields “Signature” to the Transactional API’s and Backoffice Operations: Transactional API’s: • MBWAY ID • Authorised Payment – Purchase • MULTIBANCO. MB Service Reference Backoffice Operations: • Merchant initiated Transaction API • Capture API • Cancellation API • Refund API • Cancel Authorised Payment API Added Customer information in Merchant Initiated Transaction API, Capture API, Cancellation API, Refund API.	Hugo Mateus

		The size of “CustomerName” field was changed from Maximum 256 Text to Maximum 45 Text	
2023.11	November 2023	Merchant Notification chapter edited: • Added a Table and erased previous information	Hugo Mateus
2023.12	December 2023	Added “Get Payment Modalities section in Server-to-Server chapter. Fields from “Get Payment Modalities” were added to the section “Data Dictionary” and “Complex Types”	Hugo Mateus
2024.03	March 2024	Added New tables of Error Codes based on Excel of Error Codes created for this propose In the Header of card (server to server) API, mbway and reference was changed the field “Authorisation: Bearer token” to “Authorisation: Digest {transactionSignature} Mention of “STATIC_QRCODE” was erased in parameters table in spg config in chapter “SIBS Gateway V2 Form Integration” Request block “action processed” was erased from MBWAY, Authorised Payments and Multibanco in chapter “Server-to-Server” In chapter “Get Payment Modalities”, the initial description was modified. “Low_Risk” parameter erased from SIBS Gateway V2 Form Integration and from Data Dictionary	Hugo Mateus
2024.04	April 2024	Changed the table in Backoffice operations chapter Changed text in “Form Integration” chapter Changed text in “Server to Server” chapter Changed text in “Financial Operations” In “Integration Flows” chapter: • Changing order of chapters (Authorised Payment Creation by Client) • Authorised Payment Creation during Purchase/Authorisation erased Changed sequence diagram of “Multibanco” chapter	

		• Changing "Authorisation" to "Generate" Reference "Capture" erased Erased "capture" from Reference chapter In “API's” chapter was added link to sibs api market Adding "<x-ibm-client-id>" in Frontoffice chapter Chapter “SIBS Form integration v2”: • Changing condition of some fields (optional to Mandatory) “Actionresponse” complex type erased from Authorised payments	
2024.06	June 2024	Adding a note to the field “customerinfo.key” in form integration chapter Adding Card and Token API in Server-to-Server chapter Adding some indications regarding MBWAY in-App	Hugo Mateus
2024.08	August 2024	Adding fields “CountrySubDivision”, “browserScreenHeight”, “browserScreenWidth”, “browserIp” in FORM integration regarding 3DS VISA Parameters initiative. Error Codes tables updated – Error Code Table “Authorised Payments” was included in Error Codes Table “MBWAY” CARD API field “create token” detail was added.	Hugo Mateus
2024.10	October 2024	Data Dictionary – Changing TRANSACTIONRECIPIENTID instead of Max100Text to Max50Text Billing address fields for Card/Token Purchase are mandatory except street2 and CountrySubDivision In MBWAY API, the action response field was erased since MBWAY transaction doesn’t need 3DS “secureCode” field changed the type integer (3) to string (Max4Numeric) Field “Channel” limited as a string with maximum size three Sub-chapter of Merchant Notification – Webhook Retry System CustomerName field length changed it to Max35Text (this was the length presented to the merchants) Cardholder field length changed it to Max35Text (because of CustomerName field. In 3DS Parameters the customerName may be filled with the value of the cardholderName)	Hugo Mateus

		Added the POST ROOT URL for the checkout status inquiry using the merchantTransactionId Adding a note in “Payment Methods” chapter about the ability of making refunds Adding note about the reuse of MB Reference number (Chapter Generate MB Reference) Adding note in chapter “Form Integration Operations” about the languages available in SPG Form	
2024.11	November 2024	Added webhook examples Edited Merchant Notification System chapter. CardholderName e CustomerName fields with max length 45 Added in chapter “Form Integration” a note about the languages (FR, IT, DE)	Hugo Mateus
2024.12	December 2024	Added xPAY API Erasing “signing string” from API’s because is a field generated by sibs and not send through the API’s Added the submerchant fields in the scope of MB Reference. This fields will be included in the checkout service and must be filled by the merchant. Specified the mandateAction status “LMUP” in the webhook chapter. Changing fonts in some field tables	Hugo Mateus
2025.01	January 2025	Adding the note about the notification send by SIBS Gateway V2 to the merchant when creating or updating a Authorised Payment. Adding the fields “Mandate Expiration Date” and “MandateAmountLimit” to the webhook example of Authorised Payment Creation	Hugo Mateus
2025.02	February 2025	Added Checkout Status API Updated the error codes of Payment	
2025.03	March 2025	Glossary Update Erasing xPAY api Modified condition of Submerchant fields to optional Adding Cashout API, Cashout Error Codes and Cashout description (Integration Flows chapter) Adding in the checkout status the field “clientIBAN” regarding the cashout operation Adding Cashout webhook examples	

		Adding error code to Checkout Request “Invalid authentication or authorisation data”	

Table of Contents

List of Figures

List of Tables

Introduction

SIBS Gateway V2 enables you to accept e-commerce payments, having at your disposal several payment methods to make accepting payments online easy. This solution provides an online payment provider, gateway and Backoffice that integrates your website to a javascript solution with a secure form or through an Application Programming Interface (API).

Payment Methods

SIBS Gateway V2 supports both synchronous and asynchronous payments methods

- A synchronous payment method means that the final authorisation status is received in near real time in the API response and the payment is completed (Request - Answer).
- An asynchronous method means that the final authorisation is done in two steps to complete the payment (Request – Pending, need action).

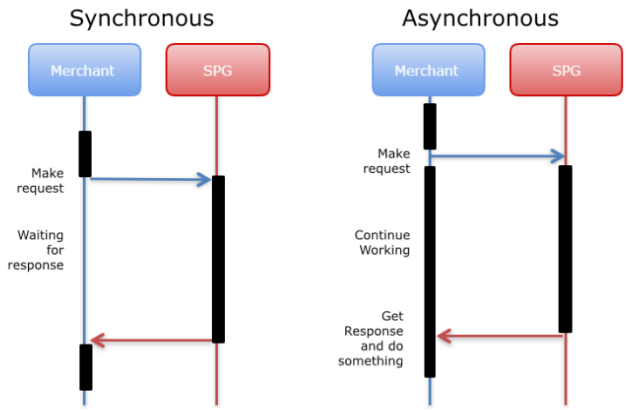

FIGURE 1 – SYNCHRONOUS & ASYNCHRONOUS

- **MB WAY ID** – is an asynchronous payment that requests the MB WAY enabled App, associated with the mobile phone number sent within the request, to accept the payment. This payment method has an amount limit of 99.999,99€.
- **Authorised Payment –** is a synchronous payment based on client alias (mobile phone number) and a previous established Authorised Payment identifier informed in the request. This allows the payment to be performed without client confirmation in App. To do so, the Authorised Payment Monthly Amount Limit and the Expiration Date must not have been reached. This payment method has an amount limit of 99.999,99€.
- **Card** – is a synchronous payment that allows the client to pay to the Merchant trough a banking card by inserting the Name, Card Number, Expiring date and CVV. This payment method has an amount limit of 499.999,99€.

- **Multibanco (MULTIBANCO Payment Reference) –** is an asynchronous payment that requests the generation of a MULTIBANCO Reference related to the payment registered by the previous Checkout. With the MULTIBANCO Reference returned in this operation, the customer can perform the payment. This payment method has an amount limit of 499.999,99€.

PAYMENT METHODS	CARD	MULTIBANCO	MBWAY ID	AUTHORISED PAYMENT
Synchronous	X			X
Asynchronous		X	X	

TABLE 1 – SYNCHRONOUS & ASYNCHRONOUS

All Payment Methods will be available for a refund for a maximum of 15 months.

Scope

Description of Relationship:

Frontoffice Operations

SIBS Gateway V2 supports two types of operations, purchases and authorisations, where the purchase triggers the payment immediately and the authorisation keeps the amount captive until a capture is performed.

- **Purchase**

  This transaction allows a cardholder to pay for a good or service via an electronic operation that debits his account and will credit the Acceptors banking account. Consists in an immediate debit in the client account and is an effective payment on the moment of the transaction.

- **Authorisation**

  Consists in an authorisation for a future payment, allows the Merchant to get an authorisation from the Issuer keeping the amount captive in the cardholders banking account, the transaction is authorised and the purchase will be triggered in a second moment where the merchant needs to trigger a “Capture” on the Backoffice, accordingly with the table below.

The following table describe the Frontoffice operations per payment method:

	PURCHASE	AUTHORISATION
CARD	X	X
MULTIBANCO	X	
MBWAY ID (ALIAS)	X	X
AUTHORISED PAYMENT	X	X

TABLE 2 – PAYMENT METHOD VS OPERATIONS TYPE

Backoffice Operations

At a Backoffice level, SIBS Gateway V2 supports the following types of operations:

- **Capture**

  Used after the cardholder provided an authorisation for the payment on a 1st stage and the amount is captured. Used on a 2nd stage to execute the payment that can happen in different ways (always triggered over an authorisation):

  - Full, capture the full amount authorized and finish the purchase
  - Partial, split the capture and do several payments up to the total amount authorized

- **Refund**

  The purpose of this operation is to refund the amount of a previous payment, crediting the cardholder account and debiting the Merchant account. A refund can be:

  - Full, when the total amount of the purchase is refunded to the cardholder
  - Partial, when a subtotal of the total purchase is refunded to the cardholder

- **Cancellation**

  Requests the cancellation of the amount (full or partial) of a previous authorisation or MULTIBANCO Reference generated, but not paid

- **Merchant Initiated Transaction**

  The purpose of this operation is to allow the merchant to charge a service, in the first stage the cardholder requests to register the payment as a Cardholder Initial Transaction and authorize the merchant to do future debits (Merchant Initiated Transactions) accordingly with the service.

  - Cardholder Initiated Transaction – Needed to register the initial conditions, authentication and consent collect from cardholder.
  - Merchant Initiated Transaction – Triggers following payments.

- Currently, two types of Merchant Initiated Transactions are accepted:
  - Recurring – Transactions processed at fixed, regular intervals not to exceed one year between Transactions, representing an agreement between a cardholder and a merchant to purchase goods or services provided over a period of time.
  - Unscheduled Card On File – A transaction using a stored credential for a fixed or variable amount that does not occur on a scheduled or regularly occurring transaction date, where the cardholder has provided consent for the merchant to initiate one or more future transactions which are not initiated by the cardholder.

- **Authorised Payments**

These operations allow the merchant to query Authorised Payments and if necessary, perform cancellations:

- Authorised Payment Creation – Allows the Merchant to request a Authorised Payment creation
- Authorised Payment List – Lists Merchant's Authorised Payments
- Authorised Payment Inquiry – Queries a Merchant's Authorised Payment
- Authorised Payment Cancel – Cancels an active Merchant's Authorised Payment

The following table describe the Backoffice operations per payment method:

	CARD	MIT	MULTIBANCO	MBWAY ID	AUTHORISED PAYMENT
CAPTURE	X	X		X	X
REFUND	X	X	X	X	X
CANCELLATION	X	X	X	X	X
CREATION		X			X
LIST		X			X
INQUIRY					X
CANCEL		X			X

TABLE 3 – BACKOFFICE OPERATIONS

It is possible to establish the below relationship among the Frontoffice & Backoffice operations, having in mind that Backoffice operations will act on the Frontoffice Operations:

	PURCHASE	AUTHORISATION	CAPTURE
CAPTURE		1	
REFUND	4		3
CANCELLATION		2	
MIT	5	5	

TABLE 4 – FRONTOFFICE & BACKOFFICE OPERATIONS

1 – A capture can act over an Authorisation of the total or partial amount

2 – An authorisation can be cancelled and the amount kept on hold will be released

3 – An effective payment through a capture can only be refunded

4 – An effective purchase only allows to trigger a refund in the Backoffice or via API

5 – A Merchant Initiated Transaction can be actioned over a purchase or an authorisation

Integration Options

SIBS Gateway V2 provides a collection of APIs that enables our clients to process and manage payments. Our APIs allow to accept payments from banking cards, MBWAY and to generate Multibanco references to perform payments in the Multibanco ATM network or in the home banking solutions from card issuers. It is not necessary to choose all the SIBS Gateway V2 payment solutions, our clients can choose which APIs to use.

Form Integration

Simple form integration with javascript based widget, on four simple steps:

1. **Prepare the checkout:** sends payment data, except payment method data.
   - First, **perform a server-to-server POST request** to prepare the checkout with the required data, including the order type, amount and currency. The response to a successful request is a JSON with a transactionID and other fields, as well as, a form context field that contains data which is required in the second step to create the payment form.
   - **Create the payment form**: displays a Payment Form to allow customers to submit payment method data. To create the payment form you just need to add the following lines of HTML/Javascript to your page and populate the following variables
     - The checkout transactionID that was returned in the response from step 1

```
<script src="https://{QLY/PRD}/assets/js/widget.js?id={transactionID}"></script>
```

[Change the variable accordingly with the environment – QLY or PRD]

- The {formContext} that you get in response from step 2
  - (**2.1**) {formConfig}
  - (**2.2**) {formStyle} (optionally)

```
<form class="paymentSPG" spg-context="{formContext}" spg-config="{formConfig}" spg-style="{formStyle}"><form>
```

   - **(2.3)** Configures the merchant *redirectUrl*, which is the page on your website where the customer should be redirected after the payment.
   - **(2.4)** Optional parameter that allows the user to change the form style

2. **Get the payment status:** gets the checkout status
    - Once the payment is processed, the customer is redirect to your redirectUrl (defined in step 2.1). You can check the status of your transaction making a GET request
3. **Merchant Notification:** Receive payment status
    - SIBS Payment Gateway will send the notification of the asynchronous payments status.

Note: To be a valid Merchant on SIBS Gateway V2 will be necessary to have an authorisation token (<AuthToken>), <terminalId> and <x-ibm-client-id>. The SIBS Gateway V2 Form is provided in Portuguese (PT), English (EN), Spanish (ES), Deutsch (DE), French (FR) and Italian (IT).

Server-to-Server

The Server to Server integration is also known as direct integration as it enables communication between the two servers, SIBS and Merchant. Cardholders can finalise a payment without being redirected to the SIBS Gateway V2 Form. This integration is suitable for merchants that store the payment data before sending them to the SIBS Payment Gateway and follow the PCI DSS compliance requirements.

Financial operations

For financial operations, Server-to-Server Integration is based on the following steps:

1. **Prepare the checkout:** sends payment data, except payment method data.

    First, perform a server-to-server POST request to prepare the checkout with the required data, including the order type, amount and currency. The response to a successful request is a JSON with a transactionID and a transactionSignature, which is required in the second step.

2. **Display payment method options to customer:**

    The next step is to provide the payment method options and get payment data from the customer:

    2.1. Card
    2.2. Multibanco
    2.3. MB WAY ID
    2.4. Authorised Payment

3. **Update payment method data:** sends payment method and its required data.

   Perform a payment request, call SIBS Gateway V2 API with the collected data accordingly with the method

4. **Get the payment status:** gets the payment status

   For MB WAY payments using mobile phone number, the GET status is an alternative procedure that is used in case a webhook is not received by the Merchant. This GET status must be called 5 minutes after the operation has been initialized.

5. **Merchant Notification:** Receive payment status

   SIBS Payment Gateway will send the notification of the asynchronous payments to the merchant server.

Authorised Payments

Additionally, Authorised Payment Server-to-Server Integration is available:

1. **Create Authorised Payment:** This step allows the Merchant to initiate an Authorised Payment creation;
2. **List Authorised Payments:** With List integration, Merchant can request the access to its list of available Authorised Payments;
3. **Inquiry Authorised Payment:** With Inquiry integration, Merchant can request the inquiry of a specific Authorised Payment;
4. **Cancel Authorised Payment:** With Cancel integration, Merchant can request the cancellation of a specific Authorised Payment.

Integration Flows

Card

This payment method allows the client to make a payment using their card.

This operation has duplicate control so that in the event that two consecutive purchases occur in the same logical terminal of the Merchant, for the same amount and the same card, the second purchase will be considered a duplicate and will be rejected.

- **Authorisation/Purchase**

  The image below shows the card authorisation/purchase flow, between the Merchant and SIBS Gateway V2.

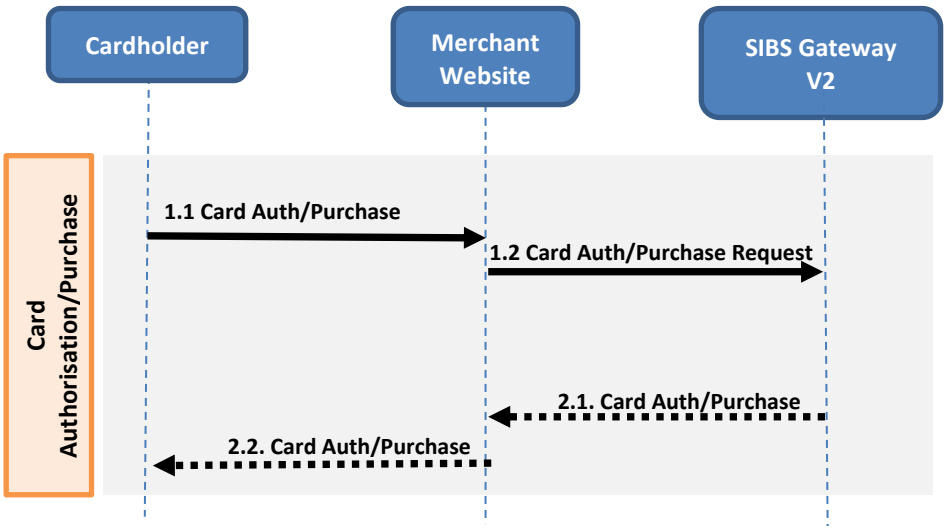

FIGURE 2 – CARD AUTHORISATION/PURCHASE

1.1 – Cardholder input card information in the merchant website

1.2 – Merchant website sends card information to the SIBS Gateway V2

2.1 – SIBS Gateway V2 returns the purchase result to the merchant website

2.2 – The merchant website displays the status of the payment to the cardholder

- **Capture**

  The image below shows the card capture flow, between the Merchant and SIBS

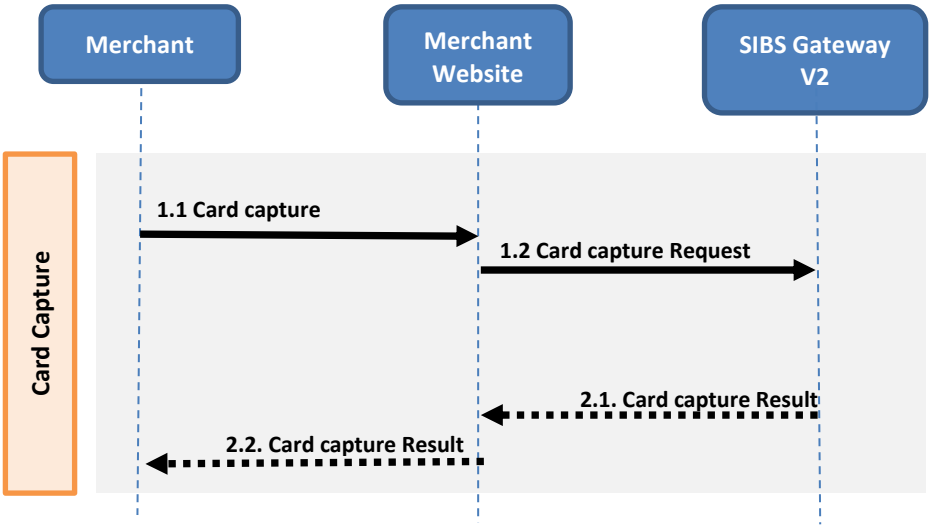

FIGURE 3 – CARD CAPTURE

1.1 – Merchant input capture information in the merchant website

1.2 – Merchant website sends the capture information to the SIBS Gateway V2

2.1 – SIBS Gateway V2 returns the capture result to the merchant website

2.2 – The merchant website displays the status of the capture

- **Authorisation Cancellation**

  The image below shows the card authorisation/Purchase flow, between the Merchant and SIBS

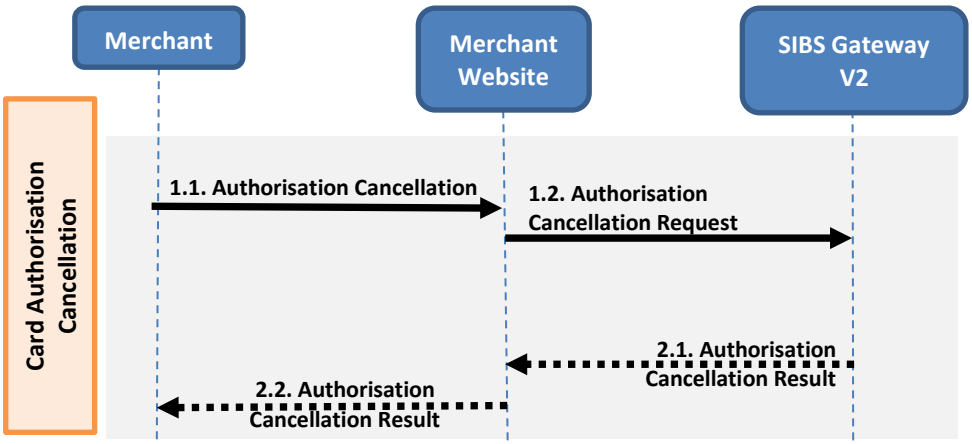

FIGURE 4 – CARD AUTHORISATION CANCELLATION

1.1 – Merchant inputs the Authorisation Cancellation in the Merchant Website

1.2 – Merchant Website sends the payment information to the SIBS Gateway V2

2.1 – SIBS Gateway V2 returns the cancellation result to the Merchant Website

2.2 – The Merchant Website returns the status of the cancellation

- **Purchase/Capture Refund**

After completing a Card Purchase, the Client may request the Refund of that Purchase. This operation can be triggered immediately.

This operation does not demand the use of the Client card. Nevertheless, in the case of partial Refunds a control is in place to prevent the refund to exceed the total amount of the original transaction.

The image below shows the card Purchase/Capture Refund flow, between the Merchant and SIBS:

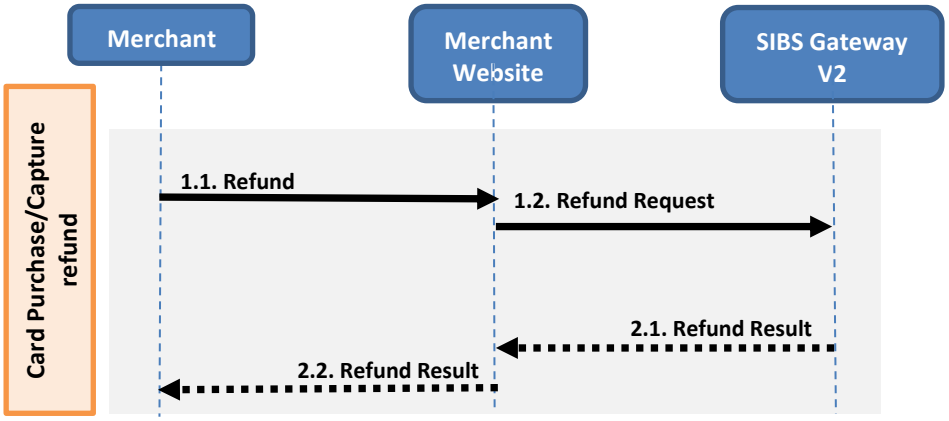

FIGURE 5 – CARD PURCHASE/CAPTURE REFUND

1.1 – Merchant input a refund request through the Backoffice or via API

1.2 – Merchant Website sends the refund request and details to the SIBS Gateway V2

2.1 – SIBS Gateway V2 returns the refund operation status result to the Merchant Website

2.2 – The Merchant Website display the status of the refund.

MB WAY (ID)

This operation allows the Client to make a Purchase using the MB WAY service. If the service is MB WAY ID, the Client alias must be informed (phone number) and an authorisation request will be sent to the MB WAY Client App.

SIBS Gateway V2 returns the result of the execution of the authorisation request. If this operation is successful, the transaction shall be considered to be in the “pending” status and the Merchant Website must wait for the result of the payment that will be informed through the notification that confirms the MB WAY Payment.

- **Authorisation/Purchase**

The picture below shows the MB WAY Authorisation/Purchase flow, between the Merchant and SIBS:

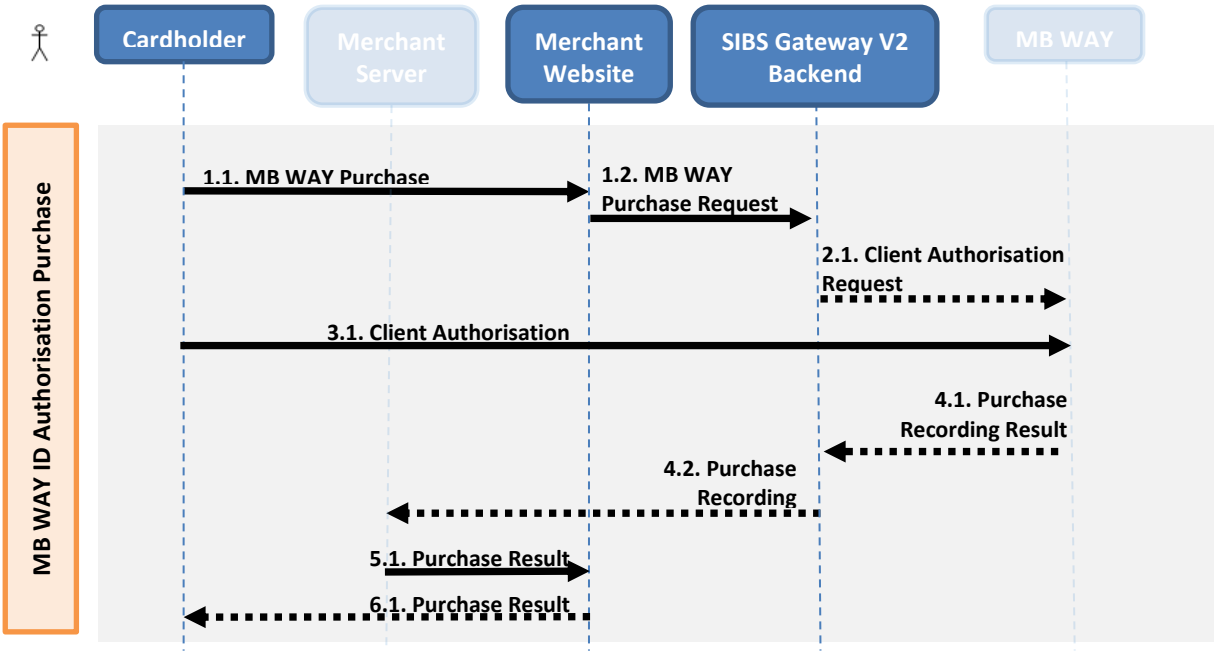

FIGURE 6 – MB WAY ID AUTHORISATION/ PURCHASE

1.1 – Cardholder input the alias in the Merchant Website

1.2 – Merchant Website sends the alias and payment information to the SIBS Gateway V2

2.1 – Client authorisation request sent to SIBS Gateway V2 (MB WAY)

3.1 – Cardholder should receive a push message to authorise, meanwhile payment stays pending

4.1 – Purchase result (Status) is sent from MB WAY to SIBS Gateway V2

4.2 – SIBS Gateway V2 will sent the Purchase result to the Merchant server

5.1 – The Merchant server will reconcile internally with the Merchant website

6.1 – Merchant website display to Cardholder the status of the payment.

