Partner in Payments

MB WAY API

Version: 01.00

Date: 2019-02-26

Status: Final

Classification: Public

Reference: DCSIBS190067

Document Info

Reference:	DCSIBS190067
Document Title:	MB WAY API
Version:	01.00
Status:	Final
Classification:	Public
Document Type:	Manual

Distribution List

Name
Comerciantes MB WAY e Developers

Version History

Version	Date	Description	Author
01.00	2019-02-26	Document creation	Product Development Unit

Table of Contents

List of Figures

List of Tables

1 Introduction

MB WAY is a service created and developed by SIBS FPS, in the scope of the most recent existing technologies that meets the current market demands. This new payments solution offers to the Merchants and Service Suppliers the possibility of providing to their clients a remote payment platform in unattended channels, such as vending machines and parking meters, as well as in attended channels such as tv-commerce, e-commerce, m-commerce and others.

In order to meet the market demands, the MB WAY service introduced a new, more integrated, payment experience for purchases performed in a Merchant mobile app. The in-app operative allows the Merchant app to invoke the MB WAY app to authenticate the transaction; afterwards, the Merchant app reappears, displaying the operation result. The Merchants and Services Suppliers can also offer the MB WAY solution in physical stores, in those cases where the “order ahead” concept represents an added value for the Merchant’s sales pipeline, or where the mobility facilitates the client check-out during the payment process (e.g.: a restaurant that has its billing application integrated in mobile devices and issues the bill in the presence of the client, at table, may now provide the MB WAY service (also integrated with the system) for the immediate payment; take-away restaurants with apps that allow the client to make remote orders and pick-up the product at the store, in a separate queue; etc.). This service lays on the relationship between financial data and the User’s mobile phone number; therefore, to use the service, the client must hold a banking card and a smartphone, to have access to an easier, more comfortable and secure payment method. The Merchants interested in offering the MB WAY service to their clients must complete the enrolment process with an Acquirer. The implementation of the technical interfaces necessary for this new payment method is made directly with SIBS FPS who supplies the solution already certified by SIBS FPS. This guide contains all the necessary technical specifications to offer the MB WAY service, by presenting the APIs that must be implemented by the Merchants. The objective of this guide is to provide to the participants the ability to plan and assess the effort of the implementation process with their technical teams.

For further information on this service and the technical details to implement the MB WAY APIs, visit the API Market website at www.sibsapimarket.com. Developers may access to a reserved area, upon registration.

1.1 Scope

This innovating solution provided by SIBS FPS associates the mobile phone number of the User to his banking card. The mobile phone number is what is called an “alias”, in the scope of MB WAY service. An “alias” is a data element that identifies the client/User. (For a better acquaintance with the acronyms and terms used along this guide, refer to Section 6 - Glossary).

The client must be a banking cardholder and have a smartphone (or any other device) with an iOS or Android operating system to be able to use this service. Considering the increasing number of Users of this type of mobile equipment, SIBS FPS expects that this solution will have a great comprehensive usage.

The User enrolls in the MB WAY service in the ATM network managed by SIBS FPS – MULTIBANCO – or in the Banking Institutions‟ channels (e.g.: home banking). In this initial phase, the User associates one or more cards to his mobile phone number and sets the MB WAY PIN; subsequently, at an app store, he must download the MB WAY authentication app, managed by SIBS FPS, that will assure the security in the capture of the User's MB WAY PIN in the mobile phone and make the app's activation. With the application already activated, the User must enter the mobile phone number and the code that he defined when he enrolled in the service in MULTIBANCO ATM network or in the home banking channel, plus the activation code that he received in the mobile phone.

Following these simple steps to successfully complete the enrolment, the client is able to make payments, by simply entering his mobile phone number, as he prefers, or what is most convenient for him, and confirm the purchase, by entering the MB WAY PIN in the MB WAY app, in his mobile phone or other device.

The transaction processing is made by SIBS FPS, based on real time authorization requests with the Banking Institution.

SIBS FPS will also supply an operating model which is more appropriate for some types of service, i.e. low value transactions and where the payment speed, ease and simplicity are differentiating and critical factors for the Merchant. In this operating model, the User can make payments without entering the MB WAY PIN for each purchase; to use this option, the User only has to make a register with a Merchant who is enrolled in this operative.

In order for these operations to be well-succeeded, the User must have his mobile phone's SIM card active to receive the SMS with the activation code and an active internet service to receive the payment notifications and the messages from the service. MB WAY allows to register other devices (mobile phones or tablets) associated to the same MB WAY service (mobile phone number).

SIBS FPS supplies the necessary interfaces, adequate to the services provided by the Merchants, namely the Application Programming Interface (API) and/or the basic web services to access the service. This guide contains the description of the technical implementation of all the MB WAY interfaces. If the Merchant opts for the in-app operative, he must refer to the *SDK (Software Development Kit) for in-app Operative* guide that describes the integration of Merchant mobile apps with the in-app operative for the different operating systems – Android and iOS.

With this functionality, the Merchants may perform the MB WAY POS closure operation at any moment, so that they can make the financial control of the business at their best convenience (refer to section 4.3).

The Merchants must enroll in the service, by contracting it with an Acquirer of the MB brand, and they may implement the necessary technical interfaces directly with SIBS FPS. After having completed one of these implementation options, the Merchant may supply this new payment method to his clients.

The figures below illustrate the enrolment process of the Merchants to the service.

First Step:

The Merchants must formalize the enrolment in the service with an Acquirer.

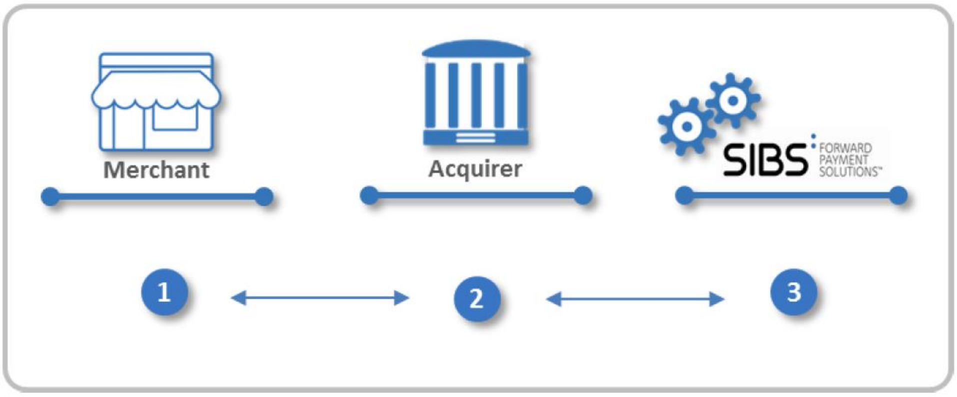


Figure 1 - Enrollment Formalization

1. The Merchant agrees the business conditions of the MB WAY service with an Acquirer.
2. The Acquirer supplies the MB WAY documentation to the Merchant.
3. The Acquirer registers the Merchant in SIBS FPS system.

Second Step:

The implementation of the technical interfaces needed to accept this payment method can be made by the Merchant himself and directly certified by SIBS FPS.

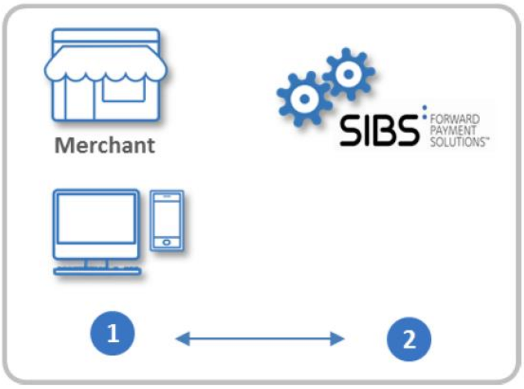


Figure 2 - Technologic adaptation by the Merchant

1. The Merchant develops his own site and makes the necessary adaptations to supply the MB WAY service.
2. The system is certified by SIBS FPS.

Additionally, SIBS FPS also offers to the Merchants the possibility of positioning an Alias at SIBS FPS, through their Acquirer, which is designated a “Merchant Alias”, different from the personal alias of the mobile phone number. This Merchant Alias facilitates the client identification, as it uses an information which is already known to the Merchant (e.g.: client number) and can also be known by SIBS FPS (SIBS FPS relates that information to the mobile phone number).

This way, the Merchant can easily identify his clients at the moment of the payment, without the need to enter any additional information. This service allows the Merchants to offer a personalized payment method to their clients.

2 Interaction Mode

This section describes the rules and guidelines that must be followed during the interaction between Merchants and SIBS FPS, to implement the technical interfaces, hereinafter designated as “RESTfull web services” or “APIs”.

2.1 Privacy and Data Protection

At the RESTfull web services implementation process, the Merchant must consider the need to post a disclaimer next to the MB WAY payment button, to guarantee that the MB WAY Users identification data, phone number, used to complete a payment, are not used for commercial distribution purposes, mass marketing or any other purpose, where personal data might be disclosed and broadcasted without the consent of the users.

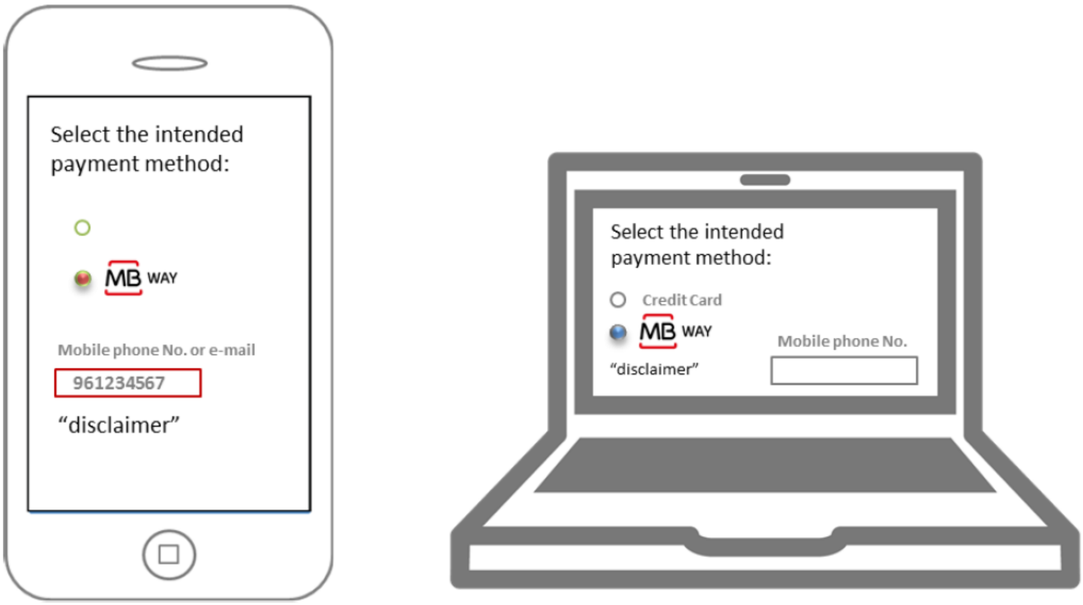

Select the intended payment method:
○
● MB WAY
Mobile phone No. or e-mail
961234567
“disclaimer”
Select the intended payment method:
○ Credit Card
● MB WAY
Mobile phone No.
“disclaimer”

Figure 3 - Example of the disclaimer positioning

2.2 Protocols

2.2.1 Message Encoding

It does not exist any standard for JSON but messages should comply with the common guidelines used, and recommended, by Google: https://google.github.io/styleguide/jsoncstyleguide.xml

2.2.2 REST-HTTPS

The web services protocol is REST, over HTTPS communications.

