TPV-Virtual

Guía de Integración BIZUM comercios SIS

Version: 2.1
Date: 05/11/2025
Reference:RS.TE.CEL.MAN.0033

Redsys, Servicios de Procesamiento, S.L. – c/ Francisco Sancha, 12 – 28034 Madrid (Spain)
www.redsys.es

Version	Date	Affects	Brief description of the change
1.0	10/01/2019	ALL	Initial version
1.1	10/10/2019	Point 4	Point 4 is added.
1.2	14/11/2019	Point 3.1	Data for the testing environment is added.
1.2.1	20/05/2020	Double Flow	Reference generation removed.
1.3	08/01/2021	Point 3	Specific parameterisation is included to report the client's mobile phone number.
1.4	23/04/2021	Point 3	Reference to the type of host-to-host connection in dual-flow and return operations is included.
1.5	28/09/2021	Points 5 and 6	REST integration specifications are included.
1.6	14/06/2022	Point 7	Point 7 included with new REST functionalities.
1.7	01/09/2022	Point 8	Point 8 is included with an introduction to PSD2 exemptions, including MIT.
1.8	18/01/2023	Point 8	MIT input parameters are updated and information on TRA and LWV exemptions is added.
1.9	31/01/2023	Points 5.1 and 8.1	Future specifications for RTP queries are added. Example of complete MIT operational flow added.
1.10	03/05/2023	Points 5.1 and 6	Definitive version of the updated RTP query defined to verify exemptions and validity of TIDs.
1.11	29/08/2024	Point 7.1	Update description and examples.
2.0	05/11/2025	ALL	References to Rest manual added. Update of merchant signature version.

2.1	05/11/2025	Points 5.2, 5.3, 6 and 10	Authentication backup in REST integration.

TABLE OF CONTENTS

1. Introduction

This guide aims to describe the modifications that a merchant must make to offer the BIZUM payment solution.

The BIZUM button must be offered to cardholders on the merchant's website. This means that the merchant must update their website to allow cardholders who are shopping at the merchant to see this option alongside the other payment options offered.

2. Operation

2.1 Description

The merchant must make changes so that BIZUM is displayed as another payment option on its website during the final purchase process (checkout).

If the merchant uses one of the e-commerce solutions on the market (Prestashop, Zen Cart, WooCommerce, OsCommerce, etc.), this must include a module that allows this connection.

An illustrative example of integration with Bizum is shown in the following image:

2.2 Procedure

The steps required for a merchant to offer BIZUM are these three:

1. The merchant must have the option to pay with BIZUM activated by its bank.
2. The merchant must include a button on its website that identifies payment via BIZUM.
3. Make a call to the Virtual POS having a specific format and being very easy to implement.

3. Technical specifications – Integration Redirection

To offer BIZUM payments through the virtual POS terminal, a few minor modifications must be made to the merchant's server.

3.1 Inclusion of the BIZUM button on the merchant's website

In addition to the non-card payment methods already in use, a BIZUM payment button must be included via the virtual POS terminal alongside the other options.

Either of these two images can be used:

These buttons are included in a *zip file* that can be downloaded from the same documentation section where this document is located. The *zip file* is named: *Botones Bizum.7z (Bizum bottons.7z).*

3.2 Payment operation

When the client clicks on the payment button, a similar request to a standard request made to the Virtual POS *(request via a redirection connection)* must be generated. This request must include the following field:

Ds_Merchant_PayMethods whose value will be the lowercase letter "z" for payment with *BIZUM*.

The Bizum operation has two different behaviours depending on the type of operation to be conducted:

a. **Standard Bizum payment:** In this Bizum operation, the Virtual POS authenticates the client and requests authorisation in a single operation.

   To use this method, the merchant must specify the following field in the request to the Virtual POS:

Ds_Merchant_OperationType = 0

NOTE: The format of the "**Ds_Merchant_Order"** parameter, the order number sent in the request, must have this format.

Campo	Descripción	Tipo	Formato
Ds_Merchant_Order	Nº De pedido	Alfanumérico	^[a-zA-Z0-9]{4,12}$ Ej. 123456Ebf

b. **Double-flow Bizum payment:** The double-flow reference indicates that the merchant has the option of performing the authentication and authorisation of the client's operation separately. The difference with *standard Bizum payment* lies in the type of operation (Ds_Merchant_OperationType) sent in the request to the Virtual POS *(request via a Redirection connection).*

NOTE: Option NOT available for e-commerce solutions such as (Prestashop, Zen Cart, WooCommerce, OsCommerce, etc.).