- **MB WAY In-App Purchase**

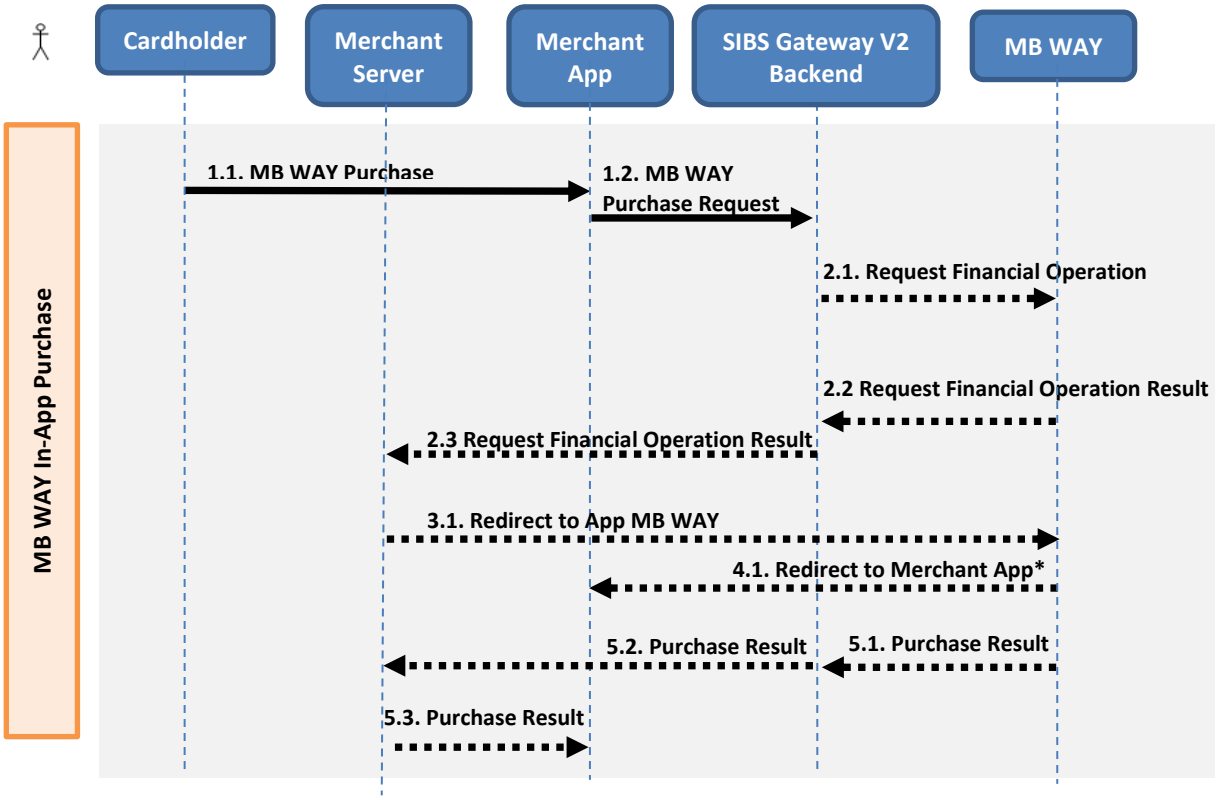

FIGURE 7 - MB WAY IN-APP PURCHASE

1.1 - Cardholder input the alias in the Merchant App;

1.2 - Merchant App sends the alias and payment information to the SIBS Gateway V2, including an InApp indicator with value "TRUE";

2.1 - MB WAY will receive the information regarding the In-App Operation;

2.2 - MB WAY sends response message with a MB WAY Redirect URL to SIBS Gateway V2;

2.3 - SIBS Gateway V2 receives message and sends it to the Merchant Server;

3.1 - Cardholder is redirected to the MB WAY App and is presented with the payment information;

4.1 - Merchant App displays the status of the payment to the Cardholder;

5.1 - Purchase result (Status) is sent from MB WAY to SIBS Gateway V2;

5.2 - SIBS Gateway V2 will sent the Purchase result to the Merchant server;

5.3 - The Merchant server will reconcile internally with the Merchant App.

*After receiving the MB WAY Redirect URL, the Merchant can include a Merchant URL inside the MB WAY Redirect URL so that, when the Cardholder accepts or declines the payment, it could be redirect to the Merchant App. If the Merchant does not include the Merchant URL, the Cardholder after the payment, will remain in MB WAY App and will have to return manually to the Merchant App to see payment status.

Example of a response message received by the Merchant with **inApp=”True”**:

{
"transactionID":"s26RxYwf7v2TxvHqjB1U",
"execution ":{
"startTime":"2023-07-18T08:21:57.995Z",
"endTime":"2023-07-18T08:21:58.632Z" },
"paymentStatus": "Pending ",
"returnStatus ":{
"statusCode":"000",
"statusMsg": "Success",
"statusDescription: "Success" },
"merchant ":{
"mbwayRedirectURL": "**mbway-qly://ACTDT?c =00100000000002386791**" }
}

The Merchant must add the Merchant URL to MB WAY Redirect URL as follows:

- “mbway-qly://ACTDT?c=00100000000002386791"**&u=[MerchantURL]”**

Note: If the cardholder has as default app the Homebanking app for MB WAY transactions, then when the redirect (with the following format:"@REDIRECT_APP_PWAY_181_4@00000000003188619454”) occurs it must be ignored and, instead, must accept the received push notification.

- **MB WAY (ID) Capture**

  The picture below shows the MB WAY Capture flow, between the Merchant and SIBS:

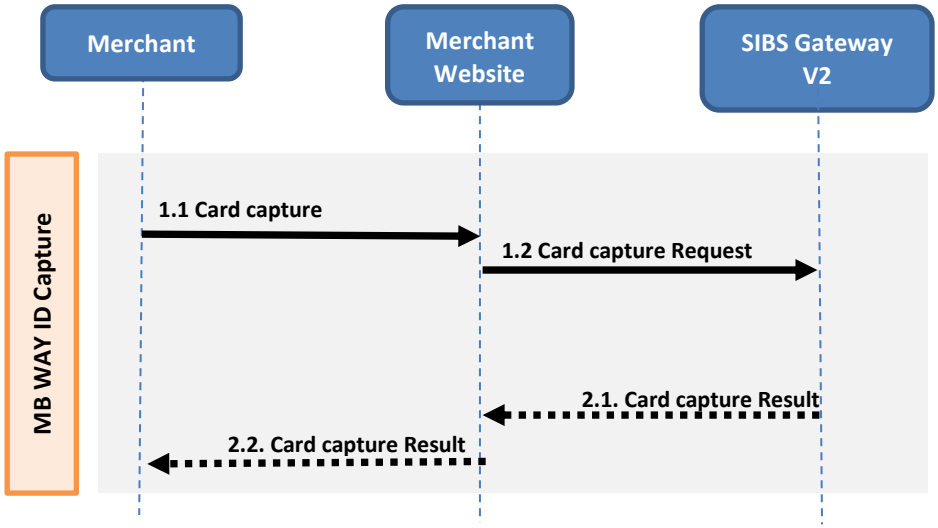

FIGURE 8 – MB WAY ID CAPTURE

1.1 – Merchant input capture information in the Merchant Website

1.2 – Merchant Website send the Capture information to the SIBS Gateway V2

2.1 – SIBS Gateway V2 returns the capture result to the Merchant Website

2.2 – The Merchant Website display the status of the capture.

- **MB WAY (ID) Authorisation Cancellation**

  The picture below shows the MB WAY Authorisation Cancellation flow, between the Merchant and SIBS:

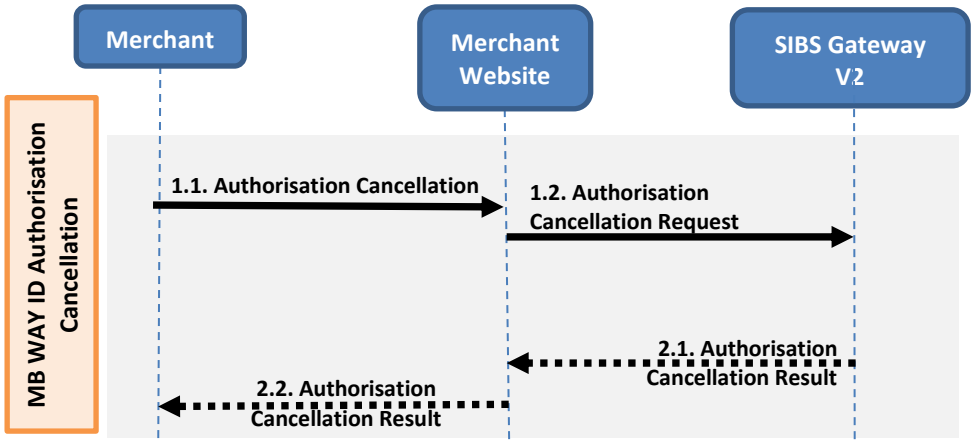

FIGURE 9 – MB WAY ID AUTHORISATION CANCELLATION

1.1 – Merchant input the Authorisation Cancellation in the Merchant Website

1.2 – Merchant Website sends the payment information to the SIBS Gateway V2

2.1 – SIBS Gateway V2 returns the cancellation result to the Merchant Website

2.2 – The Merchant Website returns the status of the cancellation

- **MB WAY (ID) Purchase/Capture Refund**

The below flow shows the MB WAY Purchase/Capture Refund flow, between the Merchant and SIBS:

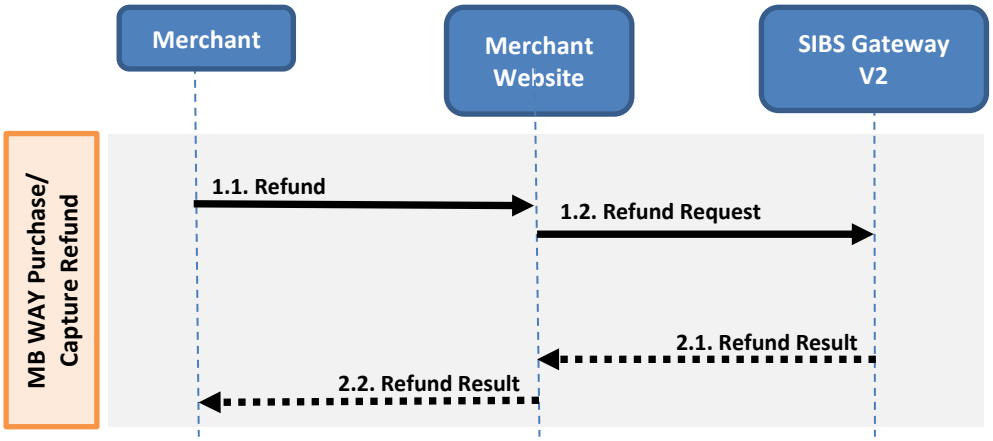

FIGURE 10 – MB WAY (ID) PURCHASE/CAPTURE REFUND

1.1 – Merchant input a refund request through the Backoffice or API

1.2 – Merchant Website sends the refund request and details to the SIBS Gateway V2

2.1 – SIBS Gateway V2 return the operation status to the Merchant Website

2.2 – Merchant Website returns the refund result to the Cardholder

Authorised Payment

These operations allow the Authorised Payment creation.

Authorised Payment Creation by Client

The picture below shows the Authorised Payment Creation flow, between the Client and SIBS:

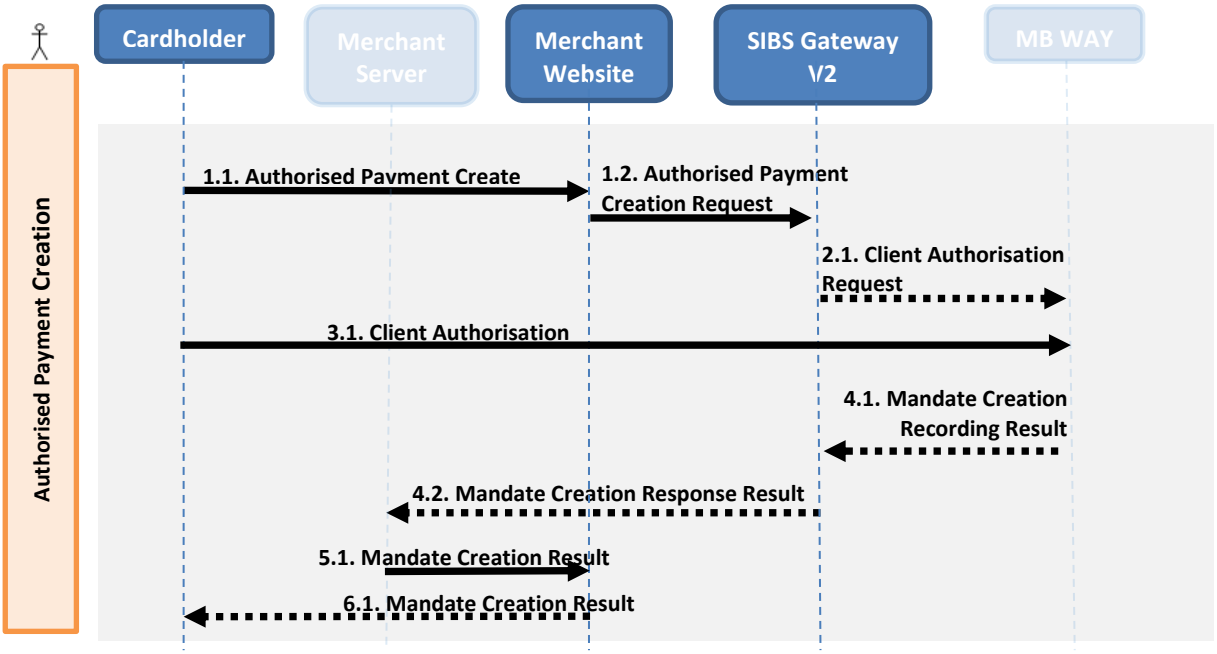

FIGURE 11 – AUTHORISED PAYMENT CREATION BY CLIENT

1.1 – Cardholder input the alias in the Merchant Website and accepts Authorised Payment Creation.

1.2 – Merchant Website sends the alias and Authorised Payment information to the SIBS Gateway V2

2.1 – Authorised Payment Creation request is sent to SIBS Gateway V2 (MB WAY).

3.1 – Cardholder should receive a push message to authorise, define monthly cap limit and expiration date.

4.1 – Authorised Payment creation details (e.g. Status) is sent from MB WAY to SIBS Gateway V2.

4.2 – SIBS Gateway V2 sends the Authorised Payment Creation result to the Merchant server.

5.1 – The Merchant server internally reconciles with the Merchant website.

6.1 – Merchant website displays to Cardholder the status of the creation.

Authorisation/Purchase under a valid Authorised Payment

The picture below shows the Authorisation/Purchase under a valid Authorised Payment flow, between the Merchant and SIBS:

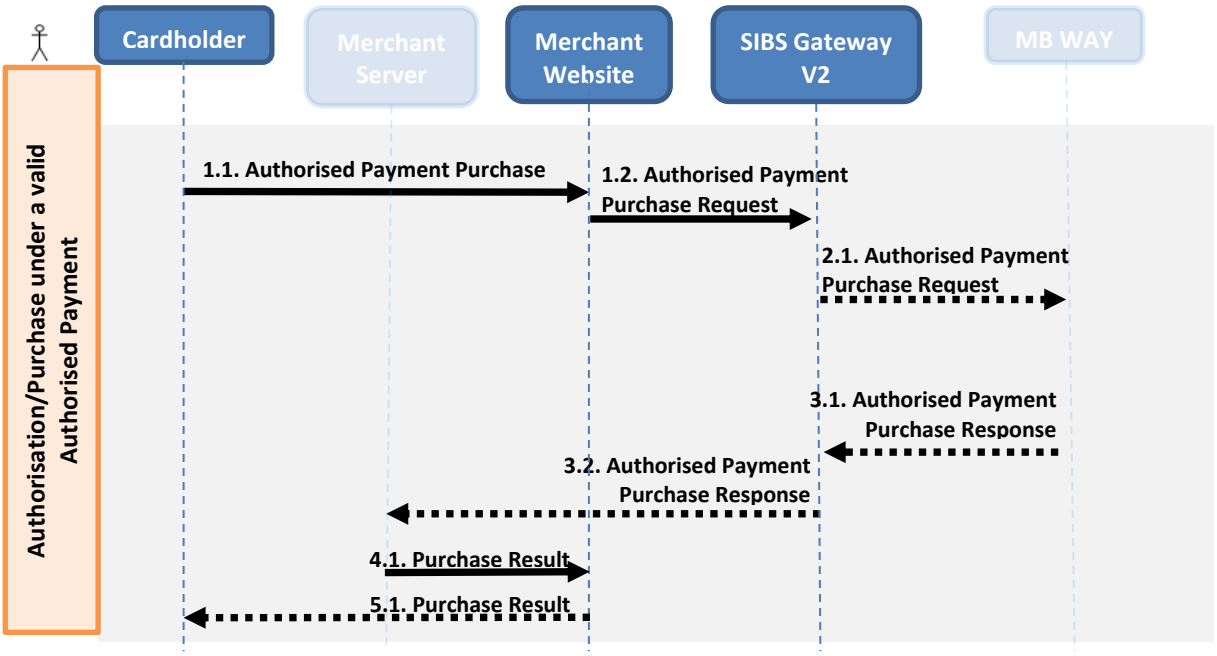

FIGURE 12 – AUTHORISED PAYMENT AUTHORISATION/ PURCHASE

1.1 – Cardholder input the alias in the Merchant Website.

1.2 – Merchant Website sends the alias and payment information to the SIBS Gateway V2.

2.1 – SIBS Gateway V2 sends Authorised Payment Purchase request to MB WAY after identifying the related Authorised Payment.

3.1 – Purchase result (Status) is sent from MB WAY to SIBS Gateway V2

3.2 – SIBS Gateway V2 sends the Purchase result to the Merchant server

4.1 – The Merchant server will reconcile internally with the Merchant website

5.1 – Merchant website display to Cardholder the status of the payment.

In point 2.1 of the flow, in case the transaction exceeds the monthly amount limit and the merchant has the “disableMandateMBWAYFallback” feature with value “False” or “Null”, MB WAY would make a MB WAY purchase with the same value of the transaction and will notify the merchant. If this feature has the value “True” then the operation is declined and a notification is sent to the client and to the merchant.

Capture

The picture below shows the Authorised Payment Capture flow, between the Merchant and SIBS:

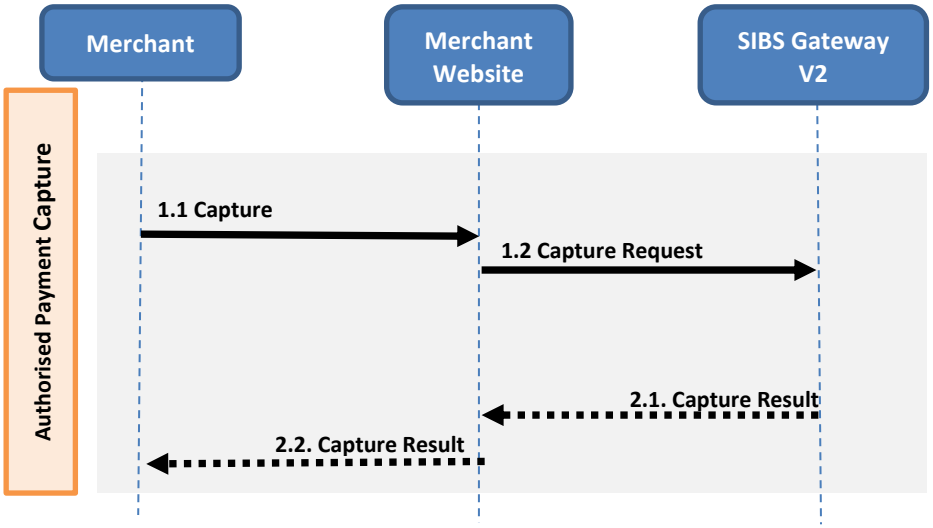

FIGURE 13 – AUTHORISED PAYMENT CAPTURE

1.1 – Merchant input capture information in the Merchant Website

1.2 – Merchant Website send the Capture information to the SIBS Gateway V2

2.1 – SIBS Gateway V2 returns the capture result to the Merchant Website

2.2 – The Merchant Website display to Cardholder the status of the capture.

Authorisation Cancellation

The picture below shows the Authorised Payment Authorisation Cancellation flow, between the Merchant and SIBS:

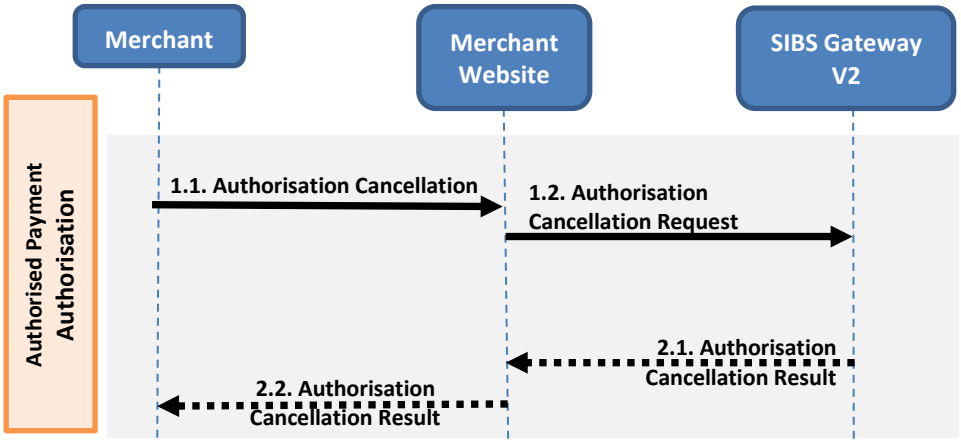

FIGURE 14 – AUTHORISED PAYMENT AUTHORISATION CANCELLATION

1.1 – Merchant input the Authorisation Cancellation in the Merchant Website

1.2 – Merchant Website sends the payment information to the SIBS Gateway V2

2.1 – SIBS Gateway V2 returns the cancellation result to the Merchant Website

2.2 – The Merchant Website returns the status of the cancellation

Purchase/Capture Refund

The below flow shows the Authorised Payment Purchase/Capture Refund flow, between the Merchant and SIBS:

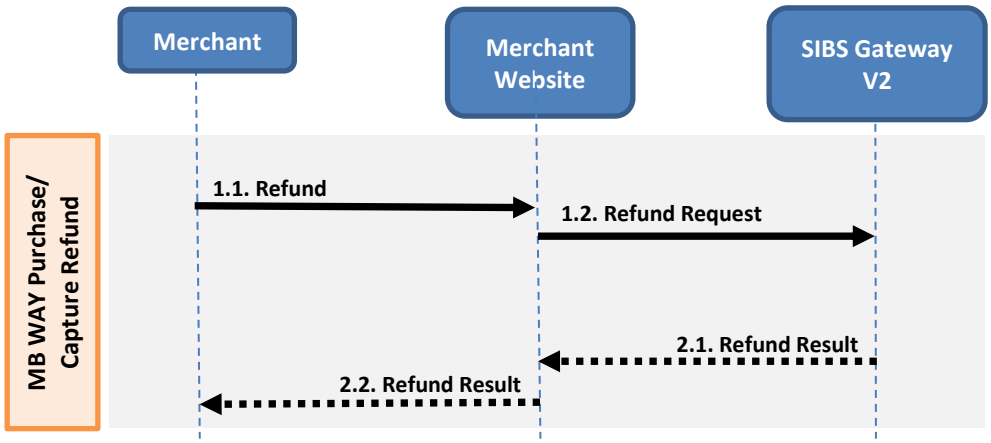

FIGURE 15 – AUTHORISED PAYMENT PURCHASE/CAPTURE REFUND

1.1 – Merchant input a refund request through the Backoffice or API

1.2 – Merchant Website sends the refund request and details to the SIBS Gateway V2

2.1 – SIBS Gateway V2 return the operation status to the Merchant Website

2.2 – Merchant Website returns the refund result to the Cardholder

Authorised Payment Creation by Merchant

The picture below shows the Authorised Payment Creation flow, between the Merchant and SIBS:

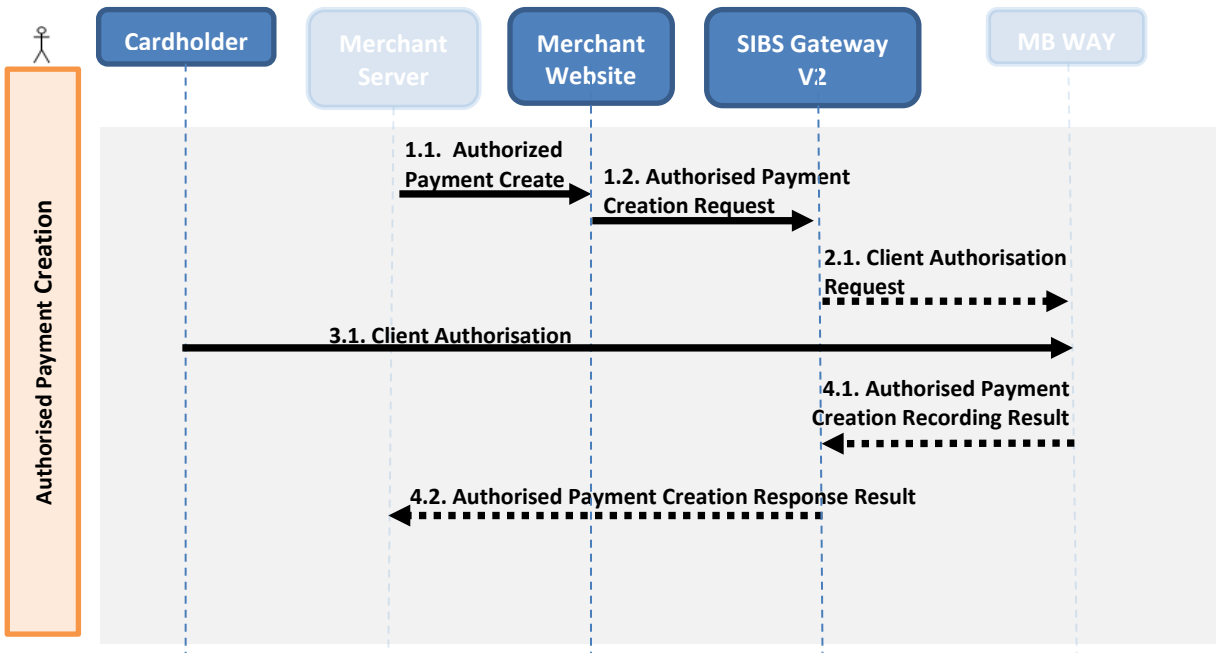

FIGURE 16 – AUTHORISED PAYMENT CREATION BY MERCHANT

- 1.1 – Merchant inputs the alias in the Merchant Server and accepts Authorised Payment Creation.
- 1.2 – Merchant Website sends the alias and Authorised Payment information to the SIBS Gateway V2
- 2.1 – Authorised Payment Creation request is sent to SIBS Gateway V2 (MB WAY).
- 3.1 – Cardholder should receive a push message to authorise, define monthly cap limit and expiration date.
- 4.1 – Authorised Payment creation details (e.g. Status) is sent from MB WAY to SIBS Gateway V2.
- 4.2 – SIBS Gateway V2 sends the Authorised Payment Creation result to the Merchant server.