2.3 Rules and Formats

This section describes the rules and formats regarding the field status description. These rules and formats relate to the interaction model used in the communications with the platform.

2.3.1 General Rules and Formats

2.3.1.1 Field Status Codes

The table below shows the codes used in the communications with the platform.

Table 1 - Field Rules and Status Codes

Status	Description
[0 - 1]	Optional element. May have one occurrence .
[1 - 1]	Mandatory element. One occurrence.

2.3.2 Message Rules and Formats

This subsection presents a set of format rules and naming applied to the APIs interfaces.

2.3.2.1 Data Types

Table 2 - Data Types

Data Types		Description
Text	*String*	Any character is allowed, according to the filling rules defined and supported for the UTF-8.
Numerical	*Integer*	Any integer number (digits from ‘0’ to ‘9’) is allowed.
Date / Time	*DateTime*	The date and time are represented according to the defined format for the DateTime data type, including the option of representing the time zone (ISO8601). E.g: 2001-12-17T09:30:47Z or 2001-12-17T09:30:47+01:00

2.4 Validations and Return Codes

The return codes of the RESTfull web services sent as a result of the technical validations are described in the following table.

Table 3 - RESTfull Web Services Return Codes

Type	Code	Description	Occurrence	Suggested Message
Success	000	Success	Operation successfully completed.	**N.A.** *The Merchant must proceed with the purchase process.*
	010	Alias rejected in the application	The User rejects the Alias association.	**It was not possible to complete the identifier association operation due to rejection at the application.**
	020	Financial operation declined in the application	The User rejects the Financial Operation.	**It was not possible to complete the purchase operation due to rejection at the application.**
General Errors	100	Service internal error	These are unforeseen exception errors, when the message processing fails and the error does not fit in any defined category.	**It was not possible to complete the operation.** *It is up to the Merchant to decide if it makes another attempt or if it ends the purchase process.*
	101	*Timeout*	The maximum time set to receive the response is over.	**It was not possible to complete the <insert action> operation due to response timeout.** *Or* **Operation has expired, please try again.**
	102	Message contents validation error	The message does not comply with the defined format for the invoked service.	**N.A.** *The Merchant must validate the message format.*
	103	Invalid POS	The POS identifier was not found.	**N.A.** *The Merchant must validate the data sent in the posId attribute.*
	104	Operation declined. Problem with the service on the User's side.	The operation is not allowed to the User due to problems in his MB WAY service. E.g.: The service is blocked on the User's side.	**Operation not allowed – Check your service's current status.**
Alias Errors	110	Alias type not authorized for the Merchant	The Merchant attempts to create an Alias type which is not previously authorized for him.	**N.A.** *The Merchant must validate if the Alias type has been authorized.*
	111	Invalid Alias format	The Alias format submitted does not match the defined Alias format	**N.A.** *The Merchant must validate if the information sent in the aliasName attribute matches the defined aliasTypeCde.*

Type	Code	Description	Occurrence	Suggested Message
	112	Duplicate Alias	The Alias has already been created.	**N.A.** *The Merchant must validate if it is attempting to create an Alias duplicate.*
	113	Unknown Alias	The identifier Alias has not been found.	*It was not possible to complete the operation; please validate the entered data.*
Financial Operation Errors	120	Duplicate Financial Operation Identification	The Merchant submits an operation identified with the same ID assigned to another previous operation.	**N.A.** *The Merchant must validate if it is sending a duplicate purchase request.*
	121	Invalid Financial Operation Type	The Financial Operation type was not found.	**N.A.** *The Merchant must validate the information sent in the operationTypeCode attribute.*
	122	Operation declined	The Operation has been declined to the User.	**Operation not allowed – Check your service's current status.**
	123	Financial Operation not found	Financial Operation not found or invalid parameters submitted. E.g.: A "Refund" operation that does not match any "Purchase" operation.	**N.A.** *The Merchant must validate if it is sending the correct information regarding the referenced operation.*
	124	Alias assignment not possible	No Financial Operation, during the last 30 minutes, was found for that User. As such, no Merchant Alias can be assigned.	**N.A.** *The Merchant must validate if a financial operation was successfully completed before the Merchant Alias association request.*
	125	Rejected financial operation	Financial operation rejected to the User	**N.A.**

2.4.1 Error Control

Based on the information of the table above, the Merchant decides which actions to take in order to complete the intended operation.

2.4.2 Timeout Management

Whenever the Merchant receives a timeout message after invoking a given service, or if he does not receive the confirmation (asynchronous) after a reference time of 300 seconds (counting from the synchronous confirmation of the request reception), the Merchant may inquire the transaction status. If the operation remains without confirmation, the Merchant has to cancel the operation that he was attempting. After having checked the successful cancellation, the Merchant can resubmit the intended operation (with a new operation ID).

2.4.3 Duplicate Control

All the operations submitted by the Merchant must be identified with a unique operation ID (*transactionUniqueID*), even when repeating an operation that has been cancelled. Any request identified with an already used ID will be declined.

3 APIs Catalog

The tables in this section contain the catalogue of the RESTfull web services supplied by the MB WAY API Market.

Table 4 - List of RESTfull web services

API	***Operation***	***Functionality***
MB WAY ID Purchase	purchase	This operation starts a purchase between a merchant and a customer
	cancel	This operation cancels a purchase
	refund	This operation refunds a customer of a previously made MB Way ID purchase
	authorization	This operation will hold the amount for a future purchase
	purchaseAfterAuthorization	This operation completes a purchase that was previously checked for funds with the 'MB Way ID authorization'
	cancelAfterAuthorization	This operation cancels an authorization that was previously checked for funds with the 'MB Way ID authorization'
	inquiry	This operation allows checking the status of previously processed operations

API operations include Sales, Cancellations, Returns, Authorizations, Purchase after Authorization (Hold Amount), Authorization Cancellation and Transaction Status (up to 50 transactions)

The validation of the confirmation messages is made by matching the “token” and the transactionUniqueID fields that must equal the corresponding synchronous responses.

In the following chapters, additional information on these fields is provided.

4 APIs Description

This section presents the RESTfull web services required to execute the MB WAY operations, supplied by SIBS FPS.

4.1 MB WAY ID Purchase

The MB WAY ID Purchase API support the entire purchase process: Purchase, Refund, Purchase Authorisation, Purchase after Authorisation, Authorisation Cancellation and Cancellation.

The Merchant may also inquire the Financial Operation Status whenever the operation result is not received; this avoids that the operation might be cancelled.

Before describing each operation it is required to present the objects required to perform the operations with success. It's defined 4 major objects: Merchant, Customer, Amount and Transaction.

Table 5 - Merchant Definition

Field	***Multi***	***Type***	***Length***	***Description***
merchantToken	[1 - 1]	String	9	*The unique Merchant Identifier provided by SIBS*
transactionUniqueId	[0 - 1]	String	16	*MB WAY APIs allow each merchant to send a unique Identifier to be used for tracking purposes between the merchant and SIBS itself. We recommend using a UUID if possible*
timestamp	[0 - 1]	DateTime	23	*DateTime of the transaction*
channel	[0 - 1]	String	7	*This field should be used to identify where the API was called from. Valid values are: mobile, web, vending, pos, tv*
merchantPosId	[0 - 1]	String	5	*The POS (Point of sale) Identifier provided by SIBS*
description	[0 - 1]	String	150	*Purchase description to be presented (limited to 150 characters)*

Table 6 - Customer Definition

Field	***Multi***	***Type***	***Length***	***Description***
customerIdentifierType	[1 - 1]	String	5	*The MB WAY ID identifier type, currently supports the values 'phone'*
customerIdentifier	[1 - 1]	String	254	*Depending on type, if phone a valid phone number must be sent with the format country_code#phone_number*

Table 7 - Amount Definition

Field	***Multi***	***Type***	***Length***	***Description***
value	[1 - 1]	Double	5	*The amount value to be charged to the Customer.*

Table 8 - Transaction Definition

Field	***Multi***	***Type***	***Length***	***Description***
transactionUniqueId	[0 - 1]	String	16	*MB WAY APIs allows each merchant to send an unique Identifier to be used for tracking purposes between the merchant and SIBS itself. Although is not mandatory, we recommend using a UUID if possible*
timestamp	[0 - 1]	DateTime	23	*The original transaction DateTime*
transactionStatusCode	[0 - 1]	String	2	*The transaction Status Code of the transaction*
transactionStatusTimestamp	[0 - 1]	DateTime	23	*The DateTime when the Status Transaction changed to this value*

4.1.1 *Purchase*

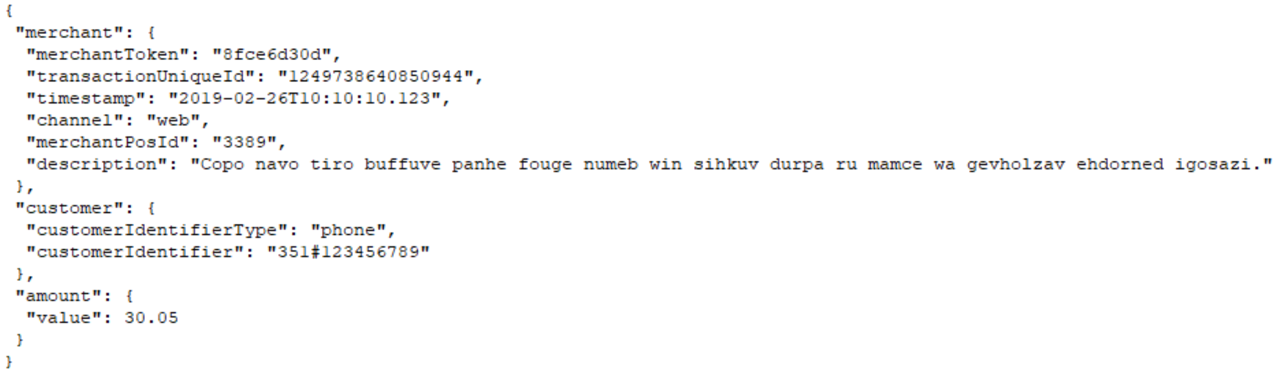
The purchase operation validates and immediately debits the customer's card (assured by the Card Issuer). This operation has accounting value since it guarantees the fulfillment of the transaction.