Authorised Payment List

The following diagram shows the Authorised Payment List flow, between the Merchant and SIBS:

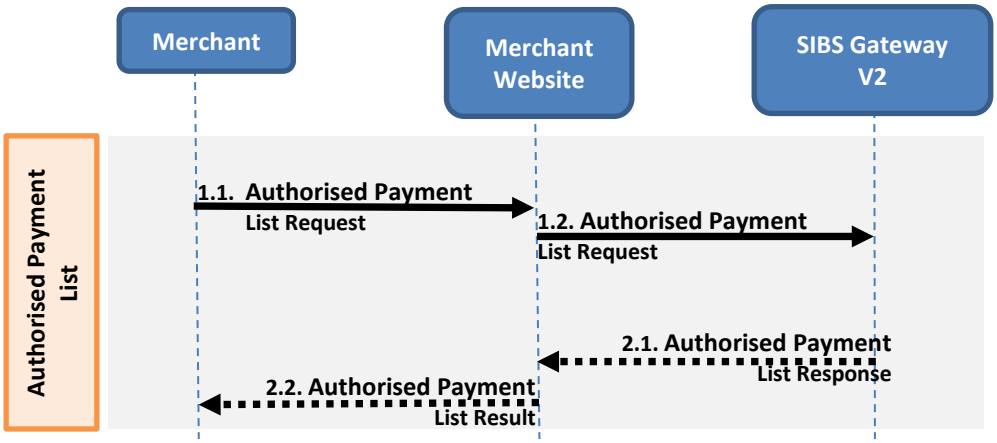

FIGURE 17 – AUTHORISED PAYMENT LIST

1.1 – Merchant requests Authorised Payment List, using the Backoffice or API.

1.2 – Merchant Website sends the list request to SIBS Gateway V2.

2.1 – SIBS Gateway V2 returns the Authorised Payment list to the Merchant Website. Pagination data may be included.

2.2 – Merchant Website presents list results.

Authorised Payment Inquiry

The following diagram shows the Authorised Payment Inquiry flow, between the Merchant and SIBS:

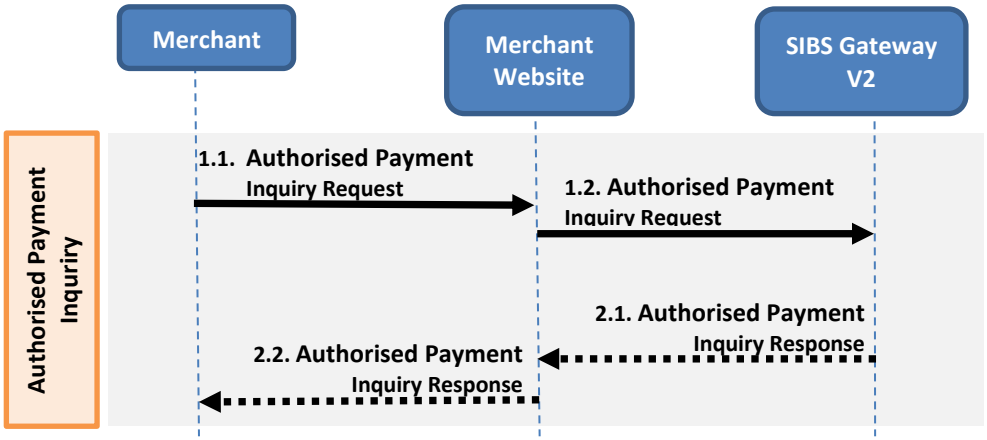

FIGURE 18 – AUTHORISED PAYMENT INQUIRY

1.1 – Merchant requests Authorised Payment Inquiry, using the Backoffice or API.

1.2 – Merchant Website sends the inquiry request to SIBS Gateway V2.

2.1 – SIBS Gateway V2 returns the Authorised Payment data to Merchant Website.

2.2 – Merchant Website presents Authorised Payment details.

Authorised Payment Inquiry Detail

The following diagram shows the Authorised Payment Inquiry detail flow, between the Merchant and SIBS where the Merchant requests for Financial Data of a Authorised Payment.

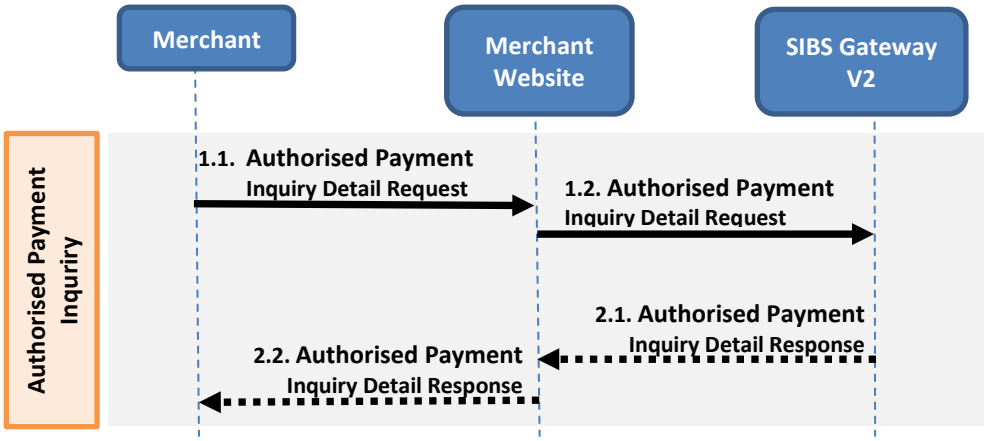

FIGURE 19 - AUTHORISED PAYMENT INQUIRY DETAIL

1.1 – Merchant requests Authorised Payment Inquiry Details, using the API.

1.2 – Merchant Website sends the inquiry detail request to SIBS Gateway V2.

2.1 – SIBS Gateway V2 returns the Authorised Payment Financial data to Merchant Website.

2.2 – Merchant Website presents Authorised Payment financial data.

Authorised Payment Cancellation

The following diagram shows the Authorised Payment Cancellation flow, between the Merchant and SIBS:

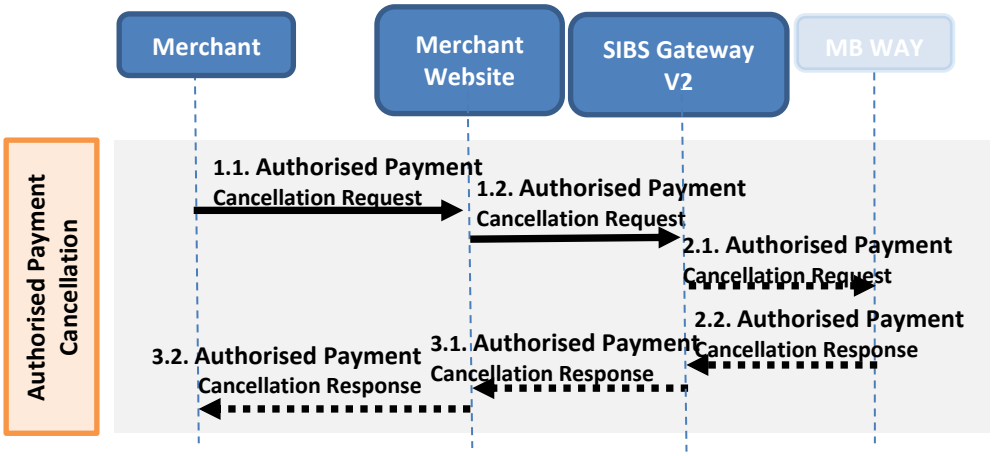

FIGURE 20 – AUTHORISED PAYMENT CANCELLATION

1.1 – Merchant inputs an Authorised Payment Cancellation request through the Backoffice or API.

1.2 – Merchant Website sends the Authorised Payment Cancellation request and to the SIBS Gateway V2.

2.1 – SIBS Gateway V2 sends Authorised Payment Cancellation request to MB WAY after identifying the related Authorised Payment.

2.2 – Authorised Payment Cancellation result (Status) is sent from MB WAY to SIBS Gateway V2.

3.1 – SIBS Gateway V2 returns the operation status to the Merchant Website.

3.2 – Merchant Website presents the cancellation result.

Multibanco

This operation allows the Client to make a Purchase by requesting a reference to do the payment through the generation of a MULTIBANCO reference. The Cardholder can use any MULTIBANCO Reference channel available (e.g. ATM, home banking, mobile banking or MB SPOT) to pay.

- **Generate/Purchase**

  The picture below shows the Multibanco flow, between the Merchant, SIBS & Cardholder:

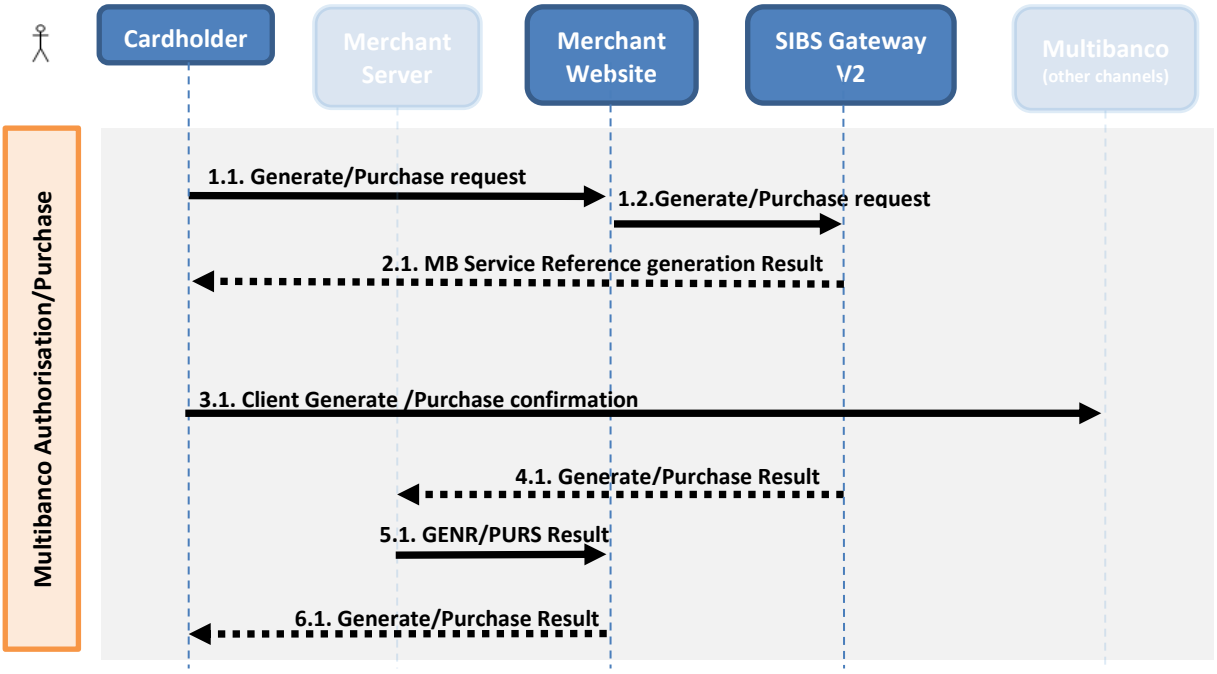

FIGURE 21 – MULTIBANCO (REFERENCE) GENERATE / PURCHASE

1.1 – Cardholder request the payment Multibanco reference in the Merchant Website
1.2 – Merchant Website sends the payment details to the SIBS Gateway V2
2.1 – SIBS Gateway V2 sends the Multibanco Service Reference to the Merchant Website
2.2 – SIBS Gateway V2 sends the reference to the Merchant Website (Waiting cardholder payment confirmation)
3.1 – Client proceeds to the payment and payment confirmation is sent to the SIBS Gateway V2
4.1 – SIBS Gateway V2 sends the Payment Authorisation Status to the Merchant
4.2 – The Merchant informs the Cardholder the status of the payment.

Note: A MB Reference number can be reuse however, the most recent MB Reference generated does not have any correlation with the same MB Reference number previously generated.

- **Reference Cancellation**
  The picture below shows the MB WAY Authorisation Cancellation flow, between the Merchant and SIBS:

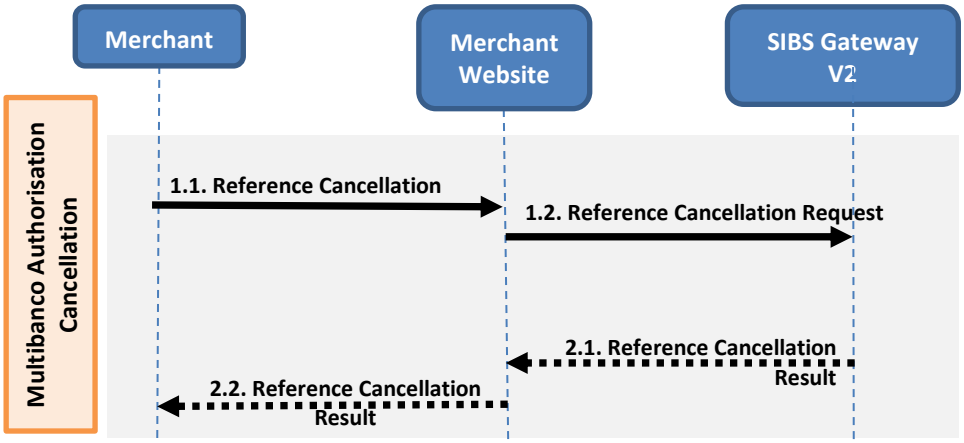

FIGURE 22 – MULTIBANCO REFERENCE CANCELLATION

1.1 – Merchant input the Reference Cancellation in the Merchant Website

1.2 – Merchant Website sends the payment information to the SIBS Gateway V2

2.1 – SIBS Gateway V2 returns the cancellation result to the Merchant Website

2.2 – The Merchant Website returns the status of the cancellation.

- **Purchase Refund**
  The below flow shows the MB WAY Purchase Refund flow, between the Merchant and SIBS:

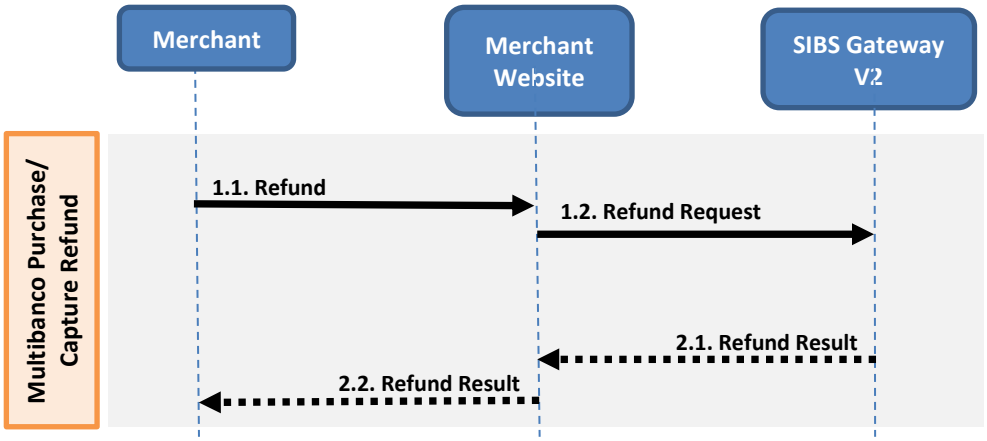

FIGURE 23 – MULTIBANCO PURCHASE/ REFUND

1.1 – Merchant input a refund request through the Backoffice or API

1.2 – Merchant Website sends the refund request and details to the SIBS Gateway V2

2.1 – SIBS Gateway V2 returns the operation status to the Merchant Website

2.2 – Merchant Website returns the refund result to the Cardholder

Cashout

This operation allows the merchant to send instant credit transfer (cashout) to their clients. The client will provide via API the MBWAY alias in order to receive the cashout.

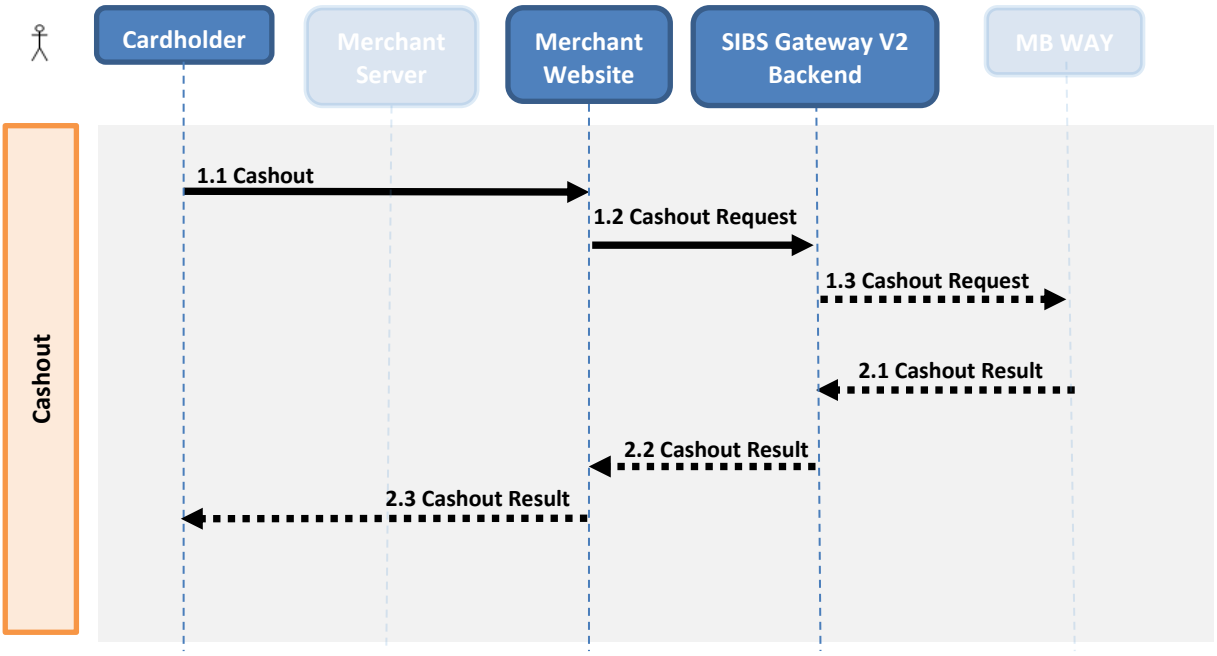

FIGURE 24 - CASHOUT

1.1 – Cardholder input the alias in the Merchant Website;

1.2 – Merchant Website sends the alias and Cashout information to the SIBS Gateway V2;

1.3 – SIBS Gateway V2 sends Cashout information to MB WAY;

2.1 – MB WAY processes the Cashout and sends a response message to SIBS Gateway V2;

2.2 – SIBS Gateway V2 will sent the Cashout result to the Merchant website;

2.3 – Merchant website display to Cardholder the status of the Cashout.

API's

- **Checkout / Get Status**

  Checkout API includes two operation:

  - **Checkout Payment** (POST <ROOT_URL>/api/v2/payments)
  - **Checkout Status**
    - (GET <ROOT_URL>/api/v2/payments/{id}/status) – {id} = transactionId

- (GET<ROOT_URL>/api/{v2}/payments/status) – using query parameter merchantTransactionId

Checkout Status API

Req. Header	Authorisation: 'Bearer* <*AuthToken*>' X-IBM-Client-Id: '<*clientid***>' Content-Type: 'application/json' Signature: 'string (Maximum128 Base64 Text)' (optional)

Response (with success)

Header	signature (optional)	string (Maximum 128 Text)	
Body	merchant	terminalId (optional)	integer (int32)
		Channel (optional)	string (Maximum 3 Text)
		merchantTransactionId (optional)	string (Maximum 35 Text)
	returnStatus	statusCode	string (Maximum 11 Text)
		statusMsg	string (Maximum 256 Text) (["Success", "Partial", "Declined", "InProcessing", "Pending", "Timeout", "Error"]).
		statusDescription	string (Maximum 256 Text)
	paymentType (optional)	string ("AUTH", "PURS", "CAPT", "CAUT", "RFND", "RCON", "RVSL", "STIQ", "PREF", "CPRF", "CMBW", "MAND", "MAUT", "MPUR", "MITR"])* * Some types are not applicable for this operation	

paymentStatus (optional)	string ([“Success”, “Partial”, “Declined”, “InProcessing”, “Pending”, “Timeout”, “Error”]) Present in success response
transactionStatusCode (optional)	string (Maximum 5 Text) Present in success response
transactionStatusDescription (optional)	string (Maximum 256 Text) Present in success response
transactionID (optional)	string (Maximum 35 Text) (“<*transactionID*>”)
amount (optional)	value: number (double) currency: string (ISO 4217 Alpha-3 Code)
paymentReference (optional)	reference (optional): string (Maximum 50 Text) entity (optional): string (Maximum 50 Text) paymentEntity (optional): string (Maximum 50 Text) amount (optional): value: number (double); currency: string (ISO 4217 Alpha-3 Code) expireDate (optional): string (date-time) status (optional): string ([“UNPAID”, “PAID”, “PARTPAIDCLS”, “PARTPAIDOPN”, “CANC”, “UNKN”])
merchantInitiatedTransaction (optional)	status: string ([“Success”, “Decline”, “Error”])
clientIBAN (optional)	string (Maximum 34 Text)
threeDSecure (optional)	whitelistStatus: string Possible values: • “Y” = 3DS Requestor is whitelisted by cardholder

			• "N" = 3DS Requestor is not whitelisted by cardholder • "E" = Not eligible as determined by issuer • "P" = Pending confirmation by cardholder • "R" = Cardholder rejected • "U" = Whitelist status unknown, unavailable, or does not apply.
tokenResponse (optional)	tokenName (optional)	string (Maximum 50 Text)	
	tokenType (optional)	string string ("TokenType1": "Email", "MobilePhone", "Card")	
	value (optional)	string (Maximum 50 Text)	
	maskedPAN (optional)	string (Maximum 23 Text)	
	expireDate (optional)	string (date-time)	
execution (optional)	startTime (optional)	string (date-time)	
	endTime (optional)	string (date-time)	

For more information about API's please follow the below link and choose SPG.

https://www.pay.sibs.com/solucoes/api-market/api-sibs-gateway/

Frontoffice

SIBS Gateway V2 Form Integration

Widget Endpoint: The <ROOT_URL> will depend of the environment where we are working

ROOT_URL QLY	https://api-qly.sibspayments.com
ROOT_URL PRD	https://api.sibspayments.com

To be a valid Merchant on SIBS Gateway V2 it will be necessary to have an authorisation token (<AuthToken>), <terminalId> and <x-ibm-client-id>.

Request API [POST] <*ROOT_URL*>**/api/v2/payments**

Req. Header	Authorisation: 'Bearer* <*AuthToken*>' X-IBM-Client-Id: '<*clientid***>' Content-Type: 'application/json'					
Req. Body	merchant(1) (optional)	terminalId (Mandatory)	integer (int32)			
		channel (Mandatory)	string (Maximum 3Text)			
		merchantTransactionId (optional)	string (Maximum 35 Text)			
	customer (Mandatory*) *Only for Card/Token Purchases	customerInfo (Mandatory*)	customerName (optional)	string (Maximum 45 Text)		
			customerEmail (Mandatory*)	string (Maximum 256 Text)		
			shippingAddress (optional)	street1 (optional)	string (Maximum 50 Text)	
				street2 (optional)	string (Maximum 19 Text)	
				city (optional)	string (Maximum 35 Text)	
				postcode (optional)	string (Maximum 16 Text)	
				countrySubDivision (optional)	string (Maximum 35 Text)	
				country (optional)	string (ISO 3166-1 alpha-2)	
			billingAddress (Mandatory*)	street1 (Mandatory*)	string (Maximum 50 Text)	

				street2 (optional)	string (Maximum 19 Text)
				city (Mandatory*)	string (Maximum 35 Text)
				postcode (Mandatory*)	string (Maximum 16 Text)
				countrySubDivision (optional)	string (Maximum 35 Text)
				country (Mandatory*)	string (ISO 3166-1 alpha-2)
		extendedInfo (optional)	key		string (Maximum 256 Text)
			value		string (Maximum 4096 Text)
	transaction[2] (optional)	transactionTimestamp (Mandatory)			string (date-time)
		description (optional)			string (Maximum 100 Text)
		moto (Mandatory)			boolean (["true", "false"])
		paymentType (Mandatory)			string(["AUTH", "PURS", "CAPT", "CAUT", "RFND", "RCON", "RVSL", "STIQ", "PREF", "CPRF", "CMBW", "MAND", "MAUT", "MPUR", "MITR"])
		paymentMethod (Mandatory)			string (["MBWAY", "REFERENCE", "CARD", "TOKEN", , "MANDATE","XPAY"])
		amount (Mandatory)	value		number (double)

				currency	string (ISO 4217 Alpha-3 Code)
			PaymentReference (Mandatory for MB Ref.)	entity	string (Maximum 50 Text)
				minAmount	value: number (double) currency: string (ISO 4217 Alpha-3 Code)
				maxAmount	value: number (double) currency: string (ISO 4217 Alpha-3 Code)
				initialDateTime	string (date-time)
				finalDateTime	string (date-time)
			threeDSecureOptions	authenticationExemption	string: "(["LOW_VALUE", ""NONE"])
			deviceInfo (optional)	browserAcceptHeader (optional)	string (Maximum 2048 Text)
				browserJavaEnabled (optional)	string (["True", "False"])
				browserJavascriptEnabled (optional)	string (["True", "False"])
		info (optional)		browserLanguage (optional)	string Value representing the browser language as defined in IETF BCP47
				browserColorDepth (optional)	string Required when Browser JavaScript Enable = True Possible values:

					(["1", "4", "8", "15", "16", "24", "32", "48"]) • 1 = 1 bit • 4 = 4 bits • 8 = 8 bits • 15 = 15 bits • 16 = 16 bits • 24 = 24 bits • 32 = 32 bits • 48 = 48 bits
				browserScreenHeight (mandatory)	string (Maximum 6 Numeric Text) Required when Browser JavaScript Enable = True
				browserScreenWidth (mandatory)	string (Maximum 6 Numeric Text) Required when Browser JavaScript Enabled = true
				browserTZ (optional)	string Required when Browser JavaScript Enabled = true Value is returned from the getTimezoneOffset() method Examples: If UTC -5 hours: (["300", "+300"]) If UTC +5 hours: (["-300"])
				browserUserAgent (optional)	string (Maximum 2048 Text)
				browserIp (optional)	string (Maximum 39 Text)
				systemFamily (optional)	string (Maximum 128 Text)
				systemVersion (optional)	string (Maximum 128 Text)
				systemArchitecture (optional)	string

					(Maximum 128 Text)
			deviceManufacturer (optional)		string (Maximum 35 Text)
			deviceModel (optional)		string (Maximum 35 Text)
			deviceID (optional)		string (Maximum 70 Text)
			applicationName (optional)		string (Maximum 100 Text)
			applicationVersion (optional)		string (Maximum 50 Text)
			geoLocalization (optional)		string (Maximum 256 Text)
			ipAddress (optional)		string (Maximum 39 Text)
		customerInfo (optional)	key* For Server-to-Server, The field "Key" must be filled with only uppercase letters and words must be separated with "_". (for example: Key = "POST_CODE")		string (Maximum 256 Text)
			value		string (Maximum 4096 Text)
		channelInfo (optional)	key		string (Maximum 256 Text)
			value		string (Maximum 4096 Text)
	merchantInitiatedTransaction[3] (optional)		type	string	(["UCOF", "RCRR"])
			validityDate (optional)	string	date-time
			amountQualifier (mandatory)	string	(["ACTUAL", "ESTIMATED", "DEFAULT"]) If not present, t"e "DEFAULT" value is assumed. If merchantInitiated

						Transaction has MITType "UCOF", the amountQualifier must be set to "ESTIMATED".
		Description (optional)	string			(Maximum 50 Text)
		schedule (optional)	initialDate (optional)	string	date-time	
			finalDate (optional)	string	date-time	
			interval (optional)	string	(["DAILY", "WEEKLY", "BIWEEKLY", "MONTHLY", "QUARTERLY", "SEMIANNUAL", "ANNUAL"])	
	tokenisation[4] (optional)	tokenisationRequest	tokeniseCard	boolean	([“true”, “false”])	
		paymentTokens	tokenType	string	([“Email”, “MobilePhone”, “Card”])	
			value	string	Maximum 50 Text	
	mandate[5] (optional)	mandateId (optional)	string	Maximum 64 Text		
		mandateType (optional)	string	(["ONECLICK", "SUBSCRIPTION”])		
		mandateCreationOnly (optional)	boolean	([“true”, “false”])		

* - If “Bearer” is a part of a header then it will not have limitations in terms of field size, If the filed isn’t a part of an header, the size must be Max1024Text.

** - If “clientId” is a part of a header then it will not have limitations in terms of field size, If the filed isn’t a part of an header, the size must be Max36Text.

TABLE 5 – SIBS GATEWAY V2 FORM INTEGRATION

- [1] **Merchant**: parameterization defined by the agreement between SIBS and the merchant
- [2] **Transaction**: parameterization related with the transaction, “moto” is in the case of being a transition “Mail Order Telephone Order”. The paymentType is ‘AUTH’ or ‘PURS’
- [3] **MerchantInitiatedTransaction**: optional parameterization to create a transaction with InitialCardholder Initiated Transaction.
- [4] **Tokenization**: optional parameterization to create a transaction with Tokenization

- [5] **Mandate:** In Authorised Payment creation, mandateId is not present and mandateType must indicate type (“ONECLICK" or "SUBSCRIPTION”). In this case MandateCreationOnly can also be present. When MandateCreationOnly is present "True"- Indicates that Only Creates Authorised Payment, ["False"] – Indicates that Creates Authorised Payment and requests payment.
- When Client requests a payment using an Authorised Payment previously created, mandateId must be filled with the Authorised Payment Identification. In this case merchantType and MandateCreationOnly are not present.

Response (with success)

Body of the response	returnStatus	statusCode	string (Maximum 11 Text)	
		statusMsg	string (["Success", "Partial", "Declined", "InProcessing", "Pending", "Timeout", "Error"]).	
	paymentMethodList[6]	string [] (["MBWAY", "REFERENCE", "CARD", "TOKEN", "STATIC_QRCODE", "MANDATE"])		
	merchant	terminalId	integer (int32) (“*<terminalId>*”)	
	amount	value	number (double)	
		currency	string (ISO 4217 Alpha-3 Code)	
	formContext[7]	string (Maximum 2048 Text) (“*<formContext>*”)		
	transactionID[5]	string (Maximum 35 Text recommend) (“*<transactionID>*”)		
	mandate[8] (optional)	mandateAvailable	boolean ([“true”, “false”])	
		termsAndConditions	string (Maximum 500 Text)	

- [5] **transactionID**: SIBS ID of transaction is used as the identifier of the transaction on the next APIs (<transactionID>)
- [6] **paymentMethodList**: Payment Methods available for the Merchant

- [7] **formContext**: Parameter to pass when building the SPG-form (Widget & Construction of SPG-form)
- [8]**mandate:** mandateAvailable is set to 'True' if current Client has an active Authorised Payment for current Merchant and is ready for use. Attribute termsAndConditions indicates Merchant URI with Authorized Payment Terms & Conditions.

Widget Construction

After a successfully request checkout, we can display the SPG-FORM, by adding a form with the class 'paymentSPG', and run *"ROOT_URL/assets/js/widget.js?id=<transactionID>"* script (Note that *ROOT_URL* is root URL depending on the environment). For the Merchant to have access to the SPG-FORM it is necessary to insert a form with the payment class in the DOM of Merchant HTML page with the following attributes:

- *attr.spg-context*: *formContext*[7] received from Checkout
- *attr.spg-config*: configurations defined by the merchant (spg-config)

```
<form class="paymentSPG"
    [attr.spg-context]="FORM-CONTEXT"
    [attr.spg-config]="SPG-CONFIG"
    [attr.spg-style]="SPG-STYLE">
</form>
```

Context: Information received from the API inserted on the FORM

Configuration: Where it is possible to do the general configuration

Style: Where it is possible to do a Customization of the Layout

Note: This form has three input attributes *attr.spg-context, attr.spg-config*, and *attr.spg-style*.

The attribute *attr.spg-context* should be the *formContext* from the Successful Checkout Response.

- **spg-config**

The attribute *attr.spg-config* is a JSON in a string format with the follow parameters:

PARAMETERS NAME	TYPE	AVAILABLE VALUES	DESCRIPTION
paymentMethodList	PaymentMethodList	'MBWAY' 'REFERENCE', 'CARD', 'TOKEN', , 'MANDATE'	Payment methods to show on the form. Payment methods that the merchant does not have permissions will not work.

redirectUrl	string (Maximum 2048 Text)	-	URL where the user should be redirected at the end of the checkout
amount	{ value: number, currency: string }	value with format double. Currency in ISO 4217 Alpha-3 Code.	Transaction amount value
language	ISO 3166	'pt' 'en' 'de' 'fr' 'it'	Form language
CustomerData	-	-	-

TABLE 6 – SPG CONFIGURATIONS

- **spg-style – Merchant Configurations**

The attribute *attr.spg-style* is a JSON string, optional allowing the merchant to customize the SPG-FORM:

PARAMETER	TYPE	POSSIBLE VALUES (EXAMPLES)	DEFAULT	DESCRIPTION
LAYOUT	string	'default', 'accordion', 'list'	'default'	SPG Form Layout
THEME	string	'default', 'light', 'dark', 'grey'	'default'	SPG Form Theme
COLOR.PRIMARY	string	'#ff0000'	-	Primary Color in Hexadeximal
COLOR.SECONDARY	string	'#ff0000'	-	Secondary Color in Hexadeximal
COLOR.BORDER	string	'#ff0000'	-	Border Color in Hexadeximal
COLOR.SURFACE	string	'#ff0000'	-	Surface Color in Hexadeximal - surface is the 'main background' of the html page
COLOR.HEADER.TEXT	string	'#ff0000'	-	Card Header Text Color in Hexadeximal
COLOR.HEADER.BACKGROUND	string	'#ff0000'	-	Card Header Color in Hexadeximal
COLOR.BODY.TEXT	string	'#ff0000'	-	Color in Hexadeximal of the Text in the Background
COLOR.BODY.BACKGROUND	string	'#ff0000'	-	Background Color in Hexadeximal - background

				is the background of the payment cards
FONT	string	'Times New Roman'	-	font Family used

TABLE 7 – SPG STYLE

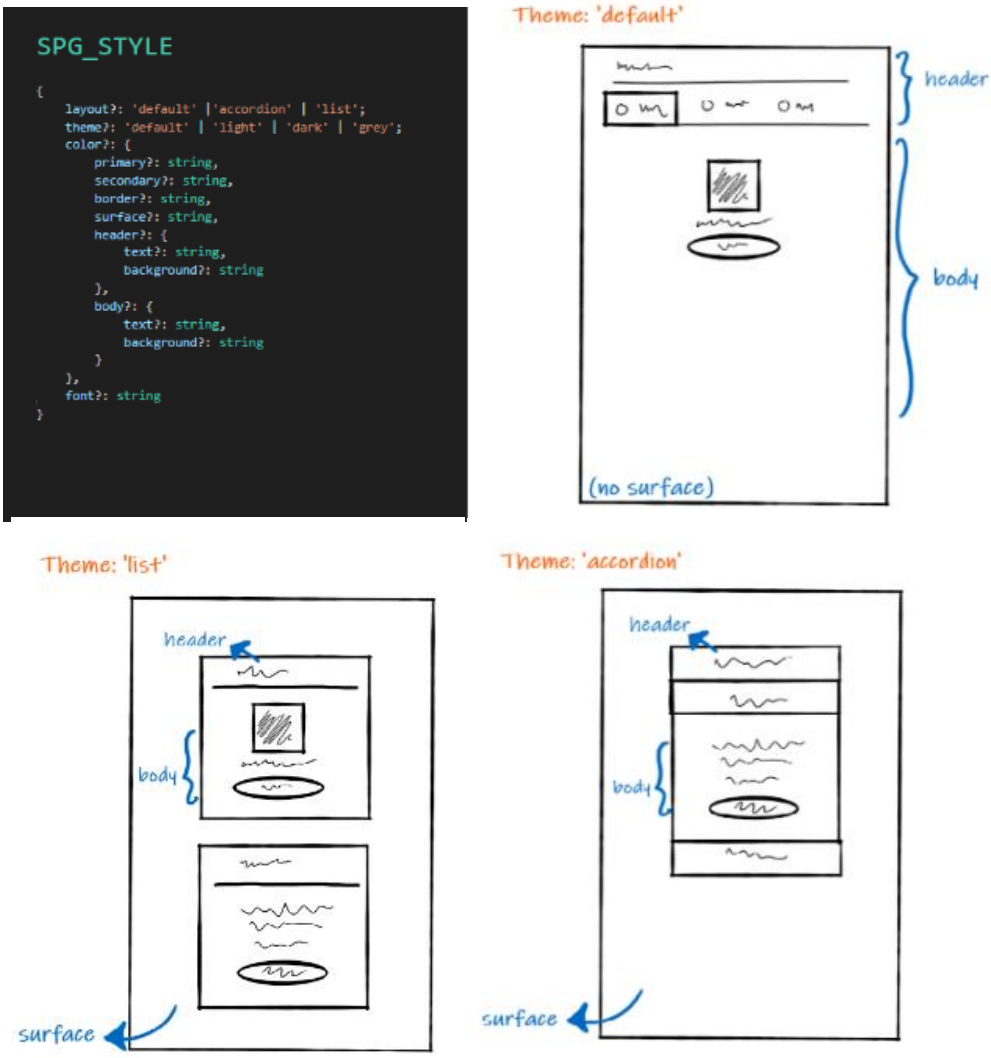

FIGURE 25 - SIBS GATEWAY V2 FORM STYLE EXAMPLE

- **Transaction Status Inquiry**

When the checkout is finished, the Merchant will be redirected to a URL (or to the root of his own website), and to check the transaction status needs to do a new request to the API (Transaction Status Inquiry).

API

[GET] <*ROOT_URL*>**/api/v2/payments/**<*transactionID*[5]>**/status**

Request

Header	Authorisation: 'Bearer <*AuthToken*> } X-IBM-Client-Id: '<*clientid*>'
Body of the request	{}

Response (with success)

Response Body	returnStatus	statusCode	string (Maximum 11 Text recommended)	
		statusMsg	string (Maximum 256 Text) (["Success", "Partial", "Declined", "InProcessing", "Pending", "Timeout", "Error"]).	
	merchant	terminalId	integer (int32) ("<*terminalId*>")	
	amount	value	number (double)	
		currency	string (ISO 4217 Alpha-3 Code)	
	paymentStatus[8]	string (Maximum 256 Text) (["Success", "Partial", "Declined", "InProcessing", "Pending", "Timeout", "Error"]).		
	transactionID	string (Maximum 35 Text) ("<*transactionID*>")		
	token[9]	string (Maximum 50 Text) ("<*token*>")		
	payment Reference[10]	entity	string (Maximum 50 Text)	
		reference	string (Maximum 50 Text)	
		amount	value	number (double)
			currency	string (ISO 4217 Alpha-3 Code)
		status	string (["UNPAID", "PAID", "PARTPAIDCLS", PARTPAIDOPN", "CANC", "UNKN"])	

- [8] **paymentStatus**: payment status of the transaction
- [9] **token**: In the cases that a token was requested, card token generated.
- [10] **paymentReference**: In the case that a payment with reference (Multibanco) has been requested, generate reference

Server to Server

Card

Card: (POST version-id/{id}/card/purchase)

API

[POST] <ROOT_URL>/api/v2/payments/<transactionId>/card/purchase

Request

Header	Authorisation: Digest {transactionSignature} X-IBM-Client-Id: '*<clientid>*' Content-Type: 'application/json' Signature: 'string (Maximum128 Base64 Text)' (optional)			
Body	cardInfo (optional)	PAN (optional)	string (Maximum 35 Text)	
		secureCode (optional)	string (Max4Numeric) (for AMEX Cards the CVV can be length 4)	
		validationDate (optional)	string (date-time)	
		cardholderName (optional)	string (Maximum 45 Text)	
		createToken (optional)	boolean	
	deviceInfo (optional)	browserAcceptHeader (optional)	string (Maximum 2048 Text)	
		browserJavaEnabled (optional)	string (["True", "False"])	
		browserJavascriptEnabled (optional)	string (["True", "False"])	
		browserLanguage (optional)	string Value representing the browser language as defined in IETF BCP47	
		browserColorDepth (optional)	string Required when Browser JavaScript Enable = True Possible values: (["1", "4", "8", "15", "16", "24", "32", "48"]) • 1 = 1 bit	

			• 4 = 4 bits • 8 = 8 bits • 15 = 15 bits • 16 = 16 bits • 24 = 24 bits • 32 = 32 bits • 48 = 48 bits
		browserScreenHeight (optional)	string (Maximum 6 Numeric Text) Required when Browser JavaScript Enable = True
		browserScreenWidth (optional)	string (Maximum 6 Numeric Text) Required when Browser JavaScript Enabled = true
		browserTZ (optional)	string Required when Browser JavaScript Enabled = true Value is returned from the getTimezoneOffset() method If UTC -5 hours: • 300 • +300 If UTC +5 hours: • -300
		browserUserAgent (optional)	string (Maximum 2048 Text)
		systemFamily (optional)	string (Maximum 128 Text)
		systemVersion (optional)	string (Maximum 128 Text)
		systemArchitecture (optional)	string (Maximum 128 Text)
		deviceManufacturer (optional)	string (Maximum 35 Text)
		deviceModel (optional)	string (Maximum 35 Text)
		deviceID (optional)	string (Maximum 70 Text)
		applicationName (optional)	string (Maximum 100 Text)
		applicationVersion (optional)	string (Maximum 50 Text)
		geoLocalization	string (Maximum 256 Text)

		(optional)	
		ipAddress (optional)	string (Maximum 39 Text)
	customerInfo (optional)	Key*	string (Maximum 256 Text) *Merchants must guarantee the integrity of the data. The field “Key” must be filled with only uppercase letters and words must be separated with “_”. (for example: Key = “POST_CODE”)
		value	string (Maximum 4096 Text)
	actionProcessed (optional)	Id (optional)	string (Maximum 50 Text)
		Type (optional)	string (["THREEDS_METHOD", "THREEDS_CHALLENGE", "DCC"]) *THREEDS_METHOD – value not used at the moment.
		executed (optional)	boolean ([“true”,- “false”])
	tokenInfo	tokenType (optional)	string (["Email", "MobilePhone", "QRCodeMBWAY", "Card")]
		value (optional)	string (Maximum 50 Text)
		secureCode (optional)	string (Max4Numeric) (for AMEX Cards the CVV can be length 4)
	instalmentplan	Id string (Maximum 3 Numeric Text)	
	merchantInitiatedTransaction (optional)	customerAcceptance* (Mandatory)	boolean ([“true”, “false”]) Set to 'True' if Merchant Initiated Transaction Terms and Conditions were presented by Merchant and explicitly accepted by Customer. Otherwise must set to 'False'. When not present, value 'False' is assumed.

Response (with success)

Header	signature (optional)	string (Maximum 128 Text)	
Body	merchant	terminalId (optional)	integer (int32)
		Channel (optional)	string (Maximum 3 Text)
		merchantTransactionId (optional)	string (Maximum 35 Text)
	returnStatus	statusCode	string (Maximum 11 Text)
		statusMsg	string (Maximum 256 Text) (["Success", "Partial", "Declined", "InProcessing", "Pending", "Timeout", "Error"]).
		statusDescription	string (Maximum 256 Text)
	paymentType (optional)	string (“AUTH”, “PURS”, “CAPT”, “CAUT”, “RFND”, “RCON”, “RVSL”, “STIQ”, “PREF”, “CPRF”, “CMBW”, “MAND”, “MAUT”, “MPUR”, “MITR”])* * Some types are not applicable for this operation	
	transactionID(optional)	string (Maximum 35 Text) (“*<transactionID>*”)	
	actionResponse (optional)	id (optional)	string (Maximum 50 Text)
		type (optional)	string ([“THREEDS_METHOD”, “THREEDS_CHALLENGE”, “DCC”]) *THREEDS_METHOD – value not used at the moment.
		data (optional)	url (optional) — String (Maximum 50 Text)
			params (optional) — parameter (optional) — name (optional) — String (Maximum 50 Text)
			data (optional) — string (Maximum 1024 Text)

	tokenResponse (optional)	tokenName (optional)	string (Maximum 50 Text)	
		tokenType (optional)	string string ("TokenType1": "Email", "MobilePhone", "Card")	
		value (optional)	string (Maximum 50 Text)	
		maskedPAN (optional)	string (Maximum 23 Text)	
		expireDate (optional)	string (date-time)	
	merchantInitiatedTransaction (optional)	status (Mandatory)	String (["Success", "Declined","Error"])	
	threeDSecure (optional)	whiteListStatus (Mandatory)	string (Exact 1Text) Y = 3DS Requestor is whitelisted by cardholder • N = 3DS Requestor is not whitelisted by cardholder • E = Not eligible as determined by issuer • P = Pending confirmation by cardholder • R = Cardholder rejected • U = Whitelist status unknown, unavailable, or does not apply.	
	execution (optional)	startTime	string (date-time)	
		endTime	string (date-time)	
	paymentStatus (optional)	string (["Success", "Partial", "Declined", "InProcessing", "Pending", "Timeout", "Error"])		
	signature (optional)	string (Maximum 128 Base64 Text)		

Token

Token: (POST/sibs/spg/{version-id}/payments/{id}/token/purchase)

API

[POST] <*ROOT_URL*> **/sibs/spg/{version-id}/payments/{OriginalTransactionId}/token/purchase**

Request

Header	Authorisation: Digest {transactionSignature} X-IBM-Client-Id: '<*clientid*>' Content-Type: 'application/json' Signature: 'string (Maximum128 Base64 Text)' (optional)		
Body	deviceInfo (optional)	browserAcceptHeader (optional)	string (Maximum 2048 Text)
		browserJavaEnabled (optional)	string (["True", "False"])
		browserJavascriptEnabled (optional)	string (["True", "False"])
		browserLanguage (optional)	string Value representing the browser language as defined in IETF BCP47
		browserColorDepth (optional)	string Required when Browser JavaScript Enable = True Possible values: (["1", "4", "8", "15", "16", "24", "32", "48"]) • 1 = 1 bit • 4 = 4 bits • 8 = 8 bits • 15 = 15 bits • 16 = 16 bits • 24 = 24 bits • 32 = 32 bits • 48 = 48 bits
		browserScreenHeight (optional)	string (Maximum 6 Numeric Text) Required when Browser JavaScript Enable = True
		browserScreenWidth (optional)	string (Maximum 6 Numeric Text) Required when Browser JavaScript Enabled = true
		browserTZ (optional)	string Required when Browser JavaScript Enabled = true Value is returned from the getTimezoneOffset() method If UTC -5 hours: • 300

			• +300 If UTC +5 hours: • -300
		browserUserAgent (optional)	string (Maximum 2048 Text)
		systemFamily (optional)	string (Maximum 128 Text)
		systemVersion (optional)	string (Maximum 128 Text)
		systemArchitecture (optional)	string (Maximum 128 Text)
		deviceManufacturer (optional)	string (Maximum 35 Text)
		deviceModel (optional)	string (Maximum 35 Text)
		deviceID (optional)	string (Maximum 70 Text)
		applicationName (optional)	string (Maximum 100 Text)
		applicationVersion (optional)	string (Maximum 50 Text)
		geoLocalization (optional)	string (Maximum 256 Text)
		ipAddress (optional)	string (Maximum 39 Text)
	customerInfo (optional)	Key*	string (Maximum 256 Text) *Merchants must guarantee the integrity of the data. The field "Key" must be filled with only uppercase letters and words must be separated with "_". (for example: Key = "POST_CODE")
		value	string (Maximum 4096 Text)
	actionProcessed (optional)	Id (optional)	string (Maximum 50 Text)
		Type (optional)	string (["THREEDS_METHOD", "THREEDS_CHALLENGE", "DCC"]) *THREEDS_METHOD – value not used at the moment.
		executed (optional)	boolean (["true",- "false"])
	tokenInfo	tokenType (optional)	string (["Email", "MobilePhone", "QRCodeMBWAY", "Card")]
		value	string (Maximum 50 Text)

		(optional)	
		secureCode (optional)	string (Max4Numeric) (for AMEX Cards the CVV can be length 4)
	merchantInitiatedTransaction (optional)	customerAcceptance* (Mandatory)	boolean ([“true”, “false”]) Set to 'True' if Merchant Initiated Transaction Terms and Conditions were presented by Merchant and explicitly accepted by Customer. Otherwise must set to 'False'. When not present, value 'False' is assumed.

Response (with success)

Header	signature (optional)	string (Maximum 128 Text)	
Body	merchant	terminalId (optional)	integer (int32)
		channel (optional)	string (Maximum 3 Text)
		merchantTransactionId (optional)	string (Maximum 35 Text)
	returnStatus	statusCode	string (Maximum 11 Text)
		statusMsg	string (Maximum 256 Text) (["Success", "Partial", "Declined", "InProcessing", "Pending", "Timeout", "Error"]).
		statusDescription	string (Maximum 256 Text)
	paymentType (optional)	string (“AUTH”, “PURS”, “CAPT”, “CAUT”, “RFND”, “RCON”, “RVSL”, “STIQ”, “PREF”, “CPRF”, “CMBW”, “MAND”, “MAUT”, “MPUR”, “MITR”])* * Some types are not applicable for this operation	
	transactionID(optional)	string (Maximum 35 Text) (“*<transactionID>*”)	
	actionResponse (optional)	id (optional)	string (Maximum 50 Text)

			type (optional)	string (["THREEDS_METHOD", "THREEDS_CHALLENGE", "DCC"]) *THREEDS_METHOD – value not used at the moment.			
			data (optional)	url (optional)	String (Maximum 50 Text)		
				params (optional)	parameter (optional)	name (optional)	String (Maximum 50 Text)
						data (optional)	string (Maximum 1024 Text)
		tokenResponse (optional)	tokenName (optional)	string (Maximum 50 Text)			
			tokenType (optional)	string string ("TokenType1": "Email", "MobilePhone", "Card")			
			value (optional)	string (Maximum 50 Text)			
			maskedPAN (optional)	string (Maximum 23 Text)			
			expireDate (optional)	string (date-time)			
		merchantInitiatedTransaction (optional)	status (Mandatory)	String (["Success", "Declined","Error"])			
		execution (optional)	startTime	string (date-time)			
			endTime	string (date-time)			
		paymentStatus (optional)	string (["Success", "Partial", "Declined", "InProcessing", "Pending", "Timeout", "Error"])				

MB WAY ID (mobile phone number)

MB WAY ID: (POST api/v2/payments/{id}/mbway-id/purchase)

API

[POST] <*ROOT_URL*>**/api/v2/payments/**< originalTransactionID [(5)]>**/mbway-id/purchase**

Request

<table>
<tr><td>Header</td><td colspan="3">Authorisation: Digest {transactionSignature}
X-IBM-Client-Id: ’<clientid>’
Content-Type: ‘application/json’
Signature: ‘string (Maximum128 Base64 Text)’ (optional)</td></tr>
<tr><td>Body</td><td>customerPhone</td><td colspan="2">string (Maximum 15 Text recommended) (ex:“351#910000000”)</td></tr>
<tr><td></td><td>inApp
(optional)</td><td colspan="2">boolean (“true”, “false”)</td></tr>
<tr><td></td><td>deviceInfo
(optional)</td><td>browserAcceptHeader
(optional)</td><td>string (Maximum 2048 Text)</td></tr>
<tr><td></td><td></td><td>browserJavaEnabled
(optional)</td><td>string
([“True”, “False”])</td></tr>
<tr><td></td><td></td><td>browserJavascriptEnabled
(optional)</td><td>string
([“True”, “False”])</td></tr>
<tr><td></td><td></td><td>browserLanguage
(optional)</td><td>string
Value representing the browser language as defined in IETF BCP47</td></tr>
<tr><td></td><td></td><td>browserColorDepth
(optional)</td><td>string
Required when Browser JavaScript Enable = True
Possible values:
([“1”, “4”, “8”, “15”, “16”, “24”, “32”, “48”])
• 1 = 1 bit
• 4 = 4 bits
• 8 = 8 bits
• 15 = 15 bits
• 16 = 16 bits
• 24 = 24 bits
• 32 = 32 bits
• 48 = 48 bits</td></tr>
<tr><td></td><td></td><td>browserScreenHeight
(optional)</td><td>string (Maximum 6 Numeric Text)
Required when Browser JavaScript Enable = True</td></tr>
<tr><td></td><td></td><td>browserScreenWidth
(optional)</td><td>string (Maximum 6 Numeric Text)
Required when Browser JavaScript Enabled = true</td></tr>
<tr><td></td><td></td><td>browserTZ
(optional)</td><td>string
Required when Browser JavaScript Enabled = true
Value is returned from the getTimezoneOffset() method
If UTC -5 hours:
• 300</td></tr>
</table>

			• +300 If UTC +5 hours: • -300
		browserUserAgent (optional)	string (Maximum 2048 Text)
		systemFamily (optional)	string (Maximum 128 Text)
		systemVersion (optional)	string (Maximum 128 Text)
		systemArchitecture (optional)	string (Maximum 128 Text)
		deviceManufacturer (optional)	string (Maximum 35 Text)
		deviceModel (optional)	string (Maximum 35 Text)
		deviceID (optional)	string (Maximum 70 Text)
		applicationName (optional)	string (Maximum 100 Text)
		applicationVersion (optional)	string (Maximum 50 Text)
		geoLocalization (optional)	string (Maximum 256 Text)
		ipAddress (optional)	string (Maximum 39 Text)
	customerInfo (optional)	Key	string (Maximum 256 Text)
		value	string (Maximum 4096 Text)

Response (with success)

Body	merchant	terminalId (optional)	integer (int32)
		Channel (optional)	string (Maximum 3 Text)
		merchantTransactionId (optional)	string (Maximum 35 Text)
		mbwayRedirectURL (optional)	string (Maximum 2048 Text)
	returnStatus	statusCode	string (Maximum 11 Text)

		statusMsg	string (Maximum 256 Text) (["Success", "Partial", "Declined", "InProcessing", "Pending", "Timeout", "Error"]).	
		statusDescription	string (Maximum 256 Text)	
	transactionID (optional)	string (Maximum 35 Text) (*"<transactionID>"*)		
	paymentStatus (optional)	string (["Success", "Partial", "Declined", "InProcessing", "Pending", "Timeout", "Error"])		
	signature (optional)	string (Maximum 128 Base64 Text)		

Authorised Payment

MB WAY ID: (POST <ROOT_URL>/api/v2/payments/{id}/mbway-id/purchase)

API

[POST] <*ROOT_URL*>**/api/v2/payments/**< originalTransactionID [(5)]>**/mbway-id/purchase**

Request

Header	Authorisation: Digest {transactionSignature} X-IBM-Client-Id: '<*clientid*>' Content-Type: 'application/json' Signature: 'string (Maximum 128 Base64 Text)' (optional)		
Body	mandate (optional)	mandateId	string (Maximum 64 Text)
		mandateCreation	boolean (["true", "false"])
		useMBWAYMandate	boolean (["true", "false"])
	deviceInfo (optional)	browserAcceptHeader (optional)	String (Maximum 2048 Text)
		browserJavaEnabled (optional)	string (["True","False"])
		browserJavascriptEnabled (optional)	string (["True","False"])
		browserLanguage	String

(optional)	Value representing the browser language as defined in IETF BCP47
browserColorDepth (optional)	string Required when Browser JavaScript Enable = True Possible values: (["1", "4", "8", "15", "16", "24", "32", "48"]) • 1 = 1 bit • 4 = 4 bits • 8 = 8 bits • 15 = 15 bits • 16 = 16 bits • 24 = 24 bits • 32 = 32 bits • 48 = 48 bits
browserScreenHeight (optional)	String (Maximum 6 Numeric Text) Required when Browser JavaScript Enable = True
browserScreenWidth (optional)	String (Maximum 6 Numeric Text) Required when Browser JavaScript Enabled = true
browserTZ (optional)	String Required when Browser JavaScript Enabled = true Value is returned from the getTimezoneOffset() method Possible values, example: If UTC -5 hours: • 300 • +300 If UTC +5 hours: • -300
browserUserAgent (optional)	String (Maximum 2048 Text)
systemFamily (optional)	String (Maximum 128 Text)

		systemVersion (optional)	String (Maximum 128 Text)
		systemArchitecture (optional)	String (Maximum 128 Text)
		deviceManufacturer (optional)	String (Maximum 35 Text)
		deviceModel (optional)	String (Maximum 35 Text)
		deviceID (optional)	String (Maximum 70 Text)
		applicationName (optional)	String (Maximum 100 Text)
		applicationVersion (optional)	String (Maximum 50 Text)
		geoLocalization (optional)	String (Maximum 256 Text)
		ipAddress (optional)	String (Maximum 39 Text)
	customerInfo (optional)	key	string (Maximum 256 Text)
		value	string (Maximum 4096 Text)

Response (with success)

Body	merchant	terminalId (optional)	integer (int32)	
		Channel (optional)	string (Maximum 3 Text)	
		merchantTransactionId (optional)	string (Maximum 35 Text)	
	returnStatus	statusCode	string (Maximum 11 Text)	
		statusMsg	string (Maximum 256 Text) (["Success", "Partial", "Declined", "InProcessing", "Pending", "Timeout", "Error"])	

		statusDescription: string (Maximum 256 Text)
	paymentType (optional)	string (“AUTH”, “PURS”, “CAPT”, “CAUT”, “RFND”, “RCON”, “RVSL”, “STIQ”, “PREF”, “CPRF”, “CMBW”, “MAND”, “MAUT”, “MPUR”, “MITR”])* * Some types are not applicable for this operation
	transactionID (optional)	string (Maximum 35 Text) (“<transactionID>”)
	paymentStatus	string ([“Success”, “Partial”, “Declined”, “InProcessing”, “Pending”, “Timeout”, “Error”])
	signature (optional)	string (Maximum 128 Base64 Text)

MULTIBANCO – MB Service Reference

In order to combat fraud associated with MB References, SIBS Gateway V2 will allow the merchant to fill more information about beneficiary of the payment (Sub-merchants). This information will be collected via **Checkout Service** and when available, will be presented to the end user by the platforms (ATM or Homebanking) and allow the payer to noticed more easily if the payment is suspicious of being a scam.