The required objects to process this transaction are:

- Merchant
- Customer
- Amount

4.1.1.1 *Input of the purchase operation* request

4.1.1.1.1 JSON Example



```
{
 "merchant": {
  "merchantToken": "8fce6d30d",
  "transactionUniqueId": "1249738640850944",
  "timestamp": "2019-02-26T10:10:10.123",
  "channel": "web",
  "merchantPosId": "3389",
  "description": "Copo navo tiro buffuve panhe fouge numeb win sihkuv durpa ru mamce wa gevholzav ehdorned igosazi."
 },
 "customer": {
  "customerIdentifierType": "phone",
  "customerIdentifier": "351#123456789"
 },
 "amount": {
  "value": 30.05
 }
}
```

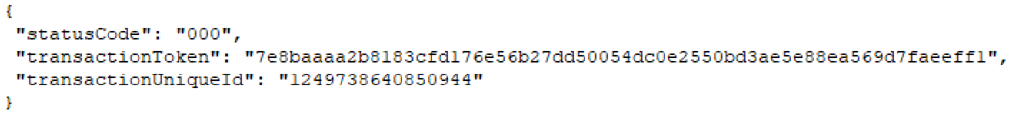
Figure 4 - JSON Example of the purchase operation request

Table 9 - Purchase Operation request – element description

Attribute/Element	**Multi**	**Type**	**Length**	**Rules/Observations**
merchant	[1 - 1]	*Merchant*	(see Table 5)	Merchant Identifier
customer	[1 - 1]	*Customer*	(see Table 6)	Customer Identifier
amount	[1 - 1]	*Amount*	(see Table 7)	Amount Identifier

4.1.1.2 *Output of the purchase operation* request

4.1.1.2.1 JSON Example



```
{
 "statusCode": "000",
 "transactionToken": "7e8baaaa2b8183cfd176e56b27dd50054dc0e2550bd3ae5e88ea569d7faeeff1",
 "transactionUniqueId": "1249738640850944"
}
```

Figure 5 - JSON Example of the purchase operation response

Table 10 - Response of the purchase operation request – element description

Attribute/Element	Multi	Type	Length	Rules/Observations
statusCode	[1 - 1]	*String*	3.	The operation result code. See table 3 for further details on error codes
transactionToken	[0 - 1]	*String*	*	This Token uniquely identifies the transaction and should be used on the remaining operations
transactionUniqueId	[0 - 1]	*String*	16	Echo of the Transaction Unique Id sent on the request

4.1.2 *Refund*

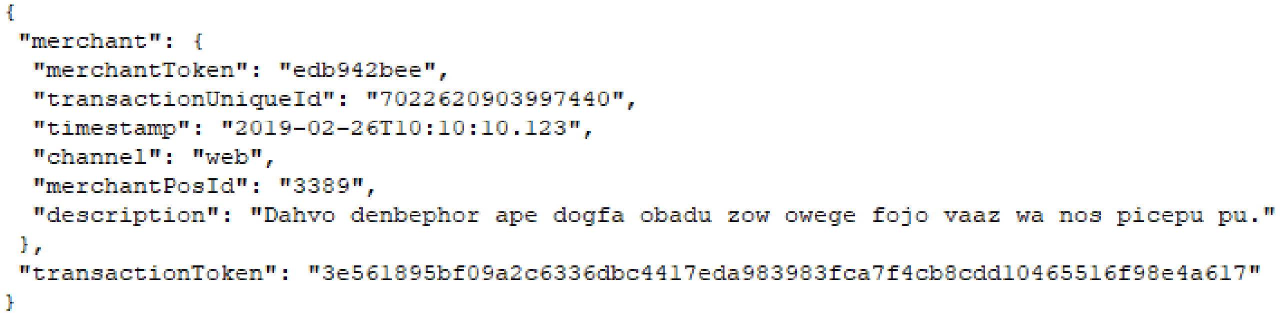
By sending a known Transaction Token, a merchant will return the amount back to the customer. This operation does not need customer approval although customer will nevertheless be notified.

The required objects to process this transaction are:

- Merchant
- transactionToken – String with the transactionToken returned by the Purchase operation.

4.1.2.1 *Input of the refund operation* request

4.1.2.1.1 JSON Example



```
{
 "merchant": {
  "merchantToken": "edb942bee",
  "transactionUniqueId": "7022620903997440",
  "timestamp": "2019-02-26T10:10:10.123",
  "channel": "web",
  "merchantPosId": "3389",
  "description": "Dahvo denbephor ape dogfa obadu zow owege fojo vaaz wa nos picepu pu."
 },
 "transactionToken": "3e561895bf09a2c6336dbc4417eda983983fca7f4cb8cdd10465516f98e4a617"
}
```

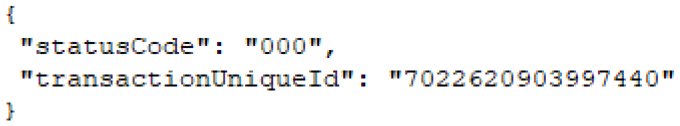
Figure 6 - JSON Example of the refund operation request

Table 11 - Refund Operation request – element description

Attribute/Element	Multi	Type	Length	Rules/Observations
merchant	[1 - 1]	*Merchant*	(see Table 5)	Merchant Identifier
transactionToken	[1 - 1]	*String*	*	Token generated on purchase operation that is used to identify the exact purchase to be refunded.

4.1.2.2 *Output of the refund operation* request

4.1.2.2.1 JSON Example



```
{
 "statusCode": "000",
 "transactionUniqueId": "7022620903997440"
}
```

Figure 7 - JSON Example of the refund operation response

Table 12 - Response of the refund operation request – element description

Attribute/Element	Multi	Type	Length	Rules/Observations
statusCode	[1 - 1]	*String*	3.	The operation result code. See table 3 for further details on error codes
transactionUniqueId	[0 - 1]	*String*	16	Echo of the Transaction Unique Id sent on the request

4.1.3 *Authorization*

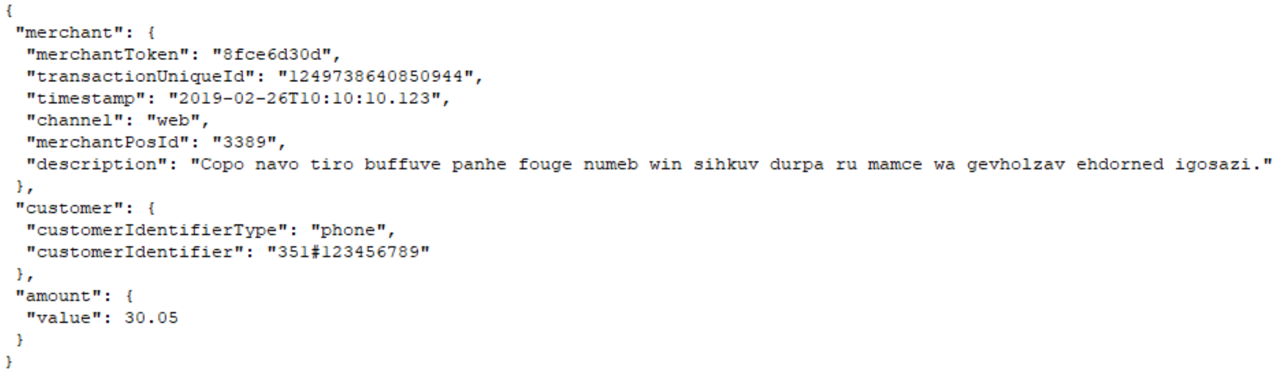
The authorization operation validates and immediately holds the amount on the customer's card (assured by the Card Issuer) so that it can be used later on a future purchase. This will allow the merchant to be sure that the customer has funds to do a future payment.

The required objects to process this transaction are:

- Merchant
- Customer
- Amount

4.1.3.1 *Input of the authorization operation* request

4.1.3.1.1 JSON Example



```
{
 "merchant": {
  "merchantToken": "8fce6d30d",
  "transactionUniqueId": "1249738640850944",
  "timestamp": "2019-02-26T10:10:10.123",
  "channel": "web",
  "merchantPosId": "3389",
  "description": "Copo navo tiro buffuve panhe fouge numeb win sihkuv durpa ru mamce wa gevholzav ehdorned igosazi."
 },
 "customer": {
  "customerIdentifierType": "phone",
  "customerIdentifier": "351#123456789"
 },
 "amount": {
  "value": 30.05
 }
}
```

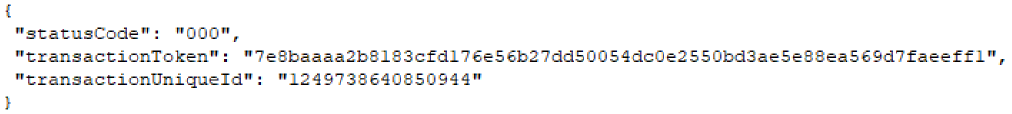
Figure 8 - JSON Example of the authorization operation request

Table 13 - Authorization Operation request – element description

Attribute/Element	Multi	Type	Length	Rules/Observations
merchant	[1 - 1]	*Merchant*	(see Table 5)	Merchant Identifier
customer	[1 - 1]	*Customer*	(see Table 6)	Customer Identifier
amount	[1 - 1]	*Amount*	(see Table 7)	Amount Identifier

4.1.3.2 *Output of the authorization operation* request

4.1.3.2.1 JSON Example



```
{
 "statusCode": "000",
 "transactionToken": "7e8baaaa2b8183cfd176e56b27dd50054dc0e2550bd3ae5e88ea569d7faeeff1",
 "transactionUniqueId": "1249738640850944"
}
```

Figure 9 - JSON Example of the authorization operation response

Table 14 - Response of the authorization operation request – element description

Attribute/Element	Multi	Type	Length	Rules/Observations
statusCode	[1 - 1]	*String*	3.	The operation result code. See table 3 for further details on error codes
transactionToken	[0 - 1]	*String*	*	This Token uniquely identifies the transaction and should be used on the remaining operations
transactionUniqueId	[0 - 1]	*String*	16	Echo of the Transaction Unique Id sent on the request

4.1.4 *Purchase after Authorization*

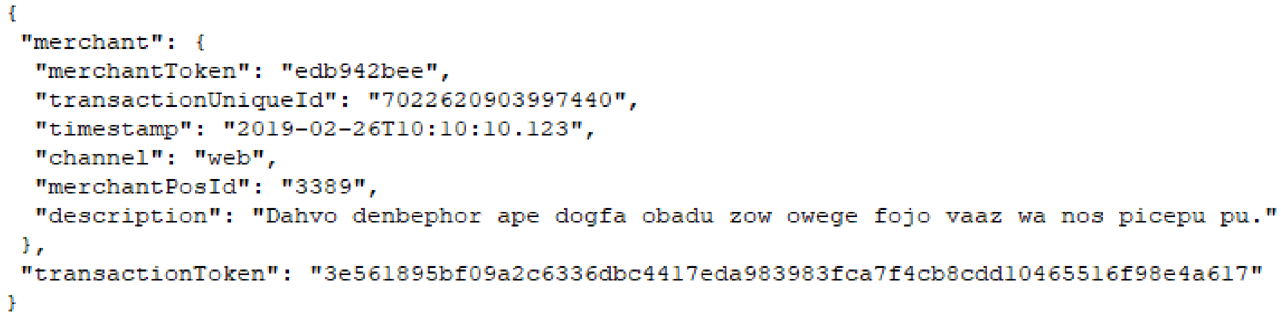
The purchase after authorization operation validates and immediately debits the customer's card (assured by the Card Issuer) by using the previously held funds. This operation has accounting value since it guarantees the fulfillment of the transaction.

The required objects to process this transaction are:

- Merchant
- transactionToken – String with the transactionToken returned by the Purchase operation.