After being filled in the **Checkout Service** and the MB Reference is generated, the information related with the Sub-merchant will be aggregated with the MB Reference generated.

Below are represented the MB Reference/Sub-merchant fields that the merchant can fill in the **Checkout Service:**

Checkout Service		
Field Name and Condition	**Field Type**	
subMerchant (optional)	tin (optional)	string* (Maximum 20 Text) *This field is send in a string format, but must be filled with 20 digits or less
	name (optional)	string (Maximum 50 Text)
	mcc (optional)	integer (int4)
	sic (optional)	integer (int5)

paymentReference (optional)	paymentEntityDescription (optional)	string (Maximum 40 Text)	

TABLE 8 - SUBMERCHANT FIELDS - CHECKOUT

MB Service Reference API (POST <ROOT_URL>/api/v2/payments/{id}/service-reference/generate)

API

[POST] <*ROOT_URL*>**/api/v2/payments/**< originalTransactionID [5]>**/service-reference/generate**

Request

Header	Authorisation: Digest {transactionSignature} X-IBM-Client-Id: '<*clientid*>' Signature: 'String (Maximum 128 Base64 Text)' (optional)		
Body	deviceInfo (optional)	browserAcceptHeader (optional)	string (Maximum 2048 Text)
		browserJavaEnabled (optional)	string (["True", "False"])
		browserJavascriptEnabled (optional)	string (["True", "False"])
		browserLanguage (optional)	string Value representing the browser language as defined in IETF BCP47
		browserColorDepth (optional)	string Required when Browser JavaScript Enable = True Possible values: (["1", "4", "8", "15", "16", "24", "32", "48"]) • 1 = 1 bit • 4 = 4 bits • 8 = 8 bits • 15 = 15 bits • 16 = 16 bits • 24 = 24 bits • 32 = 32 bits • 48 = 48 bits
		browserScreenHeight	String (Maximum 6 Numeric

		(optional)	Text)Required when Browser JavaScript Enable = True
		browserScreenWidth (optional)	String (Maximum 6 Numeric Text) Required when Browser JavaScript Enabled = true
		browserTZ (optional)	string Required when Browser JavaScript Enabled = true Value is returned from the getTimezoneOffset() method Possible values, example: If UTC -5 hours: • 300 • +300 If UTC +5 hours: • -300
		browserUserAgent (optional)	String (Maximum 2048 Text)
		systemFamily (optional)	String (Maximum 128 Text)
		systemVersion (optional)	String (Maximum 128 Text)
		systemArchitecture (optional)	String (Maximum 128 Text)
		deviceManufacturer (optional)	String (Maximum 35 Text)
		deviceModel (optional)	String (Maximum 35 Text)
		deviceID (optional)	String (Maximum 70 Text)
		applicationName (optional)	String (Maximum 100 Text)
		applicationVersion (optional)	String (Maximum 50 Text)
		geoLocalization (optional)	String (Maximum 256 Text)
		ipAddress (optional)	String (Maximum 39 Text)

	customerInfo (optional)	key	string (Maximum 256 Text)
		value	String (Maximum 4096 Text)

Response (with success)

Body	merchant	terminalId (optional)	integer (int32)	
		channel (optional)	string (Maximum 3 Text)	
		merchantTransactionId (optional)	string (Maximum 35 Text)	
	returnStatus	statusCode	string (Maximum 11 Text)	
		statusMsg	string (Maximum 256 Text) (["Success", "Partial", "Declined", "InProcessing", "Pending", "Timeout", "Error"])	
		statusDescription	string (Maximum 256 Text)	
	paymentType (optional)	string ([“AUTH”, “PURS”, “CAPT”, “CAUT”, “RFND”, “RCON”, “RVSL”, “STIQ”, “PREF”, “CPRF”, “CMBW”, “MAND “MAUT”, “MPUR”, “MITR”])		
	paymentReference	reference (optional)	string (Maximum 50 Text)	
		Entity (optional)	string (Maximum 50 Text)	
		paymentEntity (optional)	string (Maximum 50 Text)	
		amount (optional)	value	number (double)
			currency	string (ISO 4217 Alpha-3 Code)
		expireDate	string (date-time)	

		status			string ([“UNPAID”, “PAID”, “PARTPAIDCLS”, “PARTPAIDOPN”, “CANC”, “UNKN”])	
	transactionID (optional)	string (Maximum 35 Text) (“<transactionID>”)				
	actionResponse (optional)	Id (optional)	string (Maximum 50 Text)			
		Type (optional)	string ([“THREEDS_METHOD”, “THREEDS_CHALLENGE”, “DCC”]) *THREEDS_METHOD – value not used at the moment.			
		Data (optional)	url (optional)	string (Maximum 50 Text)		
			params (optional)	parameter (optional)	name (optional)	String (Maximum 50 Text)
					data (optional)	string (Maximum 1024 Text)
	paymentStatus (optional)	string ([“Success”, “Partial”, “Declined”, “InProcessing”, “Pending”, “Timeout”, “Error”])				
	signature (optional)	string (Maximum 128 Base64 Text)				

Get Payment Modalities

This API is called, after the checkout, and when there are payment modalities associated to the Merchant. With this call, the payment modalities will be available at the Payment operation.

In Server-to-Server, this API must be called by the Merchant, however, if the Merchant has a form integration, then the form must be the one that triggers this API.

In form integration, this API will be hidden from the Merchant.

API - GET /sibs/spg/{version-id}/payment-modalities/{id}/list

Request

Header	Authorisation: Bearer <*Authtoken*> X-IBM-Client-Id: ’<*clientid*>’ Content-Type: ‘application/json’
Path	Id: <*TransactionId*>

Response (with success)

Body	returnStatus	statusCode	string (Maximum 11 Text)	
		statusMsg	string (Maximum 256 Text) (["Success", "Partial", "Declined", "InProcessing", "Pending", "Timeout", "Error"])	
		statusDescription	string (Maximum 256 Text)	
	paymentModalities (optional)	modalityCode	string (Maximum 3 Numeric Text)	
		modalityDescription	string (Maximum 200 Text)	
		BIN	Array of Numbers (Maximum 8 Numeric Text)	

Cashout

<table>
<tr><td>Header</td><td>Authorisation: Digest {transactionSignature}
X-IBM-Client-Id: ’clientid’
Content-Type: ‘application/json’
Signature: ‘string (Maximum128 Base64 Text)’ (optional)</td></tr>
<tr><td>Body</td><td>
<table>
<tr><td>alias
(optional)</td><td><table><tr><td>aliasName</td><td>string (Maximum 256 Text)</td></tr></table></td></tr>
<tr><td>originApplication
(optional)</td><td>integer (Exact 1 Numeric)
Possible Values:
'0' = UP! (not used), '1' = Merchant)
Default: '1'</td></tr>
<tr><td>cardInfo
(optional)</td><td><table><tr><td>PAN</td><td>string (Maximum 35 Text)</td></tr><tr><td>validationDate</td><td>string
(date-time)</td></tr></table></td></tr>
<tr><td>deviceInfo
(optional)</td><td>
<table>
<tr><td>browserAcceptHeader
(optional)</td><td>string (Maximum 2048 Text)</td></tr>
<tr><td>browserJavaEnabled
(optional)</td><td>string
([“True”, “False”])</td></tr>
<tr><td>browserJavascriptEnabled
(optional)</td><td>string
([“True”, “False”])</td></tr>
<tr><td>browserLanguage
(optional)</td><td>string
Value representing the browser language as defined in IETF BCP47</td></tr>
<tr><td>browserColorDepth
(optional)</td><td>string
Required when Browser JavaScript Enable = True
Possible values:
([“1”, “4”, “8”, “15”, “16”, “24”, “32”, “48”])
• 1 = 1 bit
• 4 = 4 bits
• 8 = 8 bits
• 15 = 15 bits
• 16 = 16 bits
• 24 = 24 bits
• 32 = 32 bits
• 48 = 48 bits</td></tr>
<tr><td>browserScreenHeight
(optional)</td><td>string (Maximum 6 Numeric Text)
Required when Browser JavaScript Enable = True</td></tr>
</table>
</td></tr>
</table>
</td></tr>
</table>

		browserScreenWidth (optional)	string (Maximum 6 Numeric Text) Required when Browser JavaScript Enabled = true
		browserTZ (optional)	string Required when Browser JavaScript Enabled = true Value is returned from the getTimezoneOffset() method If UTC -5 hours: • 300 • +300 If UTC +5 hours: • -300
		browserUserAgent (optional)	string (Maximum 2048 Text)
		systemFamily (optional)	string (Maximum 128 Text)
		systemVersion (optional)	string (Maximum 128 Text)
		systemArchitecture (optional)	string (Maximum 128 Text)
		deviceManufacturer (optional)	string (Maximum 35 Text)
		deviceModel (optional)	string (Maximum 35 Text)
		deviceID (optional)	string (Maximum 70 Text)
		applicationName (optional)	string (Maximum 100 Text)
		applicationVersion (optional)	string (Maximum 50 Text)
		geoLocalization (optional)	string (Maximum 256 Text)
		ipAddress (optional)	string (Maximum 39 Text)
	initiationMethod (optional)	integer (Exact 1 Numeric) Possible Values: '1' - 'MSISDN Alias' '2' - 'QRCode' Default - '1'	
	customerInfo (optional)	Key*	string (Maximum 256 Text) *Merchants must guarantee the integrity of the data. The

			field “Key” must be filled with only uppercase letters and words must be separated with “_”. (for example: Key = “POST_CODE”)
		value	string (Maximum 4096 Text)
	merchant (optional)	terminalId	integer (int32)
		merchantTransactionId	string (Maximum 35 Text)
		merchantBrandName	string (Maximum 40 Text)
		operationDescription	string (Maximum 80 Text)
	amount (optional)	value	number (double)
		currency	string (ISO 4217 Alpha-3 Code)
	billingProductType	string (Maximum 10 Text)	

Complex Types

Merchant

Merchant	terminalId	integer (int32)
	channel	string (Maximum 3 Text)
	merchantTransactionId	string (Maximum 35 Text)

ReturnStatus

ReturnStatus	statusCode	string (Maximum 11 Text)
	statusMsg	string (Maximum 256 Text) (["Success", "Partial", "Declined", "InProcessing", "Pending", "Timeout", "Error"])
	statusDescription	string (Maximum 256 Text)

PaymentReference

PaymentReference	reference	string (Maximum 50 Text)	
	entity	string (Maximum 50 Text)	
	paymentEntity	string (Maximum 50 Text)	
	amount	value	number (double)
		currency	string (ISO 4217 Alpha-3 Code)

ActionProcessed

ActionProcessed	id	string (Maximum 50 Text)
	type	string (["THREEDS_METHOD",

		"THREEDS_CHALLENGE" or "DCC"]) *THREEDS_METHOD – value not used at the moment.
	executed	boolean (["true", "false"])

ActionResponse

ActionResponse	id	string (Maximum 50 Text)	
	type	string ("THREEDS_METHOD", "THREEDS_CHALLENGE" or "DCC") *THREEDS_METHOD – value not used at the moment.	
	data	url	string (Maximum 50 Text)
		params	name: string (Maximum 50 Text) data: string (Maximum 1024 Text)

PaymentModalities

paymentModalities (optional)	modalityCode	string (Maximum 3 Numeric Text)	
	modalityDescription	string (Maximum 200 Text)	
	BIN	Array of Numbers (Maximum 8 Numeric Text)	

Backoffice

Merchant Initiated Transaction API

Merchant Initiated Transaction API (POST <ROOT_URL>/api/v2/payments/{original-tx-id}/mit)

When a successful transaction is generated as Cardholder Initiated Transaction [3], the Merchant will be able to do perform Merchant Initiated Transaction payments.

Note: The Recurring API was discontinued in this version. The Merchant Initiated Transaction API provides a way for Merchants to request recurring transactions.

API

[POST] <*ROOT_URL*>**/api/v2/payments/**< originalTransactionID [5]>**/mit**

Request

<table>
<tr><td>Header</td><td colspan="4">Authorisation: ‘Bearer <<i>AuthToken</i>>’
X-IBM-Client-Id: ’<<i>clientid</i>>’
Content-Type: ‘application/json’
Signature: string ‘Maximum 128 Base64 Text’ (optional)</td></tr>
<tr><td rowspan="12">Request Body</td><td>merchant</td><td colspan="3">merchant[1] do checkout</td></tr>
<tr><td rowspan="5">transaction</td><td>type</td><td colspan="2">string ([“RCRR”, “UCOF”])</td></tr>
<tr><td>transactionTimestamp (optional)</td><td colspan="2">string (date-time)</td></tr>
<tr><td>description (optional)</td><td colspan="2">string (Maximum 100 Text)</td></tr>
<tr><td>amount</td><td>value</td><td>number (double)</td></tr>
<tr><td>currency</td><td>string (ISO 4217 Alpha-3 Code)</td></tr>
<tr><td>originalTransaction[13]</td><td>id</td><td>string (Maximum 35 Text)<originalTransactionID></td></tr>
<tr><td rowspan="6">Customer (optional)</td><td>datetime (optional)</td><td>string (date-time)</td></tr>
<tr><td rowspan="5">customerInfo (optional)</td><td>customerName (optional)</td><td>string (Maximum 45 Text)</td></tr>
<tr><td>customerEmail (optional)</td><td>String (Maximum 256 Text)</td></tr>
<tr><td rowspan="3">shippingAddress</td><td>street1 (optional)</td><td>string (Maximum 50 Text)</td></tr>
<tr><td>street2 (optional)</td><td>string</td></tr>
</table>

					(Maximum 19 Text)
				city (optional)	string (Maximum 35 Text)
				postcode (optional)	string (Maximum 16 Text)
				country (optional)	string (ISO 3166-1 alpha-2)
			billingAddress	street1 (optional)	string (Maximum 50 Text)
				street2 (optional)	string (Maximum 19 Text)
				city (optional)	string (Maximum 35 Text)
				postcode (optional)	string (Maximum 16 Text)
				country (optional)	string (ISO 3166-1 alpha-2)
		extendedInfo (optional)	key	string (Maximum 256 Text)	
			value	string (Maximum 4096 Text)	

(13) *originalTransaction*: Original transaction id (Cardholder Initiated Transaction) to do future Merchant Initiated Transaction payments

Response (with success)

Response Body	merchant	terminalId	integer (int32)	
		Channel (optional)	string (Maximum 3 Text)	
		merchantTransactionId (optional)	string (Maximum 35 Text)	
		merchantTransactionTimestamp (optional)	string (Maximum 35 Text)	

	returnStatus	statusCode	string (Maximum 11 Text)	
		statusMsg	string (["Success", "Partial", "Declined", "InProcessing", "Pending", "Timeout", "Error"])* *Some status are not applicable	
		statusDescription	string (Maximum 256 Text)	
	amount (optional)	value: number (double) currency: string (ISO 4217 Alpha-3 Code)		
	transactionID (optional)	string (Maximum 35 Text) (“<transactionID>”)		
	transactionTimestamp (optional)	string (date-time)		
	transactionRecipientId (optional)	string (Maximum 50 Text)		
	paymentStatus (optional)	string ([“Success”, “Partial”, “Declined”, “InProcessing”, “Pending”, “Timeout”, “Error”])		
	signature (optional)	string (Maximum 128 Base64 Text)		

Capture API

Capture API (POST <ROOT_URL>/api/v2/payments/{original-tx-id}/capture)

When a success transaction happens from an authorisation is possible to execute a Capture.**API**

[POST] <*ROOT_URL*>**/api/v2/payments/**< originalTransactionID [(5)]>**/capture**

Request

Header	Authorisation: ‘Bearer <*AuthToken*> X-IBM-Client-Id: ’<*clientid*>’ Content-Type: ‘application/json’

	Signature: 'String (Maximum 128 Base64 Text)' (optional)					
Request Body	merchant	merchant[1] do checkout				
	transaction	transactionTimestamp (optional)	string (date-time)			
		description (optional)	string (Maximum 100 Text)			
		amount	value	number (double)		
			currency	string (ISO 4217 Alpha-3 Code)		
		originalTransaction[13]	id (optional)	string (Maximum 35 Text) <originalTransactionID>		
			datetime (optional)	string (date-time)		
	Customer (optional)	customerInfo (optional)	customerName (optional)	string (Maximum 45 Text)		
			customerEmail (optional)	String (Maximum 256 Text)		
			shippingAddress	street1 (optional)	string (Maximum 50 Text)	
				street2 (optional)	string (Maximum 19 Text)	
				city (optional)	string (Maximum 35 Text)	
				postcode (optional)	string (Maximum 16 Text)	
				country (optional)	string (ISO 3166-1 alpha-2)	
			billingAddress	street1 (optional)	string (Maximum 50 Text)	
				street2 (optional)	string (Maximum 19 Text)	
				city (optional)	string (Maximum 35 Text)	
				postcode (optional)	string (Maximum 16 Text)	
				country	string	

				(optional)	(ISO 3166-1 alpha-2)
		extendedInfo (optional)	key	string (Maximum 256 Text)	
			value	string (Maximum 4096 Text)	
	saleContext (optional)	splitPayment (optional)	Split (optional)	boolean ([“true”, “false”])	
			paymentNumber (optional)	string (Maximum 50 Text)	
			maxPayments (optional)	string (Maximum 50 Text)	

(13) *originalTransaction*: Original transaction id from the transaction to capture

Response (with success)

Response Body	merchant	terminalId (optional)	integer (int32)	
		channel (optional)	string (Maximum 3 Text)	
		merchantTransactionId (optional)	string (Maximum 35 Text)	
		merchantTransactionTimestamp (optional)	string (date-time)	
	returnStatus	statusCode	string (Maximum 11 Text)	
		statusMsg	string (["Success", "Partial", "Declined", "InProcessing", "Pending", "Timeout", "Error"])	
		statusDescription	string (Maximum 256 Text)	
	amount (optional)	value	number (double)	
		currency	string (ISO 4217 Alpha-3 Code)	
	transactionID	string (Maximum 35 Text) (“<transactionID>”)		

	(optional)	
	transactionTimestamp (optional)	string (date-time)
	transactionRecipientId (optional)	string (Maximum 50 Text)
	paymentStatus (optional)	string (["Success", "Partial", "Declined", "InProcessing", "Pending", "Timeout", "Error"])
	signature(optional)	string (Maximum 128 Base64 Text)

Cancellation API

Cancellation API (POST <ROOT_URL>/api/v2/payments/{original-tx-id}/cancellation)

When a success transaction happens from an authorisation is possible to execute a Cancellation.

API

[POST] <*ROOT_URL*>**/api/v2/payments/**< originalTransactionID [5]>**/cancellation**

Request

Header	Authorisation: 'Bearer <*AuthToken*>' X-IBM-Client-Id: '<*clientid*>' Content-Type: 'application/json' Signature: string 'Maximum 128 Base64 Text' (optional)			
Request Body	merchant	Same as merchant[1] do checkout		
	transaction	transactionTimestamp (optional)	string (date-time)	
		description (optional)	string (Maximum 100 Text)	
		amount (optional)	value	number (double)
			currency	string (ISO 4217 Alpha-3 Code)
		originalTransaction[13]	id (optional)	string (Maximum 35 Text) <originalTransactionID>
			datetime (optional)	string (date-time)

	Customer (optional)	customerInfo (optional)	customerName (optional)	string (Maximum 45 Text)	
			customerEmail (optional)	String (Maximum 256 Text)	
			shippingAddress	street1 (optional)	string (Maximum 50 Text)
				street2 (optional)	string (Maximum 19 Text)
				city (optional)	string (Maximum 35 Text)
				postcode (optional)	string (Maximum 16 Text)
				country (optional)	string (ISO 3166-1 alpha-2)
			billingAddress	street1 (optional)	string (Maximum 50 Text)
				street2 (optional)	string (Maximum 19 Text)
				city (optional)	string (Maximum 35 Text)
				postcode (optional)	string (Maximum 16 Text)
				country (optional)	string (ISO 3166-1 alpha-2)
		extendedInfo (optional)	key	string (Maximum 256 Text)	
			value	string (Maximum 4096 Text)	

(13) *originalTransaction*: Original transaction id from the transaction to cancel

Response (with success)

Response Body	merchant	terminalId (optional)	integer (int32)	
		channel (optional)	string	

			(Maximum 3 Text)
		merchantTransactionId (optional)	string (Maximum 35 Text)
		merchantTransactionTimestamp (optional)	string (date-time)
	returnStatus	statusCode	string (Maximum 11 Text)
		statusMsg	string (["Success", "Partial", "Declined", "InProcessing", "Pending", "Timeout", "Error"])
		statusDescription	string (Maximum 256 Text)
	amount (optional)	value	number (double)
		currency	string (ISO 4217 Alpha-3 Code)
	transactionID(optional)	string (Maximum 35 Text) (“<transactionID>”)	
	transactionTimestamp(optional)	string (date-time)	
	transactionRecipientId(optional)	string (Maximum 50 Text)	
	paymentStatus (optional)	string ([“Success”, “Partial”, “Declined”, “InProcessing”, “Pending”, “Timeout”, “Error”])	
	signature(optional)	string (Maximum 128 Base64 Text)	

Refund API

Refund API (POST <ROOT_URL>/api/v2/payments/{original-tx-id}/refund)

When a success transaction happens from a Capture or Purchase is possible to execute a Refund.

API

[POST] <*ROOT_URL*>**/api/v2/payments/<** originalTransactionID [5]**>/refund**

Request

Header	Authorisation: ‘Bearer <*AuthToken*>’ X-IBM-Client-Id: ’<*clientid*>’ Content-Type: ‘application/json’ Signature: string ‘Maximum 128 Base64 Text’ (optional)

Request Body	merchant	merchant[1] do checkout				
	transaction	transactionTimestamp (optional)	string (date-time)			
		description (optional)	string (Maximum 100 Text)			
		amount	value	number (double)		
			currency	string (ISO 4217 Alpha-3 Code)		
		originalTransaction[13]	id (optional)	string (Maximum 35 Text) <originalTransactionID>		
			datetime (Optional	string (date-time)		
	Customer (optional)	customerInfo (optional)	customerName (optional)	string (Maximum 45 Text)		
			customerEmail (optional)	String (Maximum 256 Text)		
			shippingAddress	street1 (optional)	string (Maximum 50 Text)	
				street2 (optional)	string (Maximum 19 Text)	
				city (optional)	string (Maximum 35 Text)	
				postcode (optional)	string (Maximum 16 Text)	
				country (optional)	string (ISO 3166-1 alpha-2)	
			billingAddress	street1 (optional)	string (Maximum 50 Text)	
				street2 (optional)	string (Maximum 19 Text)	
				city (optional)	string (Maximum 35 Text)	
				postcode (optional)	string (Maximum 16 Text)	
				country	string	

				(optional)	(ISO 3166-1 alpha-2)	
		extendedInfo (optional)	key	string (Maximum 256 Text)		
			value	string (Maximum 4096 Text)		

(13) *originalTransaction*: Original transaction id from the transaction to refund

Response (with success)

<table>
<tr><td>Response Body</td><td>merchant</td><td><table>
<tr><td>terminalId
(optional)</td><td>integer (int32)</td></tr>
<tr><td>Channel
(optional)</td><td>string
(Maximum 3 Text)</td></tr>
<tr><td>merchantTransactionId (optional)</td><td>string (Maximum 35 Text)</td></tr>
<tr><td>merchantTransactionTimestamp (optional)</td><td>string (date-time)</td></tr>
</table></td></tr>
<tr><td></td><td>returnStatus</td><td><table>
<tr><td>statusCode</td><td>string (Maximum 11 Text)</td></tr>
<tr><td>statusMsg</td><td>String (Maximum 256 Text)
([Success", "Partial", "Declined", "InProcessing", "Pending", "Timeout", "Error"])</td></tr>
<tr><td>statusDescription</td><td>string (Maximum 256 Text)</td></tr>
</table></td></tr>
<tr><td></td><td>amount
(optional)</td><td><table>
<tr><td>value</td><td>number (double)</td></tr>
<tr><td>currency</td><td>string (ISO 4217 Alpha-3 Code)</td></tr>
</table></td></tr>
<tr><td></td><td>transactionID
(optional)</td><td>string (Maximum 35 Text) (“<transactionID>”)</td></tr>
<tr><td></td><td>transactionTimestamp (optional)</td><td>string (date-time)</td></tr>
<tr><td></td><td>transactionRecipientId (optional)</td><td>string (Maximum 50 Text)</td></tr>
<tr><td></td><td>paymentStatus
(optional)</td><td>string ([“Success”, “Partial”, “Declined”, “InProcessing”, “Pending”, “Timeout”, “Error”])</td></tr>
<tr><td></td><td>signature(optional)</td><td>string (Maximum 128 Base64 Text)</td></tr>
</table>

Authorised Payment APIs

Create Authorised Payment API

Allows the Merchant to request Authorised Payment creation, related with the mobile phone number sent within the request and the mandate type.