4.1.4.1 *Input of the purchase after authorization operation* request

4.1.4.1.1 JSON Example



```
{
 "merchant": {
  "merchantToken": "edb942bee",
  "transactionUniqueId": "7022620903997440",
  "timestamp": "2019-02-26T10:10:10.123",
  "channel": "web",
  "merchantPosId": "3389",
  "description": "Dahvo denbephor ape dogfa obadu zow owege fojo vaaz wa nos picepu pu."
 },
 "transactionToken": "3e561895bf09a2c6336dbc4417eda983983fca7f4cb8cdd10465516f98e4a617"
}
```

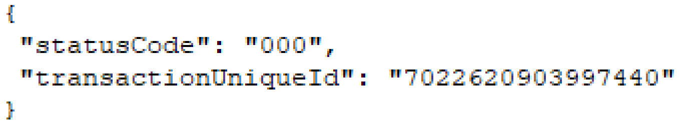
Figure 10 - JSON Example of the purchase after authorization operation request

Table 15 - Purchase after Authorization Operation request – element description

Attribute/Element	Multi	Type	Length	Rules/Observations
merchant	[1 - 1]	*Merchant*	(see Table 5)	Merchant Identifier
transactionToken	[1 - 1]	*String*	*	Token generated on authorization operation that is used to identify the exact purchase to be refunded.

4.1.4.2 *Output of the purchase after authorization operation* request

4.1.4.2.1 JSON Example



```
{
 "statusCode": "000",
 "transactionUniqueId": "7022620903997440"
}
```

Figure 11 - JSON Example of the purchase after authorization operation response

Table 16 - Response of the purchase after authorization operation request – element description

Attribute/Element	Multi	Type	Length	Rules/Observations
statusCode	[1 - 1]	*String*	3.	The operation result code. See table 3 for further details on error codes
transactionUniqueId	[0 - 1]	*String*	16	Echo of the Transaction Unique Id sent on the request

4.1.5 *Cancel after Authorization*

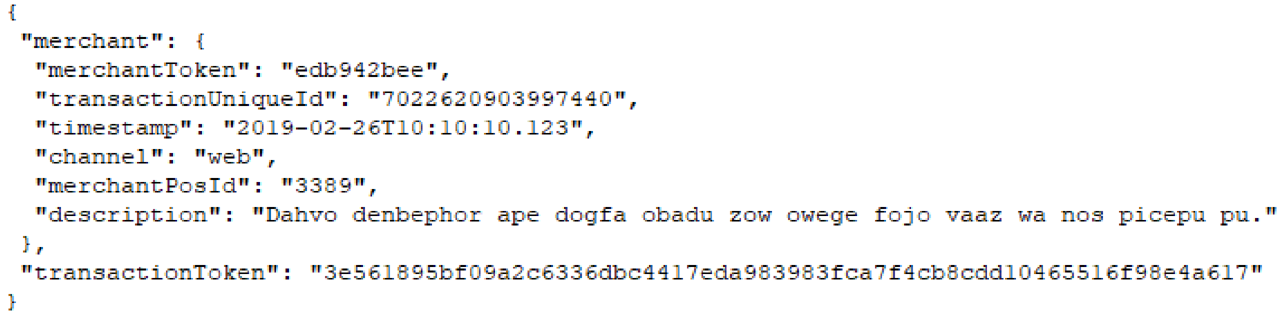
By sending a known Transaction Token, a merchant can cancel the authorization previously made. This operation doesn't need customer approval although customer will nevertheless be notified.

The required objects to process this transaction are:

- Merchant
- transactionToken – String with the transactionToken returned by the Purchase operation.

4.1.5.1 *Input of the cancel after authorization operation* request

4.1.5.1.1 JSON Example



```
{
 "merchant": {
  "merchantToken": "edb942bee",
  "transactionUniqueId": "7022620903997440",
  "timestamp": "2019-02-26T10:10:10.123",
  "channel": "web",
  "merchantPosId": "3389",
  "description": "Dahvo denbephor ape dogfa obadu zow owege fojo vaaz wa nos picepu pu."
 },
 "transactionToken": "3e561895bf09a2c6336dbc4417eda983983fca7f4cb8cdd10465516f98e4a617"
}
```

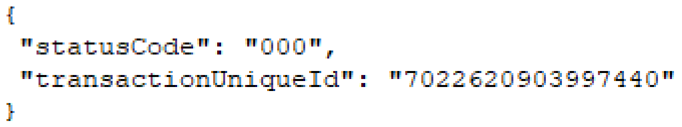
Figure 12 - JSON Example of the cancel after authorization operation request

Table 17 - Cancel after Authorization Operation request – element description

Attribute/Element	Multi	Type	Length	Rules/Observations
merchant	[1 - 1]	*Merchant*	(see Table 5)	Merchant Identifier
transactionToken	[1 - 1]	*String*	*	Token generated on authorization operation that is used to identify the exact purchase to be refunded.

4.1.5.2 *Output of the cancel after authorization operation* request

4.1.5.2.1 JSON Example



```
{
 "statusCode": "000",
 "transactionUniqueId": "7022620903997440"
}
```

Figure 13 - JSON Example of the cancel after authorization operation response

Table 18 - Response of the cancel after authorization operation request – element description

Attribute/Element	Multi	Type	Length	Rules/Observations
statusCode	[1 - 1]	*String*	3.	The operation result code. See table 3 for further details on error codes
transactionUniqueId	[0 - 1]	*String*	16	Echo of the Transaction Unique Id sent on the request

4.1.6 *Cancel*

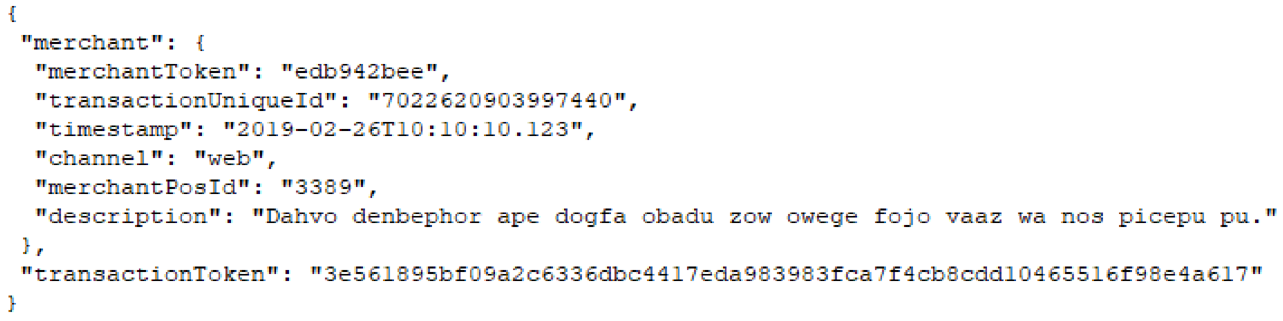
By sending a known Transaction Token, a merchant will cancel a previously made purchase. This operation does not need customer approval although customer will nevertheless be notified.

The required objects to process this transaction are:

- Merchant
- transactionToken – String with the transactionToken returned by the Purchase operation.

4.1.6.1 *Input of the cancel operation* request

4.1.6.1.1 JSON Example



```
{
 "merchant": {
  "merchantToken": "edb942bee",
  "transactionUniqueId": "7022620903997440",
  "timestamp": "2019-02-26T10:10:10.123",
  "channel": "web",
  "merchantPosId": "3389",
  "description": "Dahvo denbephor ape dogfa obadu zow owege fojo vaaz wa nos picepu pu."
 },
 "transactionToken": "3e561895bf09a2c6336dbc4417eda983983fca7f4cb8cdd10465516f98e4a617"
}
```

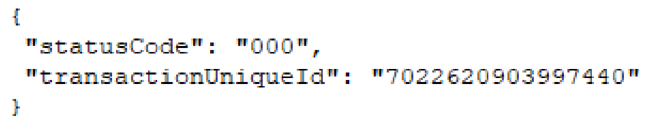
Figure 14 - JSON Example of the cancel operation request

Table 19 - Cancel Operation request – element description

Attribute/Element	Multi	Type	Length	Rules/Observations
merchant	[1 - 1]	*Merchant*	(see Table 5)	Merchant Identifier
transactionToken	[1 - 1]	*String*	*	Token generated on purchase operation that is used to identify the exact purchase to be refunded.

4.1.6.2 *Output of the cancel operation* request

4.1.6.2.1 JSON Example



```
{
 "statusCode": "000",
 "transactionUniqueId": "7022620903997440"
}
```

Figure 15 - JSON Example of the cancel operation response

Table 20 - Response of the cancel operation request – element description

Attribute/Element	Multi	Type	Length	Rules/Observations
statusCode	[1 - 1]	*String*	3.	The operation result code. See table 3 for further details on error codes
transactionUniqueId	[0 - 1]	*String*	16	Echo of the Transaction Unique Id sent on the request

4.1.7 *Inquiry*

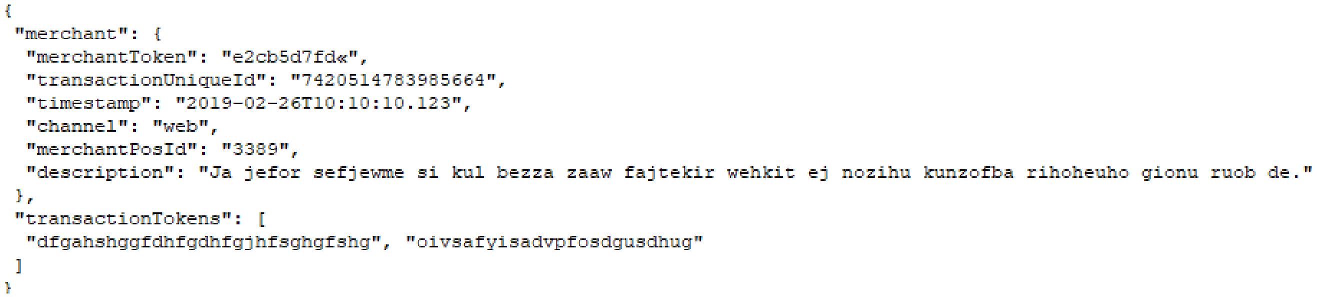
By sending a list of up to 50 known Transaction Tokens, this operation returns the status for each one.

The required objects to process this transaction are:

- Merchant
- transactionTokens – Array of transactionToken that can identify the previously processed transactions.

4.1.7.1 *Input of the inquiry operation* request

4.1.7.1.1 JSON Example



```
{
 "merchant": {
  "merchantToken": "e2cb5d7fd«",
  "transactionUniqueId": "7420514783985664",
  "timestamp": "2019-02-26T10:10:10.123",
  "channel": "web",
  "merchantPosId": "3389",
  "description": "Ja jefor sefjewme si kul bezza zaaw fajtekir wehkit ej nozihu kunzofba rihoheuho gionu ruob de."
 },
 "transactionTokens": [
  "dfgahshggfdhfgdhfgjhfsghgfshg", "oivsafyisadvpfosdgusdhug"
 ]
}
```

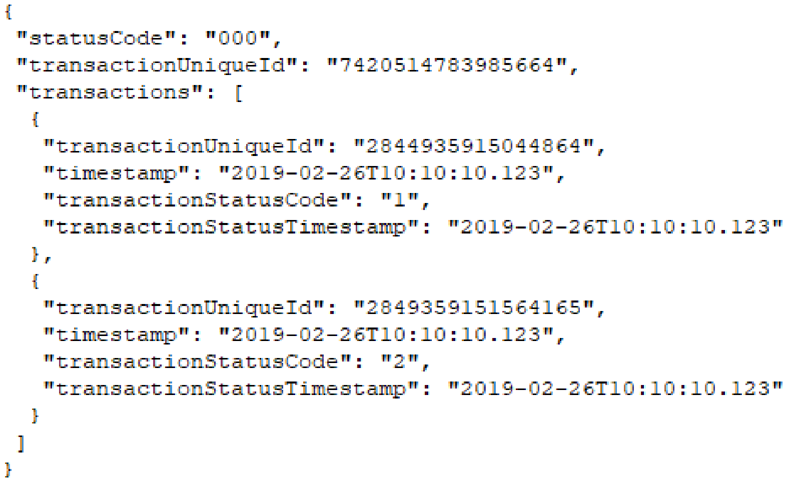
Figure 16 - JSON Example of the inquiry operation request

Table 21 - Inquiry Operation request – element description

Attribute/Element	Multi	Type	Length	Rules/Observations
merchant	[1 - 1]	*Merchant*	(see Table 5)	Merchant Identifier
transactionTokens	[1 - 50]	*String*	*	Array of Tokens generated that can identify a previously processed transaction.

4.1.7.2 *Output of the cancel operation* request

4.1.7.2.1 JSON Example



```
{
 "statusCode": "000",
 "transactionUniqueId": "7420514783985664",
 "transactions": [
  {
   "transactionUniqueId": "2844935915044864",
   "timestamp": "2019-02-26T10:10:10.123",
   "transactionStatusCode": "1",
   "transactionStatusTimestamp": "2019-02-26T10:10:10.123"
  },
  {
   "transactionUniqueId": "2849359151564165",
   "timestamp": "2019-02-26T10:10:10.123",
   "transactionStatusCode": "2",
   "transactionStatusTimestamp": "2019-02-26T10:10:10.123"
  }
 ]
}
```

Figure 17 - JSON Example of the inquiry operation response

Table 22 - Response of the cancel operation request – element description

Attribute/Element	Multi	Type	Length	Rules/Observations
statusCode	[1 - 1]	*String*	3.	The operation result code. See table 3 for further details on error codes
transactionUniqueId	[0 - 1]	*String*	16	Echo of the Transaction Unique Id sent on the request

4.1.8 *Get Token*

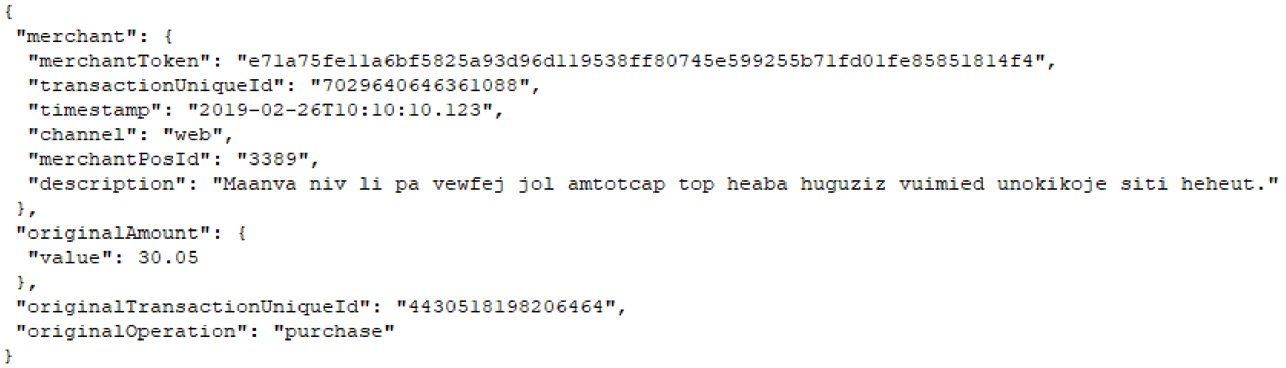
Whenever needed, the merchant can ask back what was the Transaction Token of a previously 'purchase' or ‘authorization’. This operation is usually only needed if the merchant never got or lost the previously generated Transaction Token.

The required objects to process this transaction are:

- Merchant
- Original Amount – amount processed in the original transaction.
- originalTransactionUniqueId – transactionUniqueId used in the original transaction
- originalOperation – name of the original operation (purchase or authorization).

4.1.8.1 *Input of the getToken operation* request

4.1.8.1.1 JSON Example



```
{
 "merchant": {
  "merchantToken": "e71a75fe11a6bf5825a93d96d119538ff80745e599255b71fd01fe85851814f4",
  "transactionUniqueId": "7029640646361088",
  "timestamp": "2019-02-26T10:10:10.123",
  "channel": "web",
  "merchantPosId": "3389",
  "description": "Maanva niv li pa vewfej jol amtotcap top heaba huguziz vuimied unokikoje siti heheut."
 },
 "originalAmount": {
  "value": 30.05
 },
 "originalTransactionUniqueId": "4430518198206464",
 "originalOperation": "purchase"
}
```

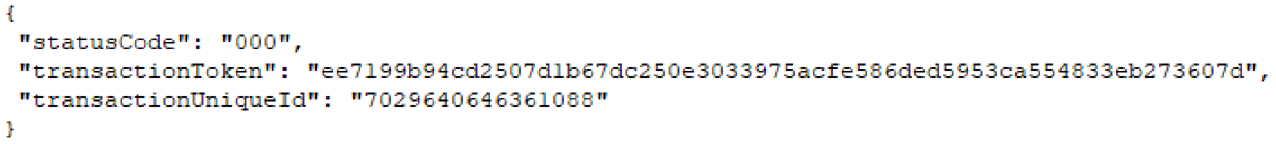
Figure 18 - JSON Example of the getToken operation request

Table 23 - getToken Operation request – element description

Attribute/Element	Multi	Type	Length	Rules/Observations
merchant	[1 - 1]	*Merchant*	(see Table 5)	Merchant Identifier
originalAmount	[0 - 1]	*Amount*	(see Table 7)	amount processed in the original transaction
originalTransactionUniqueId	[0 - 1]	*String*	16	transactionUniqueId used in the original transaction
originalOperation	[0 - 1]	*String*	13	name of the original operation (purchase or authorization).

4.1.8.2 *Output of the getToken operation* request

4.1.8.2.1 JSON Example



```
{
 "statusCode": "000",
 "transactionToken": "ee7199b94cd2507d1b67dc250e3033975acfe586ded5953ca554833eb273607d",
 "transactionUniqueId": "7029640646361088"
}
```

Figure 19 - JSON Example of the getToken operation response

Table 24 - Response of the getToken operation request – element description

Attribute/Element	Multi	Type	Length	Rules/Observations
statusCode	[1 - 1]	*String*	3.	The operation result code. See table 3 for further details on error codes
transactionToken	[0 - 1]	*String*	*	This Token uniquely identifies the transaction and should be used on the remaining operations
transactionUniqueId	[0 - 1]	*String*	16	Echo of the Transaction Unique Id sent on the request

5 Use Cases

The table below contains a list of the most relevant use cases of the MB WAY ID Purchase API where the Merchant application participates. The sections of this chapter detail each one of the use cases here represented.

Table 25 - Use Cases

Activity	ID	Use Case
Purchase	UC01	A simple purchase done with success
	UC02	A simple purchase done with error
	UC03	Simple purchase done with success. After that, the client requests a refund.
Authorization	UC04	Authorization and purchase both done with success.
GetToken	UC05	When a merchant loses the transactionToken is possible to request it to the system

5.1 Purchase

5.1.1 UC01

The first use case is a simple purchase request done with success.

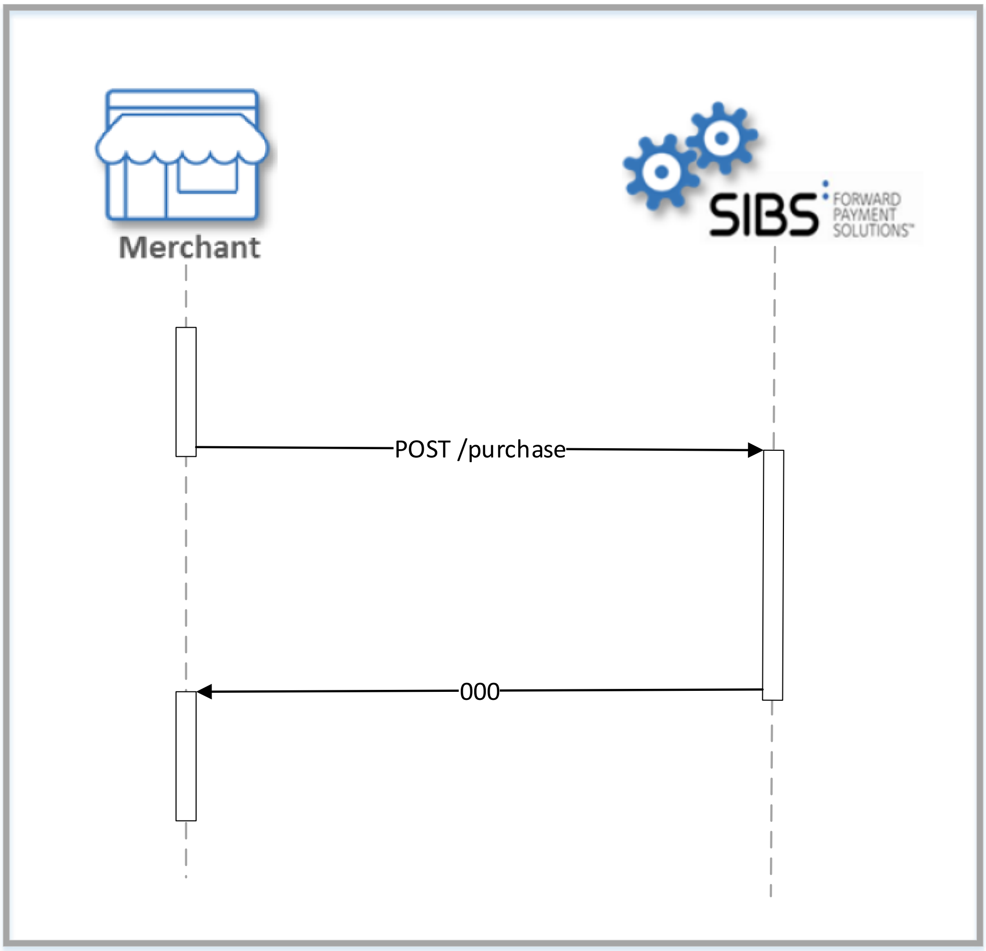


Figure 4 - Use case of a simple purchase request

5.1.2 UC02

This use case is very similar to the previous but with an error response.