After the creation of an Authorised Payment, SIBS Gateway V2 will send a notification to the Merchant informing the details of the Authorised Payment created (Authorised Payment Limits, Expiration Date, etc..).

API

[POST] *<ROOT_URL>***/api/v2/mbway-mandates/creation**

Request

Header	Authorisation: Digest {transactionSignature} X-IBM-Client-Id: '*<clientid>*' Content-Type: 'application/json'				
Request Body	merchant	terminalId	integer (int32)		
		channel	string (Maximum 3 Text)		
		merchantTransactionId	string (Maximum 35 Text)		
	mandate	mandateType	string ("ONECLIK", "SUBSCRIPTION")		
		aliasMBWAY	string (Maximum 256 Text)		
		customerName	string (Maximum 45 Text)		
	info	customerInfo (optional)	key	string (Maximum 256 Text)	
			value	string (Maximum 4096 Text)	
		deviceInfo (optional)	browserAcceptHeader (optional)	string (Maximum 2048 Text)	
			browserJavaEnabled (optional)	string (["True", "False"])	
			browserJavascriptEnabled (optional)	string (["True", "False"])	
			browserLanguage (optional)	string Value representing the browser language as defined in IETF BCP47	

			browserColorDepth (optional)	string Required when Browser JavaScript Enable = True Possible values: (["1", "4", "8", "15", "16", "24", "32", "48"]) • 1 = 1 bit • 4 = 4 bits • 8 = 8 bits • 15 = 15 bits • 16 = 16 bits • 24 = 24 bits • 32 = 32 bits • 48 = 48 bits
			browserScreenHeight (optional)	string (Maximum 6 Numeric Text) Required when Browser JavaScript Enable = True
			browserScreenWidth (optional)	string (Maximum 6 Numeric Text) Required when Browser JavaScript Enabled = true
			browserTZ (optional)	string Required when Browser JavaScript Enabled = true Value is returned from the getTimezoneOffset() method Possible values, example: If UTC -5 hours: • 300 • +300 If UTC +5 hours: • -300
			browserUserAgent (optional)	string (Maximum 2048 Text)
			systemFamily (optional)	string (Maximum 128 Text)
			systemVersion (optional)	string (Maximum 128 Text)

			systemArchitecture (optional)	string (Maximum 128 Text)
			deviceManufacturer (optional)	string (Maximum 35 Text)
			deviceModel (optional)	string (Maximum 35 Text)
			deviceID (optional)	string (Maximum 70 Text)
			deviceFingerprint (optional)	string (Maximum 512 Text)
			applicationName (optional)	string (Maximum 100 Text)
			applicationVersion (optional)	string (Maximum 50 Text)
			geolocation (optional)	string (Maximum 256 Text)
			ipAddress (optional)	string (Maximum 39 Text)

Response (with success)

Header				
Response Body	returnStatus	statusCode	string (Maximum 11 Text)	
		statusMsg	string (["Success", "Partial", "Declined", "InProcessing", "Pending", "Timeout", "Error"])	
		statusDescription	string (Maximum 256 Text)	
	transactionID	string (Maximum 35 Text)(“<transactionID>”)		
	transactionSignature	string (Maximum 256 Text)		
	mandate	mandateId	string (Maximum 64 Text)	
	execution	startTime	string (date-time)	
		endTime	string (date-time)	

List Authorised Payment API

Obtains the Merchant's Authorised Payment List.

API

[POST] *<ROOT_URL>***/api/v2/mbway-mandates/list**

Request

Header	Authorisation: Digest {transactionSignature} X-IBM-Client-Id: '*<clientid>*' Content-Type: 'application/json' Page-Number: string (Maximum 3 Numeric Text)				
Request Body	info	deviceInfo (optional)	browserAcceptHeader (optional)	string (Maximum 2048 Text)	
			browserJavaEnabled (optional)	string (["True", "False"])	
			browserJavascriptEnabled (optional)	string (["True", "False"])	
			browserLanguage (optional)	string Value representing the browser language as defined in IETF BCP47	
			browserColorDepth (optional)	string Required when Browser JavaScript Enable = True Possible values: (["1", "4", "8", "15", "16", "24", "32", "48"]) • 1 = 1 bit • 4 = 4 bits • 8 = 8 bits • 15 = 15 bits • 16 = 16 bits • 24 = 24 bits	

				• 32 = 32 bits • 48 = 48 bits
			browserScreenHeight (optional)	string (Maximum 6 Numeric Text) Required when Browser JavaScript Enable = True
			browserScreenWidth (optional)	string (Maximum 6 Numeric Text) Required when Browser JavaScript Enabled = true
			browserTZ (optional)	string Required when Browser JavaScript Enabled = true Possible values, example: If UTC -5 hours: • 300 • +300 If UTC +5 hours: • -300
			browserUserAgent (optional)	string (Maximum 2048 Text)
			systemFamily (optional)	string (Maximum 128 Text)
			systemVersion (optional)	string (Maximum 128 Text)
			systemArchitecture (optional)	string (Maximum 128 Text)

			deviceManufacturer (optional)	string (Maximum 35 Text)
			deviceModel (optional)	string (Maximum 35 Text)
			deviceID (optional)	string (Maximum 70 Text)
			deviceFingerprint (optional)	string (Maximum 512 Text)
			applicationName (optional)	string (Maximum 100 Text)
			applicationVersion (optional)	string (Maximum 50 Text)
			geolocation (optional)	string (Maximum 256 Text)
			ipAddress (optional)	string (Maximum 39 Text)

Response (with success)

Header				
Response Body	returnStatus	statusCode	string (Maximum 11 Text)	
		statusMsg	string ([Success", "Partial", "Declined", "InProcessing", "Pending", "Timeout", "Error"])	
		statusDescription	string (Maximum 256 Text)	
	mandate	mandateId	string (Maximum 64 Text)	
		mandateType	string (“ONECLICK”,”SUBSCRIPTION”)	
		customerName (optional)	string (Maximum 45 Text)	
		mandateStatus	string (“ACTV” - Active, “SSPN” - Suspended, ”EXPR”- Expired, ”CNCL” - Cancelled)	
		maskedPAN	string (Maximum 23 Text)	

		(optional)		
		aliasMBWAY	string (Maximum 256 Text)	
		mandateExpirationDate (optional)	string (date-time)	
		transactionId	string (Maximum 35 Text)	
	moreElementsIndicator (optional)	boolean		
	execution	startTime	string (date-time)	
		endTime	string (date-time)	

Get Authorised Payment API

Called by Merchant, to obtain a specific Authorised Payment related data.

API

[POST] <ROOT_URL>**/api/v2/mbway-mandates/**<originalTxId>**/inquiry**

Request

Header	Authorisation: Digest {transactionSignature} X-IBM-Client-Id: '*<clientid>*' Content-Type: 'application/json' Mbway-ID: 'String (Maximum 256 Text)'				
Request Body	info	deviceInfo (optional)	browserAcceptHeader (optional)	string (Maximum 2048 Text)	
			browserJavaEnabled (optional)	string (["True", "False"])	
			browserJavascriptEnabled (optional)	string (["True", "False"])	
			browserLanguage (optional)	string Value representing the browser language as defined in IETF BCP47	
			browserColorDepth (optional)	string Required when Browser JavaScript Enable = True Possible values: (["1", "4", "8", "15", "16", "24", "32", "48"]) • 1 = 1 bit • 4 = 4 bits • 8 = 8 bits • 15 = 15 bits • 16 = 16 bits • 24 = 24 bits • 32 = 32 bits • 48 = 48 bits	
			browserScreenHeight (optional)	String (Maximum 6 Numeric Text) Required when Browser JavaScript Enable = True	
			browserScreenWidth (optional)	String (Maximum 6 Numeric Text)	

				Required when Browser JavaScript Enabled = true
			browserTZ (optional)	string Required when Browser JavaScript Enabled = true Value is returned from the getTimezoneOffset() method Possible values, example: If UTC -5 hours: • 300 • +300 If UTC +5 hours: • -300
			browserUserAgent (optional)	string (Maximum 2048 Text)
			systemFamily (optional)	string (Maximum 128 Text)
			systemVersion (optional)	string (Maximum 128 Text)
			systemArchitecture (optional)	string (Maximum 128 Text)
			deviceManufacturer (optional)	string (Maximum 35 Text)
			deviceModel (optional)	string (Maximum 35 Text)
			deviceID (optional)	string (Maximum 70 Text)
			deviceFingerprint (optional)	string (Maximum 512 Text)
			applicationName (optional)	string (Maximum 100 Text)
			applicationVersion (optional)	string (Maximum 50 Text)
			geolocation (optional)	string (Maximum 256 Text)
			ipAddress (optional)	string (Maximum 39 Text)

Response (with success)

Header				
Response Body	returnStatus	statusCode	string (Maximum 11 Text)	
		statusMsg	string ([Success", "Partial", "Declined", "InProcessing", "Pending", "Timeout", "Error"])	
		statusDescription	string (Maximum 256 Text)	
	mandate	mandateId	string (Maximum 64 Text)	
		mandateType	string (“ONECLICK”, ”SUBSCRIPTION”)	
		customerName (optional)	string (Maximum 45 Text)	
		mandateStatus	string (“ACTV” - Active, “SSPN” - Suspended, “EXPR” - Expired,”CNCL” - Cancelled)	
		maskedPAN (optional)	string (Maximum 23 Text)	
		aliasMBWAY	string (Maximum 256 Text)	
		mandateExpirationDate (optional)	string (date)	
		amountLimit	value	number (double)
			currency	string (ISO 4217 Alpha-3 Code)
		transactionId	string (Maximum 35 Text)	
	execution	startTime	string (date-time)	
		endTime	string (date-time)	

Get Authorised Payment Financial Data API

Called by Merchant, to obtain a specific Authorised Payment Financial related data.

API

POST<ROOT_URL>/sibs/spg/{**v2**}/**mbway-mandates**/{originalTxId}/**inquiry-detail**

Request

Header	Authorisation: Digest {transactionSignature} X-IBM-Client-Id: '*<clientid>*' Content-Type: 'application/json' Mbway-ID: 'string (Maximum 256 Text)'				
Request Body	info	deviceInfo (optional)	browserAcceptHeader (optional)	String (Maximum 2048 Text)	
			browserJavaEnabled (optional)	string (["True", "False"])	
			browserJavascriptEnabled (optional)	string (["True", "False"])	
			browserLanguage (optional)	string Value representing the browser language as defined in IETF BCP47	
			browserColorDepth (optional)	string Required when Browser JavaScript Enable = True Possible values: (["1", "4", "8", "15", "16", "24", "32", "48"]) • 1 = 1 bit • 4 = 4 bits • 8 = 8 bits • 15 = 15 bits • 16 = 16 bits • 24 = 24 bits • 32 = 32 bits • 48 = 48 bits	
			browserScreenHeight (optional)	String (Maximum 6 Numeric Text) Required when Browser JavaScript Enable = True	
			browserScreenWidth (optional)	String (Maximum 6 NumericText)	

				Required when Browser JavaScript Enabled = true
			browserTZ (optional)	string Required when Browser JavaScript Enabled = true Value is returned from the getTimezoneOffset() method Possible values, example: If UTC -5 hours: • 300 • +300 If UTC +5 hours: • -300
			browserUserAgent (optional)	string (Maximum 2048 Text)
			systemFamily (optional)	string (Maximum 128 Text)
			systemVersion (optional)	string (Maximum 128 Text)
			systemArchitecture (optional)	string (Maximum 128 Text)
			deviceManufacturer (optional)	string (Maximum 35 Text)
			deviceModel (optional)	string (Maximum 35 Text)
			deviceID (optional)	string (Maximum 70 Text)
			deviceFingerprint (optional)	string (Maximum 512 Text)
			applicationName (optional)	string (Maximum 100 Text)
			applicationVersion (optional)	string (Maximum 50 Text)
			geolocation (optional)	string (Maximum 256 Text)
			ipAddress (optional)	string (Maximum 39 Text)

Response (with success)

Header				
Response Body	returnStatus	statusCode	string (Maximum 11 Text)	
		statusMsg	string (["Success", "Partial", "Declined", "InProcessing", "Pending", "Timeout", "Error"])	
		statusDescription	string (Maximum 256 Text)	
	mandate	mandateId	string (Maximum 64 Text)	
		mandateType	string ("ONECLICK", SUBSCRIPTION")	
		customerName (optional)	string (Maximum 45 Text)	
		mandateStatus	string ("ACTV" - Active, "SSPN" - Suspended, "EXPR" - Expired,"CNCL" - Cancelled)	
		maskedPAN (optional)	string (Maximum 23 Text)	
		aliasMBWAY	string (Maximum 256 Text)	
		mandateExpirationDate (optional)	string (date-time)	
		amountAvailable	value	number (double)
			currency	string (ISO 4217 Alpha-3 Code)
		amountLimit	value	number (double)
			currency	string (ISO 4217 Alpha-3 Code)
		transactionId	string (Maximum 35 Text)	
		lastTransactionDateTime (optional)	string (date-time)	
	execution	startTime	string (date-time)	
		endTime	string (date-time)	

Cancel Authorised Payment API

Called by Merchant, to cancel a specific Authorised Payment.

API

[POST] *<ROOT_URL>***/api/v2/mbway-mandates/**<originalTxId>**/cancel**

Request

Header	Authorisation: Digest {transactionSignature} X-IBM-Client-Id: '*<clientid>*' Content-Type: 'application/json' Mbway-ID: 'String (Maximum 256 Text)' Signature: 'string (Maximum Base64 128 Text)' (optional)				
Request Body	merchant	terminalId			integer (int32)
		channel (optional)			string (Maximum 3 Text)
		merchantTransactionId			String (Maximum 35 Text)
	info	customerInfo (optional)	key	String (Maximum 256 Text)	
			value	String (Maximum 4096 Text)	
		deviceInfo (optional)	browserAcceptHeader (optional)		string (Maximum 2048 Text)
			browserJavaEnabled (optional)		string (["True", "False"])
			browserJavascriptEnabled (optional)		string (["True", "False"])
			browserLanguage (optional)		string Value representing the browser language as defined in IETF BCP47
			browserColorDepth (optional)		string Required when Browser JavaScript Enable = True Possible values: (["1", "4", "8", "15", "16", "24", "32", "48"]) • 1 = 1 bit

				• 4 = 4 bits • 8 = 8 bits • 15 = 15 bits • 16 = 16 bits • 24 = 24 bits • 32 = 32 bits • 48 = 48 bits
			browserScreenHeight (optional)	string (Maximum 6 Numeric Text) Required when Browser JavaScript Enable = True
			browserScreenWidth (optional)	string (Maximum 6 Numeric Text) Required when Browser JavaScript Enabled = true
			browserTZ (optional)	string Required when Browser JavaScript Enabled = true Value is returned from the getTimezoneOffset() method Possible values, example: If UTC -5 hours: • 300 • +300 If UTC +5 hours: • -300
			browserUserAgent (optional)	string (Maximum 2048 Text)
			systemFamily (optional)	string (Maximum 128 Text)
			systemVersion (optional)	string (Maximum 128 Text)
			systemArchitecture (optional)	string (Maximum 128 Text)
			deviceManufacturer (optional)	string (Maximum 35 Text)
			deviceModel (optional)	string (Maximum 35 Text)
			deviceID (optional)	string (Maximum 70 Text)

			deviceFingerprint (optional)	string (Maximum 512 Text)
			applicationName (optional)	string (Maximum 100 Text)
			applicationVersion (optional)	string (Maximum 50 Text)
			geolocation (optional)	string (Maximum 256 Text)
			ipAddress (optional)	string (Maximum 39 Text)

Response (with success)

Header			
Response Body	returnStatus	statusCode	string (Maximum 11 Text)
		statusMsg	string (["Success", "Partial", "Declined", "InProcessing", "Pending", "Timeout", "Error"]).
		statusDescription	string (Maximum 256 Text)
	transactionID (optional)	string (Maximum 35 Text) (“<transactionID>”)	
	transactionSignature (optional)	string (Maximum 256 Text)	
	execution	startTime	string (date-time)
		endTime	string (date-time)
	signature(optional)	string (Maximum 128 Base64 Text)	

Data Dictionary

FIELD NAME	DATA TYPE	DESCRIPTION	EXAMPLE
AMOUNT	{value: number, currency: string }	Parameter with the value and currency of the transaction. value: Double. Integer part from one up to six digits. Decimal part up to two digits. Separator is character ‘.’.	{value: 1.23, currency: "EUR"}

FIELD NAME	DATA TYPE	DESCRIPTION	EXAMPLE
		currency: ISO 4217 Alpha-3 Code.	
AUTHENTICATIONEXEMPTION	string	Strong Customer Authentication Exemption types. Possible values are: “LOW_VALUE” – Low Value Amount Exemption; “NONE” – None;	[“LOW_VALUE”,“NONE”]
BILLINGADDRESS	Complex Type	Customer Billing Address	
BILLINGADRDRESSSTREET1	string (Maximum 50 Text)	Customer Billing Address Street.	
BILLINGADRDRESSSTREET2	string (Maximum 19 Text)	Customer Billing Address Street Additional Data.	
BILLINGADDRESSCITY	string (Maximum 35 Text)	Customer Billing Address City	
BILLINGADDRESSPOSTCODE	string (Maximum 16 Text)	Customer Billing Address Postal Code	
CHANNEL	String (Maximum 3 Numeric Text)	Merchant channel	"web"
CUSTOMERDATA	na	Customer general data - email, zip code	na
CUSTOMERNAME	string (Maximum 35 Text)	Customer Name	
CUSTOMEREMAIL	string (Maximum 256 Text)	Customer E-mail	
DESCRIPTION	string (Maximum 100 Text)	Transaction description	
ENTITY	String (Maximum 50 Text)	Entity used in Payment Reference	-
FINALDATETIME	string	Format: date-time	"2020-05-20T15:41:56.971Z"

FIELD NAME	DATA TYPE	DESCRIPTION	EXAMPLE
		Payment reference expiring date (if generated)	
FORMCONTEXT	String (Maximum2048 Text)	String given by Checkout Response, to be passed to the spg-form (see SIBS Payment Gateway Integration Guide)	na
INSTALMENTPLANID	String (Maximum 3 Numeric Text)	Instalment Plan Identification.	“123”
INTIALDATETIME	String	Format: date-time Date from when the payment reference is valid (if generated)	"2020-05-20T15:41:56.971Z"
LANGUAGE	String (Exact 2 Text)	Form language. ISO 639-1 Format.	
MAXAMOUNT	{value: number, currency: string}	value: double. Maximum amount of the transaction. currency: ISO 4217 Alpha-3 Code.	{ value: 1, currency: "EUR"}
MERCHANTTRANSACTIONID	string (Maximum 35 Text)	Id of the transaction in merchant store.	na
MERCHANTTRANSACTIONTIMESTAMP	string	Format: date-time	
MINAMOUNT	{value: number, currency: string}	value: double. Minimum amount of the transaction. currency: ISO 4217 Alpha-3 Code.	{ value: 1, currency: "EUR"}
MOTO	boolean	If it is a MOTO (Mail Order Telephone Order) [“true”, “false”]	false
ORIGINALTRANSACTION	string[] (Maximum 35 Text)	Original transaction of a Backoffice transaction	
PAYMENTMETHODLIST	string	List of payment methods. Possible values are: "MBWAY" – Alias MBWAY; "REFERENCE" – Payment Reference; "CARD" - Card; "TOKEN" - Token; "STATIC_QRCODE" – QR Code Express;	["MBWAY", "REFERENCE", "CARD","TOKEN", "STATIC_QRCODE",” MANDATE”]

FIELD NAME	DATA TYPE	DESCRIPTION	EXAMPLE
		”MANDATE” – Authorised Payment.	
PAYMENTREFERENCE	string	Payment reference details in case of customer select payment reference (MULTIBANCO) payment method	
PAYMENTTOKENS	{tokentype: string, value:string}[]	Customer payment tokens. These tokens are provided at the end of a successful. Possible values for tokenType are: “Email” “MobilePhone” “Card”	[{"tokenType": "Card", "value": "9uida0wd1211d12k09d12109"}]
PAYMENTTYPE	string	Transaction payment type. Possible values are: “PURS” – Purchase; “AUTH” – Authorisation; “CAPT” – Capture; “CAUT” – Authorisation Cancellation; “RFND” – Refund; “RCON” – Reconciliation; “RVSL” – Reversal; “STIQ” – Status Inquiry; “PREF” – Payment Reference Creation; “CPRF” – Payment Reference Cancellation; “CMBW” – MBWAY Cancellation; “MAND” – Authorised Payment Creation; “MPUR” – Purchase with Authorised Payment Creation; “MAUT” – Authorisation with Authorised Payment Creation; “MITR” – Merchant Initiated Transaction.	"AUTH"
SHIPPINGADDRESS	Complex Type	Customer Shipping Address	

FIELD NAME	DATA TYPE	DESCRIPTION	EXAMPLE
SHIPPINGADRDRESSSTREET1	string (Maximum 50 Text)	Customer Shipping Address Street.	
SHIPPINGADRDRESSSTREET2	string (Maximum 19 Text)	Customer Shipping Address Street Additional Data.	
SHIPPINGADDRESSCITY	string (Maximum 35 Text)	Customer Shipping Address City	
SHIPPINGADDRESSPOSTCODE	string (Maximum 16 Text)	Customer Shipping Address Postal Code	
MERCHANTINITIATEDTRANSACTION	{validityDate: string}	Format: date-time Provided field to create a MIT transaction. validity Date is the date until is possible to perform recurring transactions over the specified transaction	{ “RCRR”, "validityDate": "2021-06-24T15:41:56.971Z" }
REDIRECTURL	String (Maximum 256 Text)	Url where the user should to be redirected at the end of the checkout	
RETURNSTATUS		Describes the status of the request. (Attention: it doesn't describe the state of the transaction itself)	
TERMINALID	integer (int32)	format:int32 merchant terminal id	
TOKENISATIONREQUEST	{tokenizationCard: boolean}	Provided field on Checkout request to perform card tokenisation.	{"tokenizationCard": true}
TOKENIZATION		Field with tokenisation details (see tokenisation Request and paymentTokens)	
TRANSACTIONRECIPIENTID	String (Maximum 50 Text)	Original Transaction ID	
INAPP	Boolean	InApp Indicator	
MBWAYREDIRECTURL	String (Maximum 2048 Text)	MB WAY URL for redirect purposes to MB WAY App. Can include the MerchantURL, if inserted by the merchant, for	

FIELD NAME	DATA TYPE	DESCRIPTION	EXAMPLE
		redirect purposes to Merchant App	
AMOUNTLIMIT	number	Format: double Limit Amount of a Authorised Payment	
AMOUNTAVAILABLE	number	Format: double Amount Available of a Authorised Payment	
LASTTRANSACTIONDATETIME	string	Format: date-time DateTime of the last transaction made by a specific Authorised Payment	
SIGNATURE	String (Maximum 128 Base64 Text)	Hash Message Authentication Code. This field allows messages to have additional layer of security so that messages couldn't be compromised. At the moment, the type of HMAC used will be SHA2 (SHA512) This field must be formatted as Base64	
SIGNING STRING	String (Maximum 2048 Text)	String created with the concatenation of the necessary fields that must be coded through the HMAC feature.	
MODALITYCODE	String (Maximum 3 Numeric Text)	Code that identifies a payment modality	
MODALITYDESCRIPTION	String (Maximum 200 Text)	Description of the payment modality	
BIN	Array of Numbers (Maximum 8 Numeric Text)	Bank Identification Number	

TABLE 9 - DATA DICTIONARY

Inputs & Outputs

Checkout : Checkout

INPUT		OUTPUT	
merchant		returnStatus	
	• terminalId		• statusCode
	• channel		• statusMsg
	• merchantTransactionId		• statusDescription
customer		paymentStatus	
	• customerInfo	transactionID	
	• customerName	amount	
	• shippingAddress		• value
	• billingAddress		• currency
	• extendedInfo	merchant	• terminalId
Transaction			• channel
	• transactionTimestamp		• merchantTransactionId
	• description	PaymentType	
	• paymentType	paymentMethod	
	• paymentMethod	amount	
	• amount		• value
	• value		• currency
	• currency	Status	
deviceInfo* (see dedicated table)		tokenList	
customerInfo			tokenType
	• key		value
	• value		maskedPAN
originalTransaction			expireDate
	• id	formContext	
	• datetime	expiry	string (date-time)
	• recipientId	execution	
tokenisation			startTime
	• tokenisationRequest		endTime
	• tokeniseCard		
	• paymentTokens		
	• tokenType		
	• value		
mandate			

	• mandateId • mandateType • mandateCreationOnly	

TABLE 10 – INPUTS & OUTPUTS, CHECKOUT

Payment Method Card: Card

INPUT		OUTPUT	
cardInfo		actionProcessed	
	PAN		id
	secureCode		type
	validationDate		executed
	cardholderName	tokenResponse	
	createToken		tokenType
			value
			tokenName
			maskedPAN
			expireDate
instalmentPlan			
	id		

TABLE 11 – INPUTS & OUTPUTS, CARD

Payment Method Multibanco: MB Service Reference

INPUT	OUTPUT	
	returnStatus	
		• statusCode
		• statusMsg
		• statusDescription
	transactionID	
	merchant	

		• terminalId
		• channel
		• merchantTransactionId
	paymentType	
	paymentReference	
		• reference
		• entity
		• paymentEntity
		• amount
		• value
		• currency
		• expiryDate
		• status
	actionResponse	
		• id
		• type
		• data
	• paymentStatus	

TABLE 12 – INPUTS & OUTPUTS, MULTIBANCO

Payment Method MB WAY ID: MB WAY - ID

INPUT	OUTPUT	
customerPhone	returnStatus	
		statusCode
		statusMsg
		statusDescription
	transactionID	
	merchant	
		terminalId
		channel
		merchantTransactionId
	paymentType	
	paymentStatus	
	actionResponse	
		id
		type
		data

TABLE 13 – INPUTS & OUTPUTS, MB WAY ID

Payment Method Authorised Payment: MB WAY – Mandate

The Payment Method Authorised Payment shares with Payment Method MB WAY ID the same interfaces. The Payment Method Authorised Payment is applied when mandate field is fulfilled.