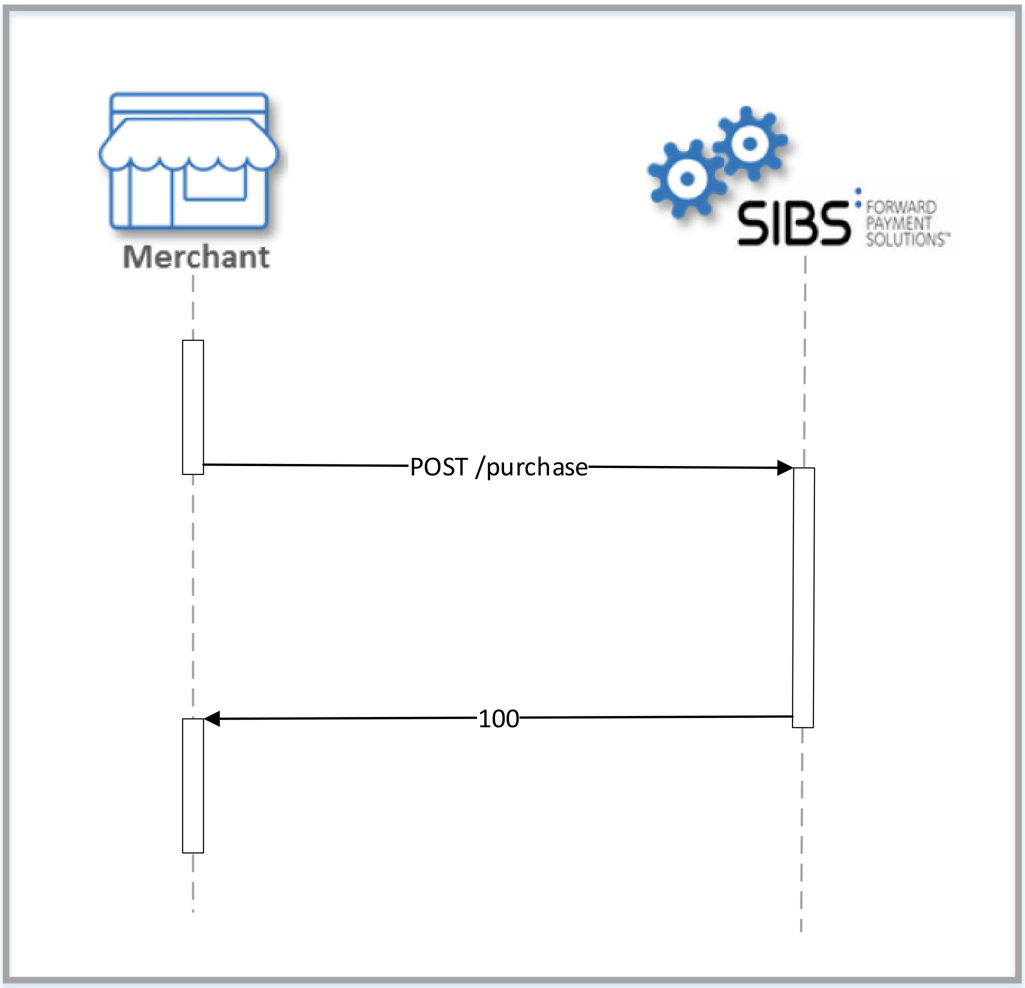


Figure 5 - Use case of a simple purchase request with error

5.1.3 *UC03*

This use case is a bit more complex than the first two. In this use case the merchant requests a purchase that was processed with success. After some time the client decides that he wants a refund and informs the merchant about it. The merchant then invokes the refund API to refund the customer.

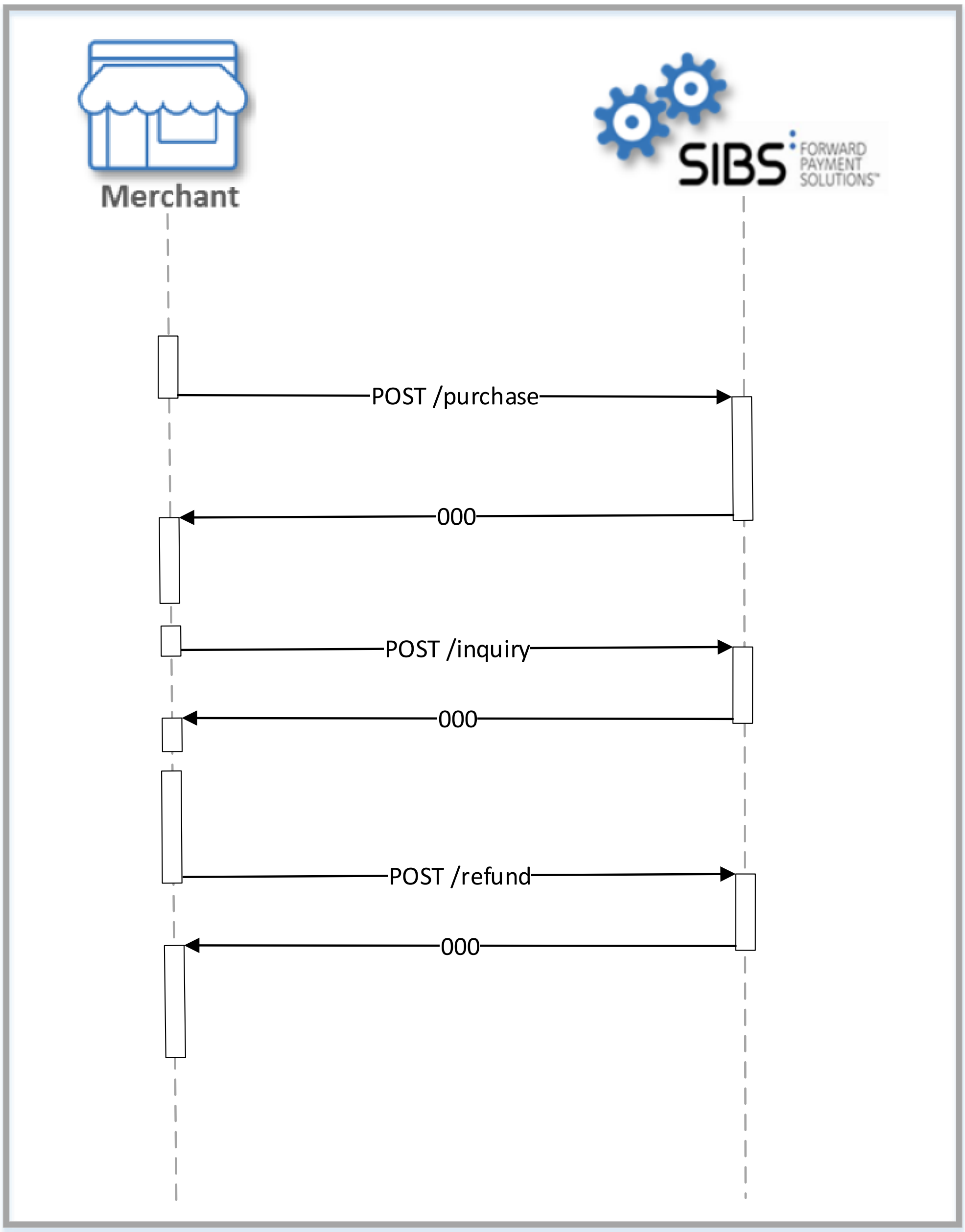


Figure 6 - Use case of a simple purchase request and a refund

5.2 Authorization

5.2.1 UC04

This is simple use case of an authorization and purchase. Initially the merchant request an authorization to captive the amount required for the purchase. When he gets a success answer he can proceed to invoke the purchaseAfterAuthorization API to finalize the purchase processs.

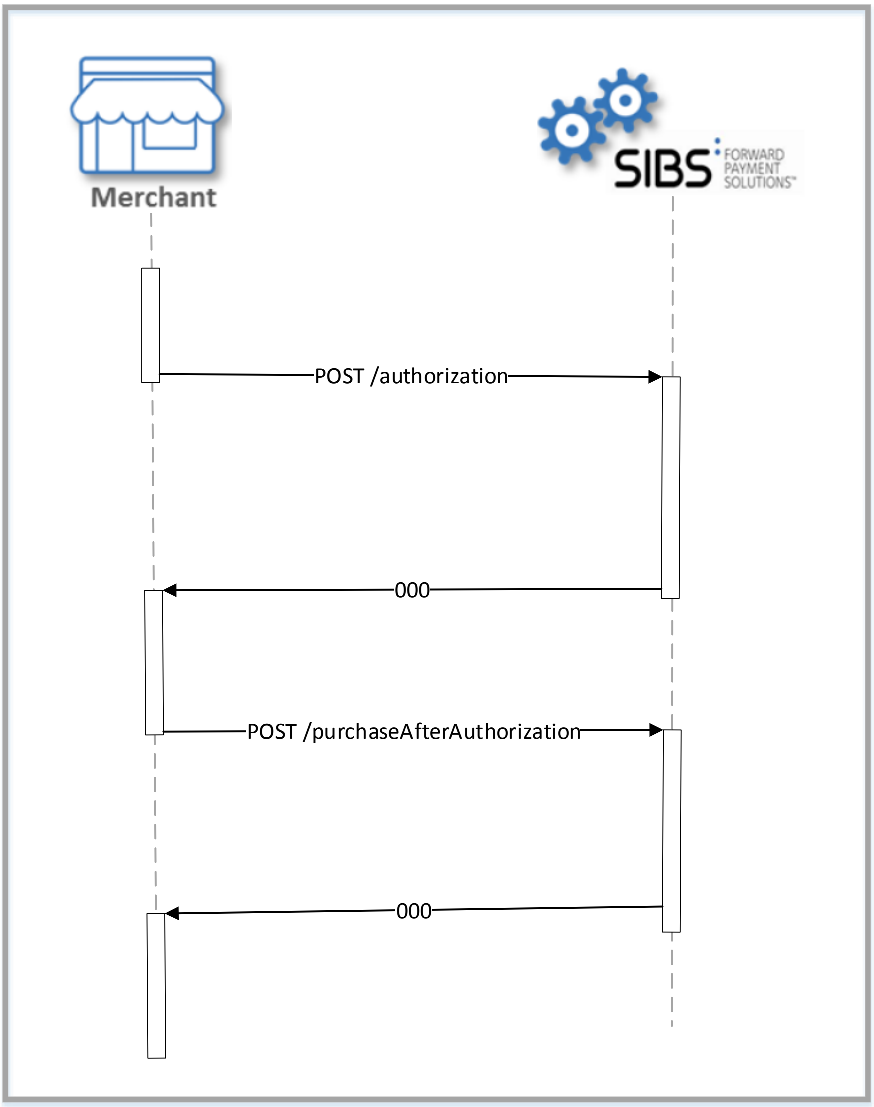

Merchant
SIBS FORWARD PAYMENT SOLUTIONS
POST /authorization
000
POST /purchaseAfterAuthorization
000

Figure 7 - Use case of an authorization request and a purchase

5.3 GetToken

5.3.1 UC06

It can be necessary for the merchant to inquiry SIBS FPS for a transactionToken of a determined transaction and at this point he invokes the getToken API with the original transaction information and he gets, if the request is rightly made, the transaction token associated with the transaction that he defined.

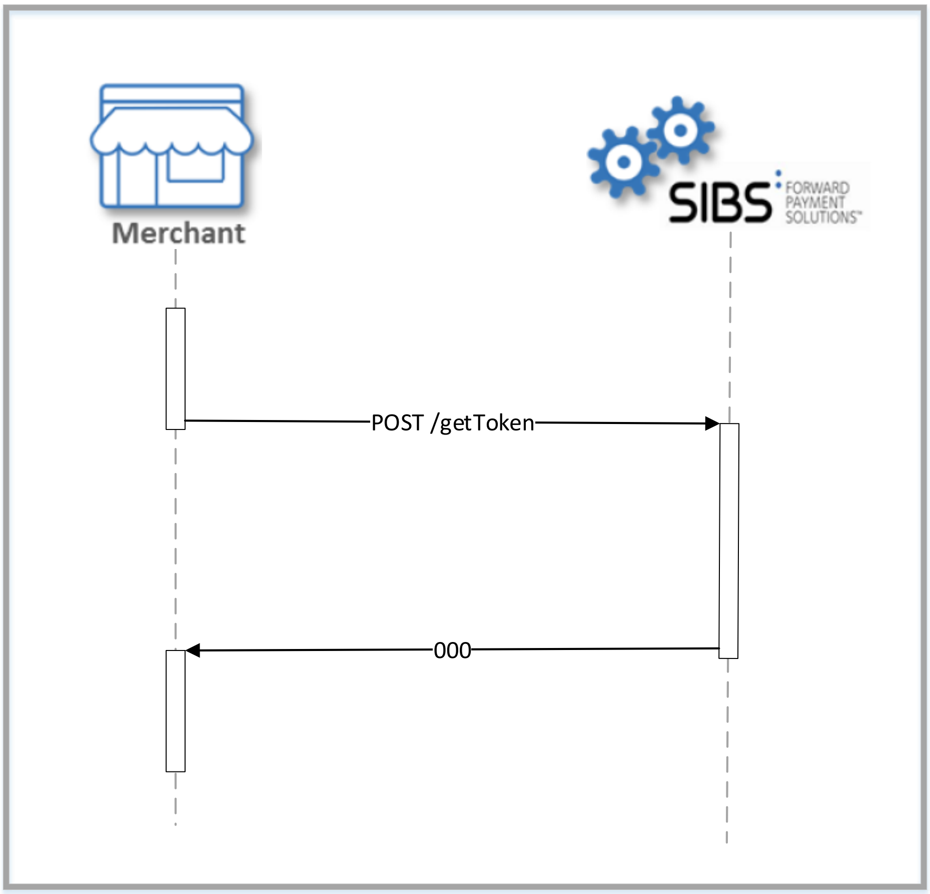


Figure 8 - Use case of a getToken request

5.4 Contact for Support

Scope	Contact
• Service enrolment • Acceptance Tests • Questions and clarification requests	mbwaysupport@sibs.pt

6 Glossary

Term	Definition
API	The Application Programming Interface (API) is a set of routines, protocols and tools for building software applications. This interface enables a program to gain access to facilities within an application, without interfering in the implemented software.
MB WAY App	The authentication application managed by SIBS FPS, installed in a smartphone of Android or iOS operating system, which assures the security during the capture of the User's MB WAY PIN and its communication to SIBS FPS for transaction processing
App store	The store of operating systems suppliers (Apple store – iOS and GooglePlay operating systems – and Android operating system) that provide the applications to be downloaded to the mobile equipment. The MB WAY app will be available in the app stores .
Activation code	The element used to activate the MB WAY app in a mobile equipment. It serves to guarantee that the User accesses the intended phone and/or to guarantee the association of the equipment used in the enrolment process.
ID	Identifier
Service limits	Several daily limits are defined by SIBS FPS, Banking Institutions and Clients.
N.A.	Not Applicable
MB WAY PIN	The code that authenticates the transactions made with all the cards associated to MB WAY. It is associated to the service, and not to the card. It is defined during the enrolment process and it can be changed lately in the banking channels, in the mobile application and in MULTIBANCO ATM terminals.
POS	*Point of Sales*
SIBS FPS	SIBS *Forward Payment Solutions*

Annex A. MB WAY brand mark use and specifications

This Annex presents a set of standards and rules which have been created to guarantee the correct use of the MB WAY brand mark.

MB WAY is a registered trademark® owned by SIBS SGPS and managed by SIBS FPS, both companies pertaining to SIBS Group. SIBS SGPS owns all industrial property, intellectual property, copyright and all other related legal rights on the MB WAY trademark.

All disclosure of information regarding the trademark requires a previous authorization of the company and its disclosure cannot be provided or facilitated, in whole or in part, to any third party without the prior written consent of SIBS FPS.

A.1. MB WAY brand mark

The MB WAY brand mark is composed of a symbol (MB) and a name (WAY) and it was produced to represent the new mobile payment method created and supplied by SIBS Group.

All the brand elements have been carefully studied and must not be altered in any way, in order to guarantee a standardized and uniform display of the brand mark at the clients and at the suppliers.

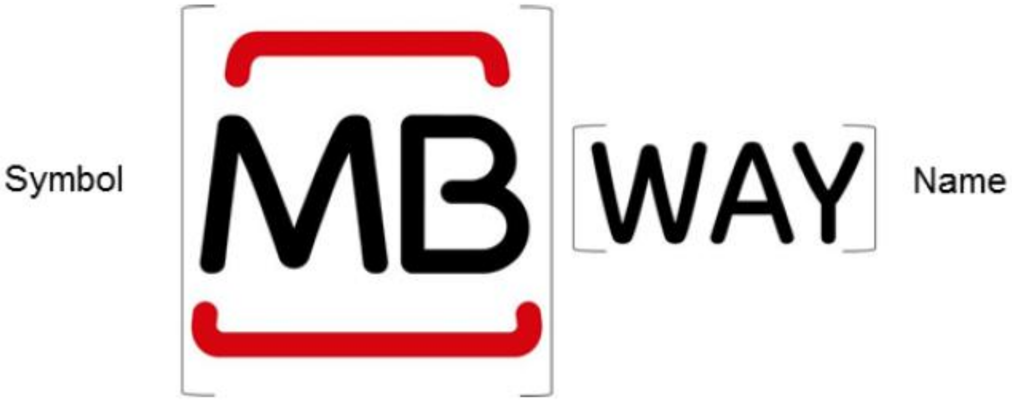


Figure 9 - Brand

- Symbol: Besides being the verbal identity of the brand mark, the symbol is an integrating part of the logotype and, in exceptional situations, it can be used alone (e.g.: app identifier);
- Name: It is also part of the brand mark, however, it cannot be used alone, or separately.

A.2. Rules for MB WAY brand use

The graphic rules created for the MB WAY brand assure the consistency of the brand and contribute to its recognition.

A.2.1 Versions

The logotype has two versions: colored and black & white.

Colored

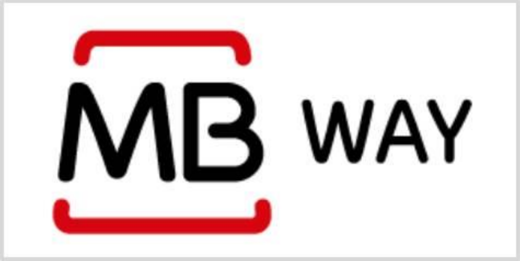


Figure 10 - Colored logotype – positive

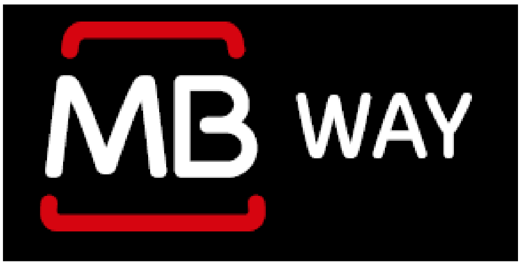


Figure 11 - Colored logotype – negative

Black and White

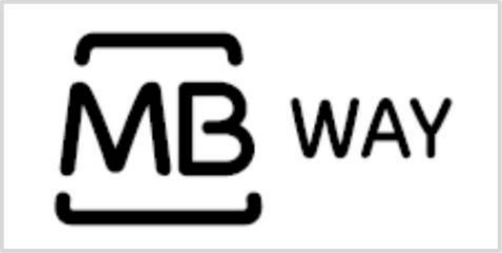


Figure 12 - Black & white logotype – positive

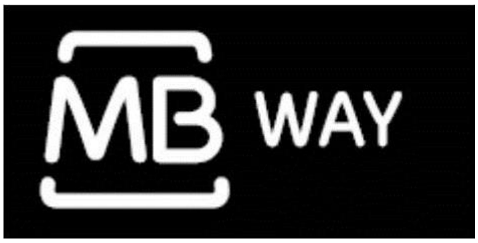


Figure 13 - Black & white logotype – negative

A.2.2 Protection Area

Aiming at preserving the normal legibility of the logotype, a security area was defined, around it, to allow protecting it from external elements that could be near and interfere in its legibility. Whenever possible, this protection area shall be enlarged.

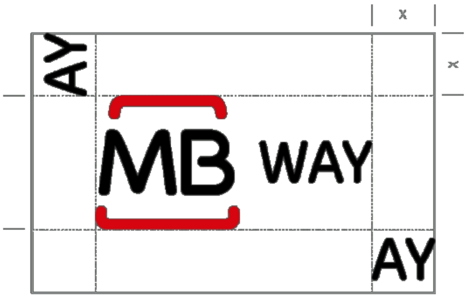


Figure 14 - Logotype Protection Area

A.2.3 Size and proportions

The logotype size and proportions must be respected to guarantee the brand mark legibility.

Minimum size requirements

The signature enlargement is limitless. The minimum size of the signature is its maximum reduction value, without any partial or total loss of its visual qualities.

For printing purposes, the logotype shall not be reproduced in a size lower than 20 mm.

To be reproduced on screen, the logotype shall not have less than 57 px.

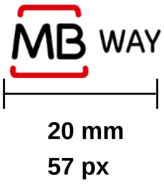


Figure 15 - Logotype minimum size

Building grid

The relationship between the logotype elements (symbol and name) were carefully studied and must not, under any circumstance, be altered.

The proportions must be preserved, regardless of the logotype dimension.

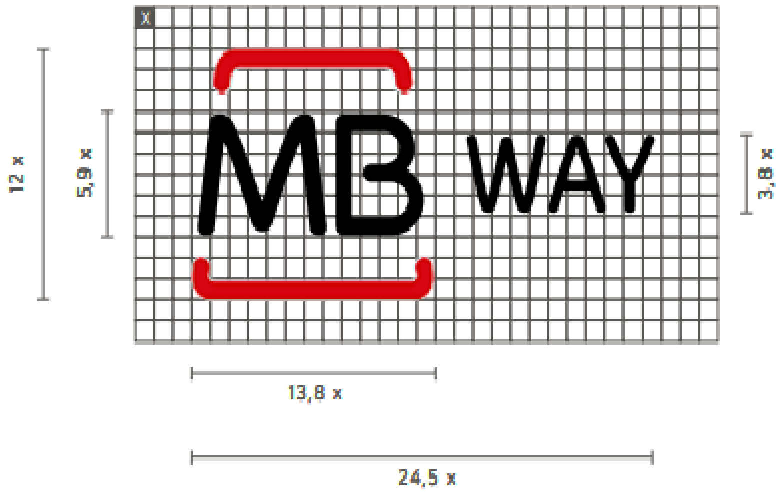


Figure 16 - Logotype building grid

A.2.4 Color combination

Colours are an essential element of the brand identity and, so, they must be reproduced as faithfully as possible. Colours must always be reproduced in Pantone direct colours.