INPUT	OUTPUT	
mandate	returnStatus	
mandateType mandateCreationOnly		statusCode
		statusMsg
		statusDescription
	transactionID	
	merchant	
		terminalId
		channel
		merchantTransactionId
	paymentType	
	paymentStatus	
	actionResponse	
		id
		type
		data

TABLE 14 – INPUTS & OUTPUTS, AUTHORISED PAYMENT

***Device Input**

INPUT	
deviceInfo	
	• browserAcceptHeader
	• browserJavaEnabled
	• browserJavascriptEnabled
	• browserLanguage
	• browserColorDepth
	• browserScreenHeight
	• browserScreenWidth
	• browserTZ
	• browserUserAgent
	• systemFamily
	• systemVersion
	• systemArchitecture
	• deviceManufacturer
	• deviceModel
	• deviceID
	• applicationName
	• applicationVersion

	• geoLocalization • ipAddress

TABLE 15 – DEVICE INPUT

Technical Architecture - FORM

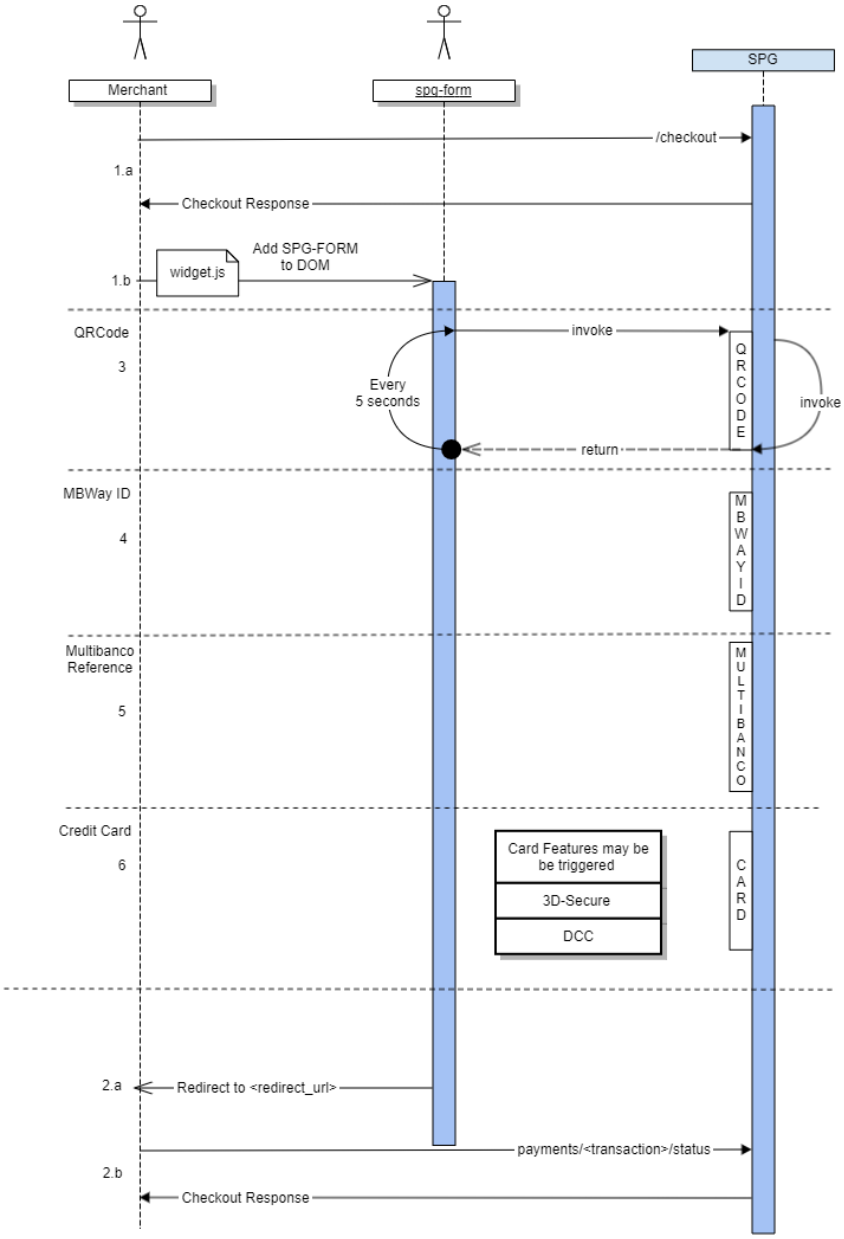

FIGURE 26 – TECHNICAL ARCHITECTURE – FORM

Transaction Status

Is possible to receive the below transaction status:

- **IN PROCESSING** – meaning that the *transaction initiated* is now in *processing*.
- **PARTIAL** – is *pending* to receive information (E.g. 3DS Challenge).
- **SUCCESS** – Informing *transaction OK,* was executed with success.
- **DECLINED** – indicate that the transaction was declined by an operation intervenient.
- **TIMEOUT** – informing a potential *communication error*
- **ERROR** – indicate that the transaction was *declined by external system*

After the transaction is in processing it will only retrieve the "partial" status if some action is pending to be taken. After that it will send the transaction status as below.

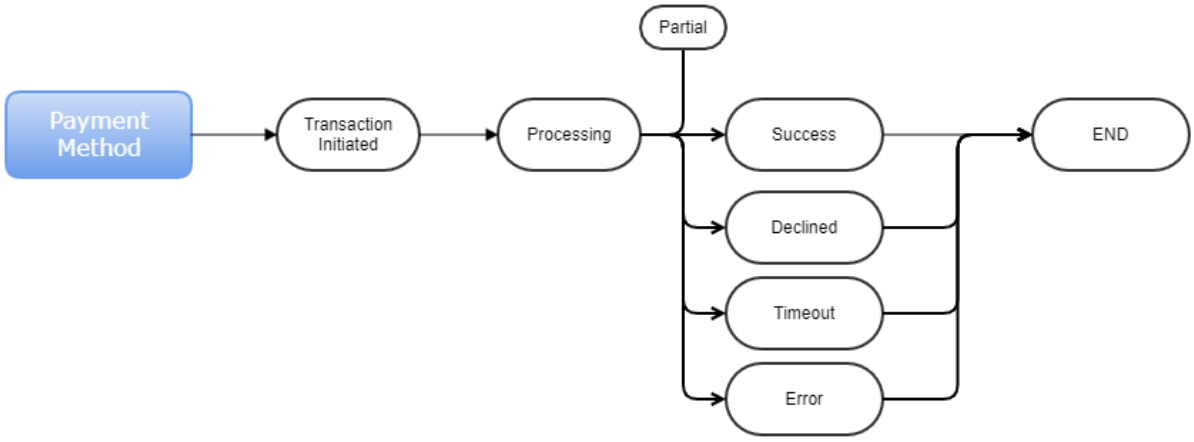

FIGURE 27 – TRANSACTION STATUS

Card Features

The card features that will be active will depend on the agreement that the merchant has with his acquirer. Only the acquirer can activate or deactivate the below options.

3D-Secure

The 3-D Secure authentication protocol is based on a three-domain model where the Acquirer Domain and Issuer Domain are connected by the Interoperability Domain for the purpose of authenticating a Cardholder during an electronic commerce (e-commerce) transaction or to provide identity verification and account confirmation.

This protocol applies to those who shop online at merchants who have online stores and adhere to 3D-Secure. For more details: https://www.emvco.com/emv-technologies/3d-secure/

Dynamic Currency Conversion

DCC is a credit card feature that allows you to make a point-of-sale credit card purchase in a foreign country using the currency of your home country, also known as cardholder preferred currency. This feature makes it easier to understand the price you are paying and avoids doing the currency conversion math.

Tokenization

The token is a reference from the card converted to a different code (i.e. identifier/token) allowing the Merchant to save the card for future payments without having the card information for PCI-DSS purposes. Tokenization is the process of substituting a sensitive data element with a non-sensitive equivalent to save this information in the client wallet and improve the client experience on the next visit. The token is a reference (i.e. identifier) that maps back to the sensitive data through a tokenisation system.

The mapping from original data to a token uses a method that render tokens infeasible to reverse in the absence of the tokenisation system, for example using tokens created from random numbers.

Merchant Notification System (Webhooks)

On every transaction made by the user, SIBS Gateway V2 sends notification via e-mail or an endpoint configuration, to the Merchant. This notification (Webhook) will inform the Merchant about the details and the status of a transaction.

The type of webhook to be sent by SIBS Gateway V2 must be configured in SIBS Gateway V2 Backoffice platform. In case the Merchant chooses the Webhook URL type, the Merchant must insert the URL where it wants to receive the webhook. In case the Merchant chooses the Webhook E-mail type, the Merchant must insert the default e-mail that will receive the notifications sent by SIBS Gateway V2. The webhook configuration can be configured by payment method and by Merchant, Store or Terminal. To configure Webhook URL type, the Merchant must be aware of some technical configurations, such as:

- The webhook's endpoint must be available in TLS 1.2 and in ports 80 or 443;
- The Merchant must save from their side and insert an encryption key in SIBS Gateway V2 Backoffice platform or use the one suggested by the platform in order to be able to decipher the webhook received;
- The webhook is encrypted with AES-256;
- The Merchant must insert a specific e-mail address where it will receive the notifications concerning webhook problems.

For the Webhook E-mail type, only one notification will be sent per transaction. If the Merchant chooses the Webhook URL type and does not acknowledge the reception of the notification, SIBS Gateway V2 will trigger the Webhook Retry System (to see how it works, please see section: Merchant Notification (Webhook) Retry System).

In the webhook structure, there's one field that plays a crucial part in the webhook sending process - the " *notificationID* ". This field identifies the webhook that was sent by SIBS Gateway V2 and will be used by the Merchant in order to send the acknowledge of the reception of the webhook to SIBS Gateway V2 in addition to the "*statusCode*" with value "200" and the "*statusMsg*" field with value "Success". This will prevent the Webhook Retry System from triggering and, in case of Webhook URL Type, from sending an e-mail with the failed notifications.

In case there's communication or system issues, the Webhook System cannot guarantee the message order, especially, if the time difference between the notifications is smaller than the time it takes to process them. Once the issues are resolved, new notifications will arrive in real time and old

notifications will be resent. If, for some reason, the webhooks are not retried or the Merchant is not able to open the webhook, SIBS Gateway V2 provides the "Checkout Status" service where the Merchant can access information about the status of a transaction.

In the sub-section below, is presented the structure of the request (sent by SIBS Gateway V2) and response message (acknowledged by Merchant) of the Webhook Notifications where some of the complex types will only be shown in case a certain payment method is used.

Merchant Notification (Webhook) Structure

MerchantNotificationRequest:

Body	returnStatus	statusDescription	string (Maximum 256 Text)
		statusMsg	string (Maximum 256 Text) (ex: Success) Possible values are {"Success", "Partial", "Declined", "InProcessing", "Pending", "Timeout", "Error"}.
		statusCode	string (Maximum 11 Text) (ex: 000)
	paymentStatus	string (“Success”, “Partial”, “Declined”, “InProcessing”, “Pending”, “Timeout”, “Error”)	
	paymentMethod	string (“MBWAY”, “CARD”, “REFERENCE”, “TOKEN”, “STATIC_QRCODE”, “MANDATE”, “XPAY”)	
	transactionID	string (Maximum 35 Text)	
	transactionDateTime	string (date-time)	
	amount	currency	string (ISO 4217 Alpha-3 Code)
		value	number (double)
	merchant	transactionId	string (Maximum 35 Text)
		terminalId	integer (int32)
		inApp	boolean
	paymentType	string (“AUTH”, “PURS”, “CAPT”, “CAUT”, “RFND”, “CANC”, “PREF”, “CPRF”. “CMBW”, “MAND”, “MAUT”. “MPUR”, “MITR”)	
	paymentReference	reference	string (Maximum 50 Text)
		entity	string (Maximum 50 Text)
		paymentEntity	string (Maximum 50 Text)
		amount	currency: string (ISO 4217 Alpha-3 Code) value: number (double)
		status	string (“UNPAID”, “PAID”, PARTPAIDLS”,” PARTPAIDOPN”, “CANC”, ”UNKN”)
		expireDate	string (date-time)
	token	tokenName	string (Maximum 50 Text)

		tokenType	string (“TokenType1”: “Email”, “MobilePhone”, “Card”)		
		value	string (Maximum 50 Text)		
		maskedPAN	string (Maximum 35 Text)		
		expireDate	string (date-time)		
	financialOperation	product	staticQRCodeId	Max36Text	
			twoStepPurchase	boolean	
			aliasName	string (Maximum 256 Text)	
			merchantContactType	string (Maximum 256 Text)	
			productName	Max100Text	
			productQuantity	Max4NumericText	
			productAmount	Decimal8_2	
			expeditionAmount	Decimal8_2	
			contactClientIndicator	Max1NumericText	
			customerSupportContact	Max100Text	
		billingInfo	billingName	Max45Text	
			billingSurname	Max45Text	
			billingTIN	Max14Text	
			billingAddressCity	string (Maximum 50 Text)	
			billingAddressCountry	Max3Text	
			billingAddressLine1	string (Maximum 50 Text)	
			billingAddressLine2	string (Maximum 50 Text)	
			billingAddressLine3	string (Maximum 50 Text)	
			billingAddressPostalCode	Max16Text	
			billingHomePhone	Max15Text	
			billingMobilePhone	Max15Text	
			sendInvoiceByEmail	boolean	
		expeditionInfo	expeditionName	Max45Text	
			expeditionSurname	Max45Text	
			expeditionAddressCity	string (Maximum 50 Text)	
			expeditionAddressCountry	Max3Text	
			expeditionAddressLine1	string (Maximum 50 Text)	
			expeditionAddressLine2	string (Maximum 50 Text)	

			expeditionAddressLine3	string (Maximum 50 Text)
			expeditionAddressPostalCode	Max16Text
			expeditionHomePhone	Max15Text
			expeditionMobilePhone	Max15Text
			expeditionEmail	Max100Text
		serviceOperationPlayer	Max36Text	
		merchantOperation	Max36Text	
		serviceType	Max10Text	
	mbwayMandate (optional)	mandateIdentification (optional)	string (Maximum 64 Text)	
		mandateAction (optional)	Max4Text ("CRTN"* - Mandate Creation, "SSPN"- Mandate Suspension by Client; "RCTV" - Mandate Reactivation by Client, "LMUP"* - Mandate Limits Update by Client, "CNCL" - Mandate Cancellation by Client)	
		mandateActionStatus (optional)	Max4Text ("SCCS" - Success (applicable to all Action Codes), "RFSD" - Refused due to validation rules (only applicable to Mandate Creation), "RJCT" - Rejected by Client (only applicable to Mandate Creation))	
		mandateActionStatusReason (optional)	Max5Text	
		merchantTransactionIdentification (optional)	string (Maximum 35 Text)	
		mandateType (optional)	Max4Text ("ONCL" - One-Click Mandate, "SBSC" - Subscription Mandate)	
		mandateAmountLimit (optional)	currency	string (ISO 4217 Alpha-3 Code)
			value	number (double)
		mandateExpirationDate (optional)	ISODate	
		clientName (optional)	string (Maximum 256 Text)	
	InstalmentPlan	PlanId (optional)	Max3NumberText	
	customer	customerName	string (Maximum 45 Text)	

		(optional)	
		customerEmail (optional)	string (Maximum 256 Text)
		customerTIN (optional)	Max14Text
	merchantInitiatedTransaction	type (optional)	Max4Text ("UCOF"-Unscheduled Credential on File, "RCRR"-Recurring)
		validityDate (optional)	string (date-time)
		amountQualifier (optional)	Max7Text ("ACTUAL"-Actual amount, "ESTIMATED"-Estimated amount (the final amount could be above or below), "DEFAULT"-Default amount) If not present, the "DEFAULT" value is assumed. If merchantInitiatedTransaction has MITType "UCOF", the amountQualifier must be set to "ESTIMATED".
		description (optional)	string (Maximum 50 Text)
		Schedule (optional)	initialDate (optional): string (date-time) finalDate (optional): string (date-time) interval (optional): Max10Text ("DAILY", "WEEKLY", "BIWEEKLY", "MONTHLY", "QUARTERLY", "SEMIANNUAL", "ANNUAL")
	threeDSecure (optional)	whiteListStatus (optional)	string (Exact 1 Text) Possible values are: • Y = 3DS Requestor is whitelisted by cardholder • N = 3DS Requestor is not whitelisted by cardholder • E = Not eligible as determined by issuer • P = Pending confirmation by cardholder • R = Cardholder rejected • U = Whitelist status unknown, unavailable, or does not apply.
	terminalBrand	string (Exact2Text)	

	(optional)	
	wrapperType (optional)	string (Exact4Text)
	internalTransactionId (optional)	string (Max50Text)
	notificationID	PaymentNotificationId (UUID – Exact36Text)

"LMUP"* - This mandateAction status is triggered when the client modifies the MBWAY Authorised Payment limits or the expiration date in the MBWAY app. After the update of information of the Authorised Payment and before sending the webhook, SIBS Gateway V2 will send a notification to the Merchant informing the details of the Authorised Payment (Authorised Payment Limits, Expiration Date, etc..).

"CRTN"* - After the creation of an Authorised Payment and before sending the webhook, SIBS Gateway V2 will send a notification to the Merchant informing the details of the Authorised Payment created (Authorised Payment Limits, Expiration Date, etc..).

MerchantNotificationResponse:

Body	statusCode	string (Maximum 11 Text) (ex: 000)	
	statusMsg	string (Maximum 256 Text) (ex: Success) Possible values are {"Success", "Partial", "Declined", "InProcessing", "Pending", "Timeout", "Error"}.	
	notificationID	PaymentNotificationId (UUID – Exact36Text)	

Merchant Notification (Webhook) Examples

In this sub-section, it is possible to find some of webhook notifications examples sent by SIBS Gateway V2 and the answer that Merchant should (Webhook Notification Response). The answer to this message is crucial to avoid more unnecessary notifications sent by SIBS Gateway V2 through the Webhook Retry System (see section Merchant Notification (Webhook) Retry System).

Webhook Notification Response

```
{
        "statusCode": "200",
        "statusMsg": "Success",
        "notificationID": "93b9b3a6-602f-4769-8158-48ae9c380ed5"
}
```

Webhook Notification - Static QR Code Purchase


```
{
        "returnStatus": {
                "statusMsg": "Success",
                "statusCode": "000"
        },
        "paymentStatus": "Success",
        "paymentMethod": "STATIC_QRCODE",
        "transactionID": "3W005000159764",
        "transactionDateTime": "2022-11-12T16:17:53.127Z",
        "amount": {
                "currency": "EUR",
                "value": 1.6
        },
        "merchant": {
                "terminalId": 46431
        },
        "paymentType": "PURS",
        "financialOperation": {
                "product": {
                        "staticQRCodeId": "7ecf9d4e23334f7da6cb",
                        "twoStepPurchase": false,
                        "aliasName": "351#934885128",
                        "merchantContactType": "NA",
                        "productName": "Nome info",
                        "productQuantity": 1,
                        "productAmount": 1.5,
                        "expeditionAmount": 0.1,
                        "contactClientIndicator": 1,
                        "customerSupportContact": "teste@info.com"
                },
                "billingInfo": {
                        "billingTIN": "123456789",
                        "billingAddressCity": "Lisboa ",
                        "billingAddressLine1": "Rua D João ",
                        "billingAddressLine2": "8 2drt",
                        "billingAddressPostalCode": "4646-846",
                        "billingMobilePhone": "+ 351 987654321"
                },
```

```
                "expeditionInfo": {
                        "expeditionName": "Olga Rodrigues",
                        "expeditionAddressCity": "Lisboa ",
                        "expeditionAddressLine1": "Rua D João ",
                        "expeditionAddressLine2": "8 2drt",
                        "expeditionAddressPostalCode": "4646-846",
                        "expeditionMobilePhone": "+ 351 987654321"
                }
        },
        "notificationID": "5544b042-8b9a-451e-8421-53fcfe9538f8"
}
```


Webhook Notification – MB WAY Authorised Payment Creation


```
{
  "returnStatus": {
    "statusMsg": "Success",
    "statusCode": "000"
  },
  "paymentStatus": "Success",
  "paymentMethod": "MBWAY",
  "transactionID": "UpxBSsjySqaR80aEzWxa",
  "transactionDateTime": "2022-11-11T16:18:53.127Z",
  "amount": {
    "currency": "EUR",
    "value": 1.96
  },
  "merchant": {
    "transactionId": "1009854",
    "terminalId": "45546",
    "merchantName": "Test Authorised Payment Creation",
    "inApp": "false"
  },
  "paymentType": "AUTH",
```

```
  "mbwayMandate": {
    "mandateIdentification": "12345690656800744652",
    "mandateAction": "CRTN",
    "mandateActionStatus": "SCCS",
    "mandateType": "SBSC",
    "clientName": "Andresa COS",
    "aliasMBWAY": "351#914341580",
    "mandateExpirationDate": "2027-12-31",
    "mandateAmountLimit": {
      "value": "100.00"
    }
  },
  "notificationID": "6e96b74f-ec33-4ca9-8cd3-1d00c5d82a0d"
}
```


Webhook Notification – MB WAY Authorised Payment Purchase after Creation


```
{
  "returnStatus": {
    "statusMsg": "Success",
    "statusCode": "000"
  },
  "paymentStatus": "Success",
  "paymentMethod": "MBWAY",
  "transactionID": "s2jrfvnriuv8tgt554tg",
  "transactionDateTime": "2022-11-11T16:18:53.127Z ",
  "amount": {
    "currency": "EUR",
    "value": 15.2
  },
  "merchant": {
```

```
    "transactionId": "863b730df285443ca404e0085dref45",
    "terminalId": 99978,
    "merchantName": "Teste Pagamentos Autorizados Sucesso"
  },
  "paymentType": "PURS",
  "token": {
    "tokenType": "MobilePhone",
    "value": "351#919992314"
  },
  "internalTransactionId": "S13074000497973S",
  "notificationID": "839ca363-8581-4b9f-8041-a45tgbh67u"
}
```


Webhook Notification - Cancel MB WAY Authorised Payment


```
{
        "returnStatus": {
                "statusMsg": "Success",
                "statusCode": "000"
        },
        "paymentStatus": "Success",
        "paymentMethod": "MANDATE",
        "transactionID": "s2t44gh56y67jAsftgt",
        "transactionDateTime": "2022-11-11T16:18:53.127Z ",
        "amount": {
                "currency": "EUR",
                "value": 15.2
        },
        "merchant": {
                "transactionId": "863b730df285443ca404e0085fd6789",
                "terminalId": 96546,
```

```
                "merchantName": "Canc Pagamento Autorizado"
        },
        "paymentType": "CAUT",
        "internalTransactionId": "S14072900000002S",
        "notificationID": "0d93a5ef-13e0-4a6b-9e40-652342321451"
}
```


Webhook Notification – Cancel MB WAY Authorised Payment - Decline


```
{
  "returnStatus": {
    "statusMsg": "Declined",
    "statusCode": "10.106.2662"
  },
  "paymentStatus": "Declined",
  "paymentMethod": "MANDATE",
  "transactionID": "s2SFFe3445gjn5hjas",
  "transactionDateTime": "2022-11-11T16:18:53.127Z",
  "amount": {
    "currency": "EUR",
    "value": 15.2
  },
  "merchant": {
    "transactionId": "863b730df285443ca404e0085fd6ssd",
    "terminalId": 96546,
    "merchantName": "Canc Pagamento Autorizado"
  },
  "paymentType": "CAUT",
  "internalTransactionId": "S14072900000007S",
  "notificationID": "0d93a5ef-13e0-4a6b-9e40-h589085hnA "
 }
```

Webhook Notification - MB WAY Purchase with Alias


```
{
   "returnStatus": {
      "statusMsg": "Success",
      "statusCode": "000"
   },
   "paymentStatus": "Success",
   "paymentMethod": "MBWAY",
   "transactionID": "s2efgrtgn456477",
   "transactionDateTime": "2022-11-11T16:18:53.127Z ",
   "amount": {
      "currency": "EUR",
      "value": "16.20"
   },
   "merchant": {
      "transactionId": "863b730df285443ca404e0085fw234",
      "terminalId": "99978",
      "merchantName": "Teste MBWAY Sucesso",
      "inApp": "false"
   },
   "paymentType": "PURS",
   "token": {
      "tokenType": "MobilePhone",
      "value": "351#912345678"
   },
   "internalTransactionId": "S14073500001359S",
   "notificationID": "839ca363-8581-4b9f-8041-a6dfgrtyu643"
}
```

Webhook Notification - MB WAY Purchase with Alias - Declined


```
{
   "returnStatus": {
      "statusMsg": "Declined",
      "statusCode": "01.106.0004"
   },
   "paymentStatus": "Declined",
   "paymentMethod": "MBWAY",
   "transactionID": "s2nvtgbntiuybn88485t",
   "transactionDateTime": "2022-11-11T16:18:53.127Z ",
   "amount": {
      "currency": "EUR",
      "value": "16.20"
   },
   "merchant": {
      "transactionId": "863b730df285443ca404e00853de45",
      "terminalId": "99978",
      "merchantName": "Teste MBWAY Erro",
      "inApp": "false"
   },
   "paymentType": "PURS",
   "token": {
      "tokenType": "MobilePhone",
      "value": "351#912345500"
   },
   "internalTransactionId": "S14073500001351S",
   "notificationID": "839ca363-8581-4b9f-8041-a6d456jkh678"
}
```

Webhook Notification - Card Purchase


```
{
   "returnStatus": {
      "statusMsg": "Success",
      "statusCode": "000"
   },
   "paymentStatus": "Success",
   "paymentMethod": "CARD",
   "transactionID": "s2shdvbevr76756er",
   "transactionDateTime": "2022-11-11T16:18:53.127Z",
   "amount": {
      "currency": "EUR",
      "value": 19.2
   },
   "merchant": {
      "transactionId": "863b730df285443ca404e008456sw2",
      "terminalId": 66645,
      "merchantName": "Teste Cartões, Lda"
   },
   "paymentType": "PURS",
   "internalTransactionId": "S140730000000711S",
   "notificationID": "8ec13f91-0129-44ff-980c-79e456fds21s"
}
```


Webhook Notification - Token Purchase


```
{
   "returnStatus": {
      "statusMsg": "Success",
      "statusCode": "000"
```

```
  },
  "paymentStatus": "Success",
  "paymentMethod": "TOKEN",
  "transactionID": "s2grnvtruigbitu845hhtn",
  "transactionDateTime": "2022-11-11T16:18:53.127Z ",
  "amount": {
    "currency": "EUR",
    "value": 19.2
  },
  "merchant": {
    "transactionId": "863b730df285443ca404e008mvn432kop",
    "terminalId": 77745,
    "merchantName": "PETROLEOS DE PORTUGAL - PETROGAL, SA"
  },
  "paymentType": "PURS",
  "token": {
    "tokenType": "Card",
    "value": "gbtyhyujyuikuier45646ger4"
  },
  "notificationID": "462ec85b-5e1a-4a00-a4d7-97d345fds234"
}
```


Webhook Notification - Token Generation


```
{
  "returnStatus": {
    "statusMsg": "Success",
    "statusCode": "000"
  },
  "paymentStatus": "Success",
  "paymentMethod": "CARD",
```

```
  "transactionID": "sandboxfghwTKNGEN000",
  "transactionDateTime": "2022-11-11T16:18:53.127Z ",
  "amount": {
    "currency": "EUR",
    "value": 19.2
  },
  "merchant": {
    "transactionId": "863b730df285443ca404e008dsw212",
    "terminalId": 77745,
    "merchantName": "GOMERCIANTE DO LEGACY LD"
  },
  "paymentType": "AUTH",
  "token": {
    "tokenType": "Card",
    "value": "fjklvnrtinrt435356jv"
  },
  "internalTransactionId": "S13074000501340S",
  "notificationID": "462ec85b-5e1a-4a00-a4d7-97ddfe321567"
}
```


Webhook Notification - Cardholder Initiated Transaction (CIT) with Type Recurring


```
{
        "returnStatus": {
                "statusMsg": "Success",
                "statusCode": "000"
        },
        "paymentStatus": "Success",
        "paymentMethod": "CARD",
        "transactionID": "s24nZiFtu15SCiW9vpDm",
```

```
        "transactionDateTime": "2022-12-27T16:05:27.585Z",
        "amount": {
                "currency": "EUR",
                "value": 11.2
        },
        "merchant": {
                "transactionId": "6637262",
                "terminalId": 45546,
                "merchantName": "GOMERCIANTE DO LEGACY"
        },
        "paymentType": "PURS",
        "notificationID": "a433453e-6e40-4f53-a80d-4985cffa36f6",
        "merchantInitiatedTransaction": {
                "type": "RCRR",
                "validityDate": "2023-06-30T01:00:00+01:00",
                "amountQualifier": "ACTUAL",
                "schedule": {
                        "initialDate": "2022-12-27T00:00:00Z",
                        "finalDate": "2023-01-06T00:00:00Z",
                        "interval": "DAILY"
                }
        }
}
```