Whenever this is not possible, other systems of colour reproduction can be used, as defined.

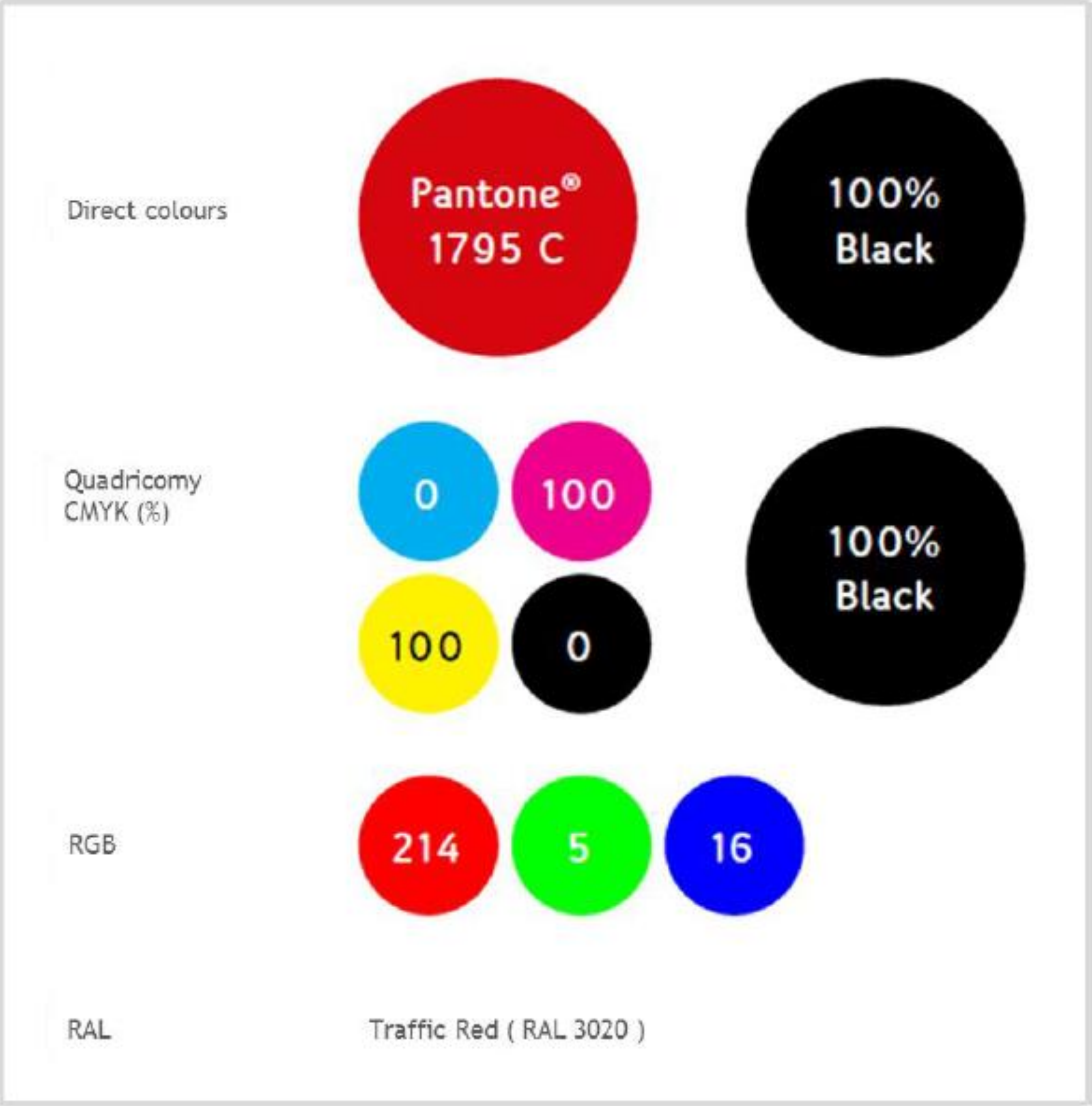


Figure 17 - Colors to be used in the logotype reproduction

A.2.5 Colors background

The basic principle for applying the brand mark over a coloured background is to maintain the chromatic integrity, with the biggest possible contrast between the logotype and the background colour.

The examples below shall serve as a reference to assure the legibility of the logotype.

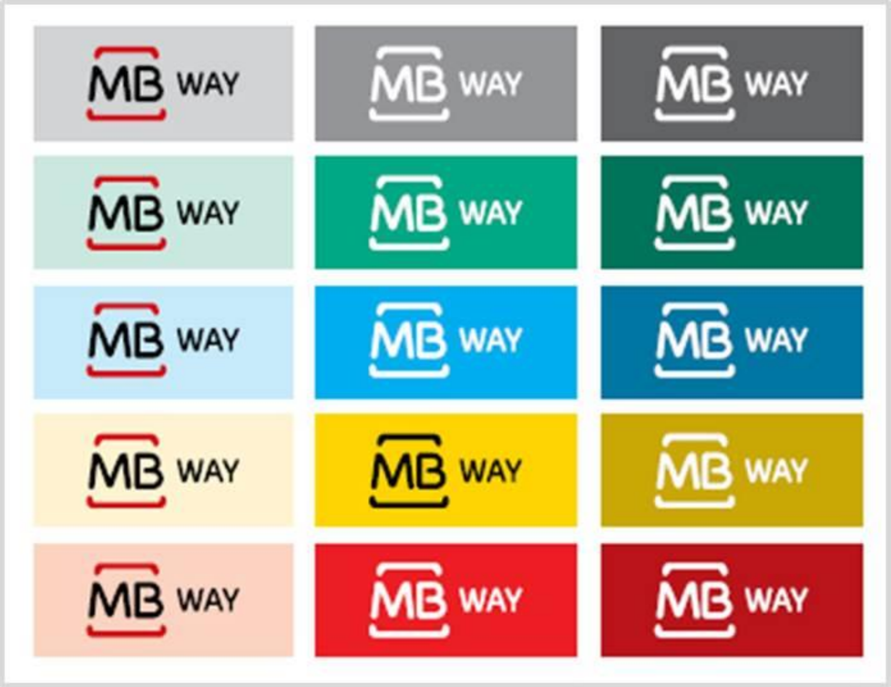


Figure 18 - Logotype application over colored backgrounds

A.2.6 Photographic background

On what concerns the application of the brand mark over photographic backgrounds, the application of the logotype over images is not recommended. However, if the reproduction of the logotype over a photographic background is indispensable, it must be placed on the areas of the image that allow the appropriate legibility of the logotype.

The examples below shall serve as a reference to assure the legibility of the logotype.

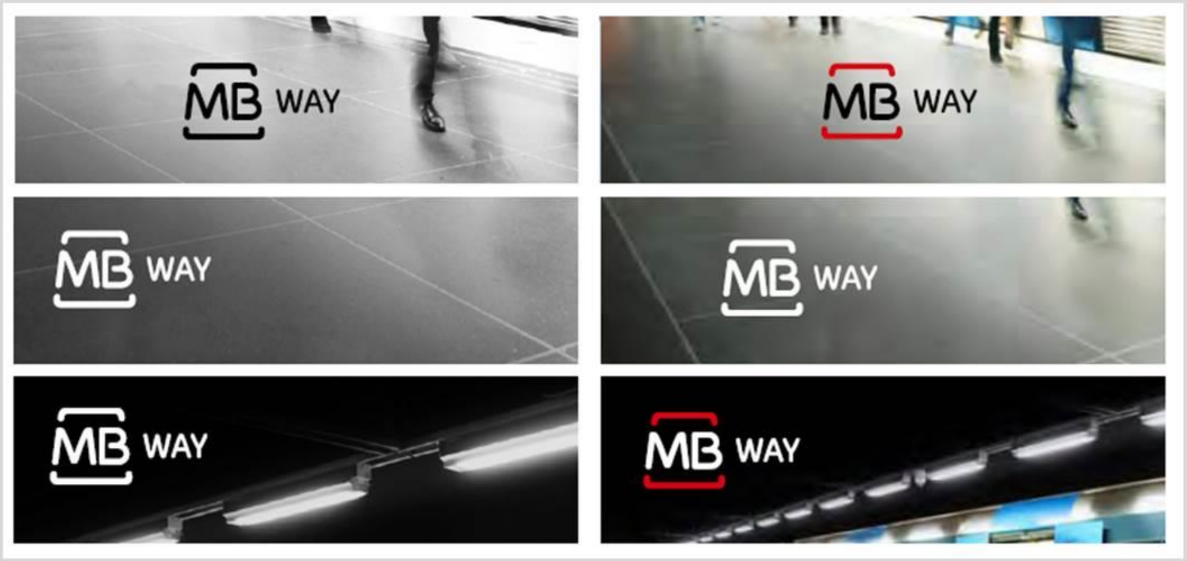


Figure 19 - Logotype application over photographic backgrounds

A.2.7 Usage of the brand in text

The usage of the brand in text must comply with the following rules:

1. The brand name must always be written in capital letters;
2. The “MB” designation must be separated by a space from the “WAY” name ;
3. The brand name must be written in the same font as the remaining text.

A.2.8 Application Tips

The application of the MB WAY brand mark, regardless of the publicizing method, or application purpose, must always comply with the rules defined in this guide.

Whenever the MB WAY service is supplied as a payment option, SIBS FPS suggests that the brand mark is placed on the home page of the information support, in its institutional colors, and guarantee that the brand layout is not smaller than other brands displayed on the same page or support. This best practice ensures a correct use of the brand mark and a better guidance for the User.

A.2.9 Misapplication

It is essential that the brand mark keeps its integrity, regardless of the used support.

The list below presents a set of situations that shall never occur in the context of the brand mark application:

- Alter the proportions and/or the placement of the logotype elements;
- Place the logotype beyond the previewed area;
- Use invalid colors;
- Distort the logotype;
- Use invalid lettering;
- Apply frames or any other decorating elements;
- Apply texts or any other elements that do not respect the minimum protection area;
- Use the signature in a lower size than the defined;
- Use outlines.

A.2.10 Restrictions

Under any circumstance will SIBS FPS give permission to use the MB WAY brand mark, if it is considered that its application might:

- Be associated to imagery or wording that create a risk of damage to the brand ;
- Mislead the consumer with respect to the quality and legitimacy of the advertised product ;
- Create confusion with respect to competing brands, products or entities ;
- Favor or encourage mistrustful, illegal or criminal activities.

A.2.11 Non-compliance

Failure to comply with the rules defined in this guide may cause the brand mark proprietary, SIBS SGPS, to apply sanctioning measures. Any conflicts arising from non-compliance of the MB WAY brand mark image shall be dealt with in the scope of the rules set in this guide and any other related documents, and in compliance with the Portuguese law in force.