Webhook Notification - Merchant Initiated Transaction (MIT) with Type Recurring


```
{
        "returnStatus": {
                "statusMsg": "Success",
                "statusCode": "000"
```

```
        },
        "paymentStatus": "Success",
        "paymentMethod": "CARD",
        "transactionID": "s2K9YuqPQuC9NkdCvDA4",
        "transactionDateTime": "2023-01-06T00:00:03.378Z",
        "amount": {
                "currency": "EUR",
                "value": 11.2
        },
        "merchant": {
                "terminalId": 45546,
                "merchantName": "GOMERCIANTE DO LEGACY"
        },
        "paymentType": "MITR",
        "notificationID": "3a9d8777-292d-48fb-971c-6e6558de0fd5",
        "merchantInitiatedTransaction": {
                "type": "RCRR",
                "amountQualifier": "ACTUAL"
        }
}
```


Webhook Notification - Cardholder Initiated Transaction (CIT) with Type UCOF


```
{
        "returnStatus": {
                "statusMsg": "Success",
                "statusCode": "000"
        },
        "paymentStatus": "Success",
        "paymentMethod": "CARD",
        "transactionID": "s2bDX8mjiFZU5ZwfvSvR",
        "transactionDateTime": "2023-01-06T16:57:11.2Z",
```

```
        "amount": {
                "currency": "EUR",
                "value": 102.38
        },
        "merchant": {
                "transactionId": "3064371",
                "terminalId": 45546,
                "merchantName": "GOMERCIANTE DO LEGACY"
        },
        "paymentType": "AUTH",
        "notificationID": "45d108d3-2cef-41a6-90af-a3bdb4af244a",
        "merchantInitiatedTransaction": {
                "type": "UCOF",
                "validityDate": "2023-01-31T00:00:00Z",
                "amountQualifier": "ESTIMATED"
        }
}
```


Webhook Notification - Merchant Initiated Transaction (MIT) with Type UCOF


```
{
        "returnStatus": {
                "statusMsg": "Success",
                "statusCode": "000"
        },
        "paymentStatus": "Success",
        "paymentMethod": "CARD",
        "transactionID": "s2nYaNUC3hFpqd0HGq4Z",
        "transactionDateTime": "2022-12-23T10:48:39.153Z",
        "amount": {
                "currency": "EUR",
```

```
            "value": 5.16
        },
        "merchant": {
            "transactionId": "6540474",
            "terminalId": 45546,
            "merchantName": "GOMERCIANTE DO LEGACY"
        },
        "paymentType": "MITR",
        "notificationID": "c9786e7e-dc2a-4611-8a31-4b8197f41138",
        "merchantInitiatedTransaction": {
            "type": "UCOF",
            "amountQualifier": "ESTIMATED"
        }
}
```


Webhook Notification - MB Reference Generation


```
{
  "returnStatus": {
    "statusMsg": "Pending",
    "statusCode": "00.110.1601"
  },
  "paymentStatus": "Pending",
  "paymentMethod": "REFERENCE",
  "transactionID": "s24587y857mtjgnbt",
  "transactionDateTime": "2022-12-23T10:48:39.153Z ",
  "amount": {
    "currency": "EUR",
    "value": "20.0"
  },
  "merchant": {
```

```
    "transactionId": "863b730df285443ca404e008sde23",
    "terminalId": "88845",
    "merchantName": "Teste Referências, Lda"
  },
  "paymentReference": {
    "reference": "256309828",
    "entity": "40200",
    "paymentEntity": "40200",
    "amount": {
      "value": "20.0",
      "currency": "EUR"
    },
    "status": "UNPAID",
    "expiryDate": "2022-01-23T10:48:39.153Z "
  },
  "paymentType": "PREF",
  "notificationID": "20273954-0540-4bd3-8e01-234eds234cds"
}
```


Webhook Notification - MB Reference “PAID”


```
{
  "returnStatus": {
    "statusMsg": "Success",
    "statusCode": "000"
  },
  "paymentStatus": "Success",
  "paymentMethod": "REFERENCE",
  "transactionID": "s2jfvbiurbg8956vng",
  "transactionDateTime": "2022-12-23T10:48:39.153Z",
  "amount": {
```

```
    "currency": "EUR",
    "value": 20
  },
  "merchant": {
    "transactionId": "863b730df285443ca404e00823sh76",
    "terminalId": 88845,
    "merchantName": "Teste Referências, Lda"
  },
  "paymentType": "PREF",
  "internalTransactionId": "S140730000000341S",
  "notificationID": "20273954-0540-4bd3-8e0-asd2w4jo0mmt"
}
```


Webhook Notification - Cashout


```
{
  "returnStatus": {
    "statusMsg": "Success",
    "statusCode": "000"
  },
  "paymentStatus": "Success",
  "paymentMethod": "MBWAY",
  "transactionID": "s2QYBqEwBAkvC1PuRDu9",
  "transactionDateTime": "2025-03-14T15:54:46.999Z",
  "amount": {
    "currency": "EUR",
    "value": 4
  },
  "merchant": {
    "transactionId": "12345678987654321",
    "terminalId": 45546,
```

```
    "merchantName": "GOMERCIANTE DO LEGACY LD",
    "merchantBrandName": "LEGACY LD",
    "inApp": "false"
  },
  "paymentType": "CSHT",
  "token": {
    "tokenType": "MobilePhone",
    "value": "351#914109379"
  },
  "operationDescription": " Cashout to client 1234",
  "clientIBAN": "PT50 0007 **** **** **** *01 63",
  "notificationID": "cefb68db-4a50-43cf-bd6d-e8395722599c"
}
```


Webhook Notification – Cashout – Declined


```
{
  "returnStatus": {
    "statusDescription": "Declined Operation",
    "statusMsg": "Declined",
    "statusCode": "10.110.3005"
  },
  "paymentStatus": "Declined",
  "paymentMethod": "MBWAY",
  "transactionID": "s2QYBqEwBAkvC1PuRDu9",
  "transactionDateTime": "2025-03-14T15:54:46.999Z",
  "amount": {
    "currency": "EUR",
    "value": 4
  },
  "merchant": {
```

```
    "transactionId": "12345678987654321",
    "terminalId": 45546,
    "merchantName": "GOMERCIANTE DO LEGACY LD",
    "merchantBrandName": "LEGACY LD",
    "inApp": "false"
  },
  "paymentType": "CSHT",
  "token": {
    "tokenType": "MobilePhone",
    "value": "351#914109379"
  },
  "operationDescription": "Cashout to client 1234",
  "notificationID": "cefb68db-4a50-43cf-bd6d-e8395722599c"
}
```


Merchant Notification (Webhook) Retry System

In case there's an error in the Merchant Notification system, SIBS Gateway V2 is ready to act and send notification retries to the Merchant until the Merchant acknowledges the reception of the notification. When SIBS Gateway V2 sends the Merchant Notification to the Merchant, the internal status of the notification stays as "Requested" until the merchant sends an acknowledge to SIBS Gateway V2. If the Merchant sends the acknowledge, the notification status is transitioned to "Processed" and the process ends, otherwise the Webhook Retry System is triggered at the end of a parameterized time.

The Webhook Retry System is a batch system that categorizes the merchant notifications that are on status "Requested" according to a minimum and maximum number of retries. This batch categorizes the merchant notifications in seven tiers, Tier 0 (zero) to Tier 6 (six). Each tier, as a specific retries' count, a specific number of webhooks to be retried and a retry period. Tier 0 represents the highest rate of retries on the Webhook Retry System and the Tier 6 represents the lowest rate of retries in terms of periodicity.

In the table below are represented the seven categories of the Webhook Retry System, as well as, the difference between them in terms of: Retries' Count, Number of webhooks retried and the Retry period time. This batch process for a specific retry will end after two months since the first retry has been started.

Tiers	Retries' Count	Number of Webhooks Retried	Retry Period Time
Tier 0	0 to 4	400 (max)	Every 5 minutes
Tier 1	5 to 8	600 (max)	Every 1 hour
Tier 2	9 to 13	800 (max)	Every 2 hours
Tier 3	14 to 19	1000 (max)	Every 4 hours
Tier 4	20 to 26	1200 (max)	Every 6 hours
Tier 5	27 to 34	1400 (max)	Every 12 hours
Tier 6	35 to 43	1600 (max)	Every 24 hours

TABLE 16 - WEBHOOK RETRY SYSTEM SPECIFICATION

Security Module

PCI DSS

Payment Card Industry Data Security Standard is an information security standard for organizations that handle branded credit cards from the major card schemes, the standard was created to increase controls around cardholder data and reduce card fraud.

Merchants that want to process, store or transmit card data will need to PCI compliant, with SIBS the Merchants have the choice to use the FORM that is already fully PCI compliant not needing any certification from the store. The “Server-to-Server” API integration variant requires the Merchant to collect the card data which increases the PCI-compliance needs.

The Council does not enforce compliance this is request by individual payment brands or acquiring banks.

PCI 3-Step Process

- **Assess** – Identifying cardholder data, taking an inventory of IT assets and business processes for payment card processing, and analysing them for vulnerabilities.
- **Remediate** – Fixing vulnerabilities and eliminating the storage of cardholder data unless necessary.
- **Report** – Compiling and submitting required reports to the appropriate acquiring bank and card brands.

For more details: https://pt.pcisecuritystandards.org/index.php

SIBS Gateway V2 Error Codes

Checkout Request

HTTP **Code**	**HTTP Message**	Status Code	Status Message	Status Description
200	OK	000	Success	Success
400	Bad Request	E0001	Error	Invalid authentication or authorisation data
400	Bad Request	E0002	Error	Invalid merchant or terminal code
400	Unauthorized	E0003	Error	Invalid request, data is missing or is invalid
400	Bad Request	E0004	Error	Operation not allowed by the Merchant
400	Bad Request	E0005	Error	Operation not allowed by the Acquirer
400	Bad Request	E0006	Error	Operation not allowed, terms and condition not configured by Merchant
400	Bad Request	E0007	Error	No payment methods available
400	Bad Request	E9999	Error	Operation declined by authorisation system
503	Internal Server Error	T9999	Temporary error	SIBS temporary internal error

TABLE 17 - ERROR CODES - CHECKOUT REQUEST

Payment Request

HTTP **Code**	**HTTP Message**	Status Code	Status Message	Status Description
206	Partial Content	000	Success	Partial
200	OK	000	Success	Success
400	Unauthorized	E0100	Error	Invalid merchant or terminal code
400	Bad Request	E0101	Error	Invalid request, data is missing or is invalid
400	Bad Request	E0102	Error	Checkout expired
409	Conflict	E0103	Error	Wrong parameterization for the payment method
400	Bad Request	E0104	Error	Acceptor not active

HTTP **Code**	**HTTP Message**	Status Code	Status Message	Status Description
400	Bad Request	E0105	Error	Invalid terminal
400	Bad Request	E0106	Error	Terminal code differs from checkout terminal code
400	Bad Request	E0107	Error	Invalid terminal status
400	Bad Request	E0108	Error	Terminal is not authorised
400	Bad Request	E0109	Error	Invalid Amount
400	Bad Request	E0110	Error	Amount not supported
400	Bad Request	E0111	Error	Amount limit exceeded
400	Bad Request	E0112	Error	Wrong Currency
400	Bad Request	E0113	Error	Insufficient funds
400	Bad Request	E0114	Error	Transaction amount exceeds the authorized amount
400	Bad Request	E0115	Error	Transaction amount is lower than the commission
400	Bad Request	E0116	Error	Invalid transaction value
400	Bad Request	E0117	Error	Daily amount limit exceeded
400	Bad Request	E0118	Error	Operation declined due to fraud suspicion
400	Bad Request	E0119	Error	Declined operation
400	Bad Request	E0120	Error	Card issuer temporarily unavailable
400	Bad Request	E0121	Error	Invalid payment request, terms and conditions were not accepted
400	Bad Request	E0122	Error	Transaction not allowed
400	Bad Request	E0123	Error	Transaction not authorized
403	Bad Request	E0124	Error	Invalid operation, transaction already processed.
400	Bad Request	E0125	Error	Invalid original transaction
400	Bad Request	E0126	Error	No match between transactions
400	Bad Request	E0127	Error	Transaction already made
400	Bad Request	E0128	Error	Exceeded the number of operations allowed
400	Bad Request	E0129	Error	Declined, recurring payment is deactivated
400	Bad Request	E0130	Error	Authorisation is expired
400	Bad Request	E0131	Error	Authorisation is cancelled.
400	Bad Request	E0132	Error	Authorisation was already used
400	Bad Request	E0133	Error	Refund amount exceeds purchase amount
400	Bad Request	E0134	Error	Refund amount exceeds available daily funds

HTTP **Code**	**HTTP Message**	Status Code	Status Message	Status Description
400	Bad Request	E0135	Error	Purchase already refunded.
400	Bad Request	E0136	Error	Invalid refund amount
400	Bad Request	E0137	Error	Cancellation amount exceeds the authorized amount
400	Bad Request	E0138	Error	Invalid Cancellation
400	Bad Request	E0139	Error	No agreement found
400	Bad Request	E0140	Error	Payment type differs from checkout payment type
400	Bad Request	E0141	Error	Transaction unknown
400	Bad Request	E0142	Error	Invalid response from issuer
400	Bad Request	E0143	Error	Issuer's Acceptor Configuration is not valid
400	Bad Request	E0144	Error	Declined by the issuer due to invalid account
400	Bad Request	E0145	Error	Operation not supported by the issuer
400	Bad Request	E0146	Error	Offline operation not allowed by the issuer
400	Bad Request	E0147	Error	Operation declined by the issuer
400	Bad Request	E0148	Error	Communication error with the issuer
400	Bad Request	E0149	Error	Issuer's acquirer configuration is not valid
400	Bad Request	E0150	Error	Unknown issuer
400	Bad Request	E0151	Error	Invalid operation
400	Bad Request	E0152	Error	Invalid original amount
400	Bad Request	E0153	Error	Invalid transaction
400	Bad Request	E0154	Error	Transaction not allowed by the issuer for the given terminal
400	Bad Request	E0155	Error	Cash Service not available
400	Bad Request	E0156	Error	Additional customer authentication required
400	Bad Request	E0157	Error	Acquirer not registered on SPI
400	Bad Request	E0158	Error	Unavailable agreement data
400	Bad Request	E0159	Error	Invalid fee
400	Bad Request	E0160	Error	Operation temporarily unavailable
400	Bad Request	E9999	Error	Operation declined by authorisation system
503	Service unavailable	T9999	Temporary error	SIBS temporary internal error

TABLE 18 - ERROR CODES - PAYMENT REQUEST

BackOffice

HTTP **Code**	**HTTP Message**	Status Code	Status Message	Status Description
200	OK	000	Success	Success
400	Bad Request	E0200	Error	Invalid request, data is missing or is invalid
400	Bad Request	E0201	Error	Operation not allowed by the Merchant
400	Bad Request	E0202	Error	Operation not supported
400	Bad Request	E9999	Error	Operation declined by authorisation system
503	Service unavailable	T9999	Temporary error	SIBS temporary internal error

TABLE 19 - ERROR CODES - BACKOFFICE

Status

HTTP **Code**	**HTTP Message**	Status Code	Status Message	Status Description
200	OK	000	Success	Success
200	OK	000	Success	Pending
200	OK	000	Success	In Processing
200	OK	000	Success	Partial
200	OK	000	Success	Timeout
400	Bad Request	E0300	Error	Invalid request, data is missing or is invalid
400	Bad Request	E0301	Error	Transaction not found
400	Bad Request	E0302	Error	Invalid merchant or terminal code
400	Bad Request	E0303	Error	Invalid transaction ID or merchant transaction ID
400	Bad Request	E9999	Error	Operation declined by authorisation system
503	Service unavailable	T9999	Temporary error	SIBS temporary internal error

TABLE 20 - ERROR CODES - STATUS

MB WAY

HTTP **Code**	**HTTP Message**	Status Code	Status Message	Status Description
200	OK	000	Success	Success
201	OK	000	Success	Success
400	Bad Request	E0500	Error	Operation was not accepted by the card holder
400	Bad Request	E0501	Error	Invalid request, data is missing or is invalid
400	Bad Request	E0502	Error	Not possible to associate MB WAY to the provided alias
400	Bad Request	E0503	Error	The provided alias is not correct
400	Bad Request	E0504	Error	The provided alias has an invalid format
400	Bad Request	E0505	Error	The provided alias is duplicated
400	Bad Request	E0506	Error	The provided alias does not exist
404	Not Found	E0507	Error	Unknown MB WAY Authorised Payment
400	Bad Request	E0508	Error	Invalid MB WAY authorised payment amount
400	Bad Request	E0509	Error	MB WAY Authorised Payment monthly amount limit reached
400	Bad Request	E0510	Error	MB WAY Authorised Payment merchant limits reached
400	Bad Request	E0511	Error	Ineligible MB WAY Authorised Payment for the MB WAY alias
400	Bad Request	E0512	Error	MB WAY Authorised Payment invalid expiration date
400	Bad Request	E0513	Error	MB WAY Authorised Payment creation timeout
400	Bad Request	E0514	Error	Operation Rejected due to fraud suspicion
400	Bad Request	E0515	Error	MB WAY Authorised Payment not created. Cardholder did not accept the operation.
400	Bad Request	E0516	Error	MB WAY Authorised Payment is expired
400	Bad Request	E0517	Error	MB WAY Authorised Payment Cancelation timeout
400	Bad Request	E0518	Error	MB WAY Authorised Payment already cancelled
400	Bad Request	E0519	Error	Invalid operation type
400	Bad Request	E0520	Error	MB WAY Authorized payment disabled
400	Bad Request	E0521	Error	Operation declined. MB WAY alias is already associated
400	Bad Request	E0522	Error	Invalid terminal
400	Bad Request	E0523	Error	Invalid Amount
400	Bad Request	E0524	Error	Declined operation

HTTP **Code**	**HTTP Message**	Status Code	Status Message	Status Description
400	Bad Request	E0525	Error	Invalid original transaction
400	Bad Request	E0526	Error	Authorisation is cancelled.
400	Bad Request	E0527	Error	Authorisation was already used
400	Bad Request	E0528	Error	Refund amount exceeds purchase amount
400	Bad Request	E0529	Error	Invalid refund amount
400	Bad Request	E0530	Error	Cancellation amount exceeds the authorized amount
400	Bad Request	E0531	Error	MB WAY purchase with an invalid card
400	Bad Request	E0532	Error	MB WAY purchase with an invalid card data
400	Bad Request	E0533	Error	Transaction already refunded
400	Bad Request	E0534	Error	MB WAY Authorised Payment creation error
400	Bad Request	E0535	Error	Invalid Service Provider
400	Bad Request	E0536	Error	MB WAY Authorised Payment is suspended
400	Bad Request	E0537	Error	MB WAY Authorised Payment - Invalid Payment method
400	Bad Request	E0538	Error	MB WAY Authorised Payment Data is missing
400	Bad Request	E0539	Error	MB WAY Authorised Payment Service is inactive
400	Bad Request	E0540	Error	MB WAY Authorised Payment Already Exists.
400	Bad Request	E0541	Error	MB WAY Authorised Payment is Cancelled
400	Bad Request	E0542	Error	MBWAY Authorised Payment Data does not match operation data
400	Bad Request	E0543	Error	MB WAY Authorised Payment Unavailable
400	Bad Request	E0544	Error	MB WAY Authorised Payment invalid status
400	Bad Request	E9999	Error	Operation declined by authorisation system
408	Request Timeout	T9999	Temporary error	SIBS temporary internal error
503	Service Unavailable	T9999	Temporary error	SIBS temporary internal error

TABLE 21 - ERROR CODES - MB WAY

MULTIBANCO

HTTP **Code**	**HTTP Message**	Status Code	Status Message	Status Description
400	Bad Request	E0600	Error	Invalid payment entity
400	Bad Request	E0601	Error	Invalid payment reference minimum amount
400	Bad Request	E0602	Error	Invalid payment reference maximum amount
400	Bad Request	E0603	Error	Invalid currency
400	Bad Request	E0604	Error	Invalid NIB or IBAN
400	Bad Request	E0605	Error	Invalid payment reference initial date time
400	Bad Request	E0606	Error	Invalid payment reference limit date time
400	Bad Request	E0607	Error	Invalid Email
400	Bad Request	E0608	Error	Payment entity is not active
400	Bad Request	E0609	Error	Missing data required to perform the payment reference generation
400	Bad Request	E0610	Error	Payment reference generation not allowed for the payment entity
400	Bad Request	E0611	Error	Unknown payment reference
400	Bad Request	E0612	Error	Payment reference is cancelled
400	Bad Request	E0613	Error	Payment reference already paid
400	Bad Request	E0614	Error	Invalid operation
400	Bad Request	E0615	Error	Transaction with error or not found
400	Bad Request	E0616	Error	Invalid refund amount
400	Bad Request	E0617	Error	Refund amount exceeds payment amount
400	Bad Request	E0618	Error	Service Payment already refunded
400	Bad Request	E0619	Error	Service Payment refund not allowed
400	Bad Request	E9999	Error	Operation declined by authorisation system

TABLE 22 - ERROR CODES - MULTIBANCO

Card

HTTP **Code**	**HTTP Message**	Status Code	Status Message	Status Description
400	Bad Request	E0700	Error	Invalid card data. Card number, CVV or expiration date is invalid
400	Bad Request	E0701	Error	Invalid card
400	Bad Request	E0702	Error	Card brand not supported
400	Bad Request	E0703	Error	Invalid payment modality
400	Bad Request	E0704	Error	DCC not allowed
400	Bad Request	E0705	Error	Card holder not authenticated
400	Bad Request	E0706	Error	Authentication method not allowed
400	Bad Request	E9999	Error	Operation declined by authorisation system
503	Service unavailable	T9999	Temporary error	SIBS temporary error

TABLE 23 - ERROR CODES - CARD

Security

HTTP **Code**	**HTTP Message**	Status Code	Status Message	Status Description
401	Unauthorized	E0900	Error	Invalid authentication or authorisation Data
400	Bad Request	E9999	Error	Operation declined by authorisation system

TABLE 24 - ERROR CODES - SECURITY

Cashout

HTTP **Code**	**HTTP Message**	Status Code	Status Message	Status Description
200	OK	000	Success	Success
400	Bad Request	E1000	Error	Invalid authorisation token
400	Bad Request	E1001	Error	Invalid merchant or terminal code
400	Bad Request	E1002	Error	Invalid request, data is missing or is invalid
400	Bad Request	E1003	Error	Invalid terminal
400	Bad Request	E1004	Error	Invalid merchant card
400	Bad Request	E1005	Error	Invalid card
400	Bad Request	E1006	Error	Daily amount limit exceeded
400	Bad Request	E1007	Error	Operation amount limit exceeded
400	Bad Request	E1008	Error	Invalid merchant card data
400	Bad Request	E1009	Error	Insufficient funds
400	Bad Request	E1010	Error	Cashout operation is disabled
400	Bad Request	E1011	Error	Operation is not in final state
400	Bad Request	E9999	Error	Operation declined by authorisation system
503	Service unavailable	T9999	Error	SIBS temporary internal error

Glossary

Term	Definition
3-D Secure (3DS)	The Three Domain Secure (3-D Secure™ or 3DS) Protocol has been developed to improve transaction performance online and to accelerate the growth of e-commerce. The objective is to benefit all participants by providing issuer with the ability to authenticate cardholders during an online purchase, thus reducing the likelihood of fraudulent usage of payment cards and improving transaction performance.
3DS Requestor	The initiator of the authentication request. For the scope of this document, this is the Merchant.
AUTH	Authorisation
Acquirer	Acquiring bank that maintains the merchant´s bank accounts. The acquiring bank passes the merchants transactions to the issuing bank to receive payment.
AUTH	Authorisation, used to obtain a promise of payment from the issuing bank.
API	Application Programming Interface. It is a set of protocols, routines, and tools that allow different software applications to communicate with each other. Defines the methods and data structures that developers can use to interact with the software components, operating systems, or external services.
Backoffice	Allow to do the Backoffice operations, capture, refund, cancellation and MIT.
BIN	Bank Identification Number. A 6-digit number assigned by the SPI's that is used to identify the institution that issues the card.
Business Portal	Provide Merchants a single point of access to information stored and to setup notifications
Card	Payment card issued by a bank, used to have access to funds or make purchases.
Cardholder	Person who owns a banking card known as customer.
CIT	Cardholder Initiated Transaction
Customer	Also called the cardholder, who wants to access the merchant products or services
DCC	Dynamic Currency Conversion. Service that allows international cardholders to pay for goods or services in their home currency at the point of sale, instead of the local currency of the country where the transaction is being made.
Front End	Users Interface
HMAC	Hash-based Message Authentication Code. It is a type of cryptographic technique used for verifying both the integrity and the authenticity of a message or data.
Issuing Bank	Customer bank that issues the cardholders card on behalf of the brands (e.g. VISA)

MULTIBANCO	Brand associated with the network composed by ATM and POS terminals, created and managed by SIBS
MB WAY	Payment Method on mobile application associated to your banking card allowing payments card not present, through a *QR Code* or phone number associated.
MB WAY Alias	Telephone number associated to a MB WAY service
MCC	Merchant Category Code. 4 Digit-Code that classifies the business by the types of goods or services it provides.
Merchant	An on-line business offering a product or service to customers
Merchant Website	Merchant Online Shop where a customer chooses and buy products.
MIT	Merchant Initiated Transaction
Multibanco (Reference)	Payment Method Allow the client to make the checkout and then receive an Entity, reference and Amount, this can be paid on the Multibank ATM or Home banking.
MOTO	Mail Order Telephone Order – become synonymous with any financial transactions where the entity taking payment does not physically see the card used to make the payment
Payment Facilitators	Entity that sets up electronic payment and processing services for business owners.
Payment Gateway	Technology that captures and transfers payment data from the issuer (customer) to the acquirer. (used by merchants to accept payments from customers)
PCI-DSS	Payment Card Industry Data Security Standard for compliance purposes.
PURS	Purchase
SIC	Standard Industrial Classification. Classification system that uses a four-digit code for classifying a company's primary industry, based upon the company's highest revenue category.
SIBS Gateway V2	SIBS Payment Gateway that allow merchants do financial transactions with their customers.
SPI	Small Payment Institution. Type of payment services provider under the European Union's Payment Services Directive 2 (PSD2). It refers to a payment institution that offers payment services but is subject to lighter regulatory requirements compared to larger, more complex institutions. This designation applies to institutions with a lower volume of transactions.

	To qualify as an SPI, the institution's total annual payment volume must not exceed a specific threshold (typically €3 million in the EU). SPIs can provide services such as: • Payment initiation services (e.g., helping customers make payments from their bank accounts); • Money remittance services (e.g., transferring money between individuals); • Operating payment accounts.
SPG	SIBS Gateway V2 former name. SIBS Payment Gateway that allow merchants do financial transactions with their customers.
SPG-FORM	Front End (webpage) that provides the SIBS Payment Gateway Layout
Submerchant	Entity that hires Payment Facilitator companies in order to provide payment solutions to customers.
TIN	Tax Identification Number. Fiscal number that identifies an entity.