Once authentication has been completed, the merchant has a maximum of 30 days to launch the operation authorisation. Once this time has elapsed, the authorisation will no longer be valid. (request via a REST connection)

The following operation types must be used for these two phases:

Phase 1. Authentication (it must be performed with a connection to the Virtual POS via redirection)

To perform authentication, the following field must be included:

```
Ds_Merchant_OperationType = 7
```

Phase 2. Authorisation (must be performed with a connection to the Virtual POS via REST)

To perform authorisation, the following field must be included:

Ds_Merchant_OperationType = 8

Optionally, the merchant may provide the client's mobile phone number so that it automatically appears in the Bizum environment. To do so, the mobile phone number must be entered in the following field of the form:

Ds_Merchant_Bizum_MobileNumber

whose value will be the mobile phone number

including the prefix (e.g. +34700000000)

NOTE: The technical details for making a Redirection or REST request can be found in the following guides: "TPV-Virtual Integration Manual - Redirection.pdf" and "TPV-Virtual Integration Manual -REST.pdf".

3.3 Refund operation

A Bizum operation can be refunded by sending a REST request with the following field:

Ds_Merchant_OperationType = 3

The authorisation operation can also be returned through the Virtual POS Administration Portal. See point 4.2

4. Test environment – Redirection integration

There is a test environment that allows you to perform the necessary tests to verify the correct operation of the system before implementing it in the real environment.

The access URL for the test environment is:

https://sis-t.redsys.es:25443/sis/realizarPago

The URL for accessing the Administration Portal in the test environment is:

https://sis-t.redsys.es:25443/canales

Test data:

BIZUM user	Test case
700000000	Successful purchase
ko@ko.ko	Purchase rejected (Bizum key authentication error)

NOTE: Payments made in the test environment will not be valid for accounting purposes.

5. Technical specifications – REST integration

To conduct BIZUM operations via REST, the client (cardholder) must be able to accept Request To Pay, which consists of sending a notification to the client's mobile phone to accept the payment from there.

To find out if the client has this operation available, an RTP query service has been enabled, where the merchant can check the operations allowed by the client. Through this service, the merchant can find out if:

- The client has the BIZUM service and Request To Pay. In this case, BIZUM payment can be requested via REST as explained below, or via redirection, as explained previously.
- The client has the BIZUM service but not Request To Pay. In this case, BIZUM payment can only be requested via redirection.
- The client does not have the BIZUM service. In this case, BIZUM payment cannot be requested from the client.
- Verify the status of a client's TID (Depends on the input parameters) .
- Verify the exemptions supported by a client (depending on the input parameters).

5.1 RTP query request

5.1.1 Input parameters

The parameters required to use the query service are as follows:

Field	Description	Type	Format
DS_MERCHANT_ORDER	Order number	Alphanumeric	^[a-zA-Z0-9]{4,12}$
DS_MERCHANT_MERCHANTCODE	Merchant number	Numeric	^[\d]{1,9}$
DS_MERCHANT_TERMINAL	Terminal number	Numeric	^[\d]{1,3}$
DS_MERCHANT_BIZUM_MOBILENUMBER	Client's phone number with prefix	String	^[\+]?[\d]{5,15}$ E.g.: + 34700000000
DS_MERCHANT_COF_TXNID	Operation ID to check its validity. **OPTIONAL**	String	^[a-zA-Z0-9]{1,15}$
DS_MERCHANT_EXCEPT_SCA	Parameter indicating the query of exemptions	String	^Y$

	permitted by the client **OPTIONAL**		

As with any type of REST integration, the merchant must send a POST request with the following parameters:

- **Ds_MerchantParameters:** Payment request data encoded in Base64.
- **Ds_SignatureVersion**: Signature algorithm version.
- **Ds_Signature**: Signature generated with the payment data.

The URL where the service is displayed is as follows, depending on the environment:

- **TEST environment**: *https://sis-t.redsys.es:25443/sis/rest/RTP/checkRtpUsuario*
- **PRODUCTION environment**: *https://sis.redsys.es/sis/rest/RTP/checkRtpUsuario*

5.1.2 Output parameters

In this case, the most important fields will be those beginning with **Ds_Rtp**:

- **Ds_RtpStatus**: Indicates whether the client can make a Request to Pay. Its values are 'OK' if possible and 'KO' if not possible. If 'KO' is returned, the **'Ds_RtpResponse'** field must be consulted to verify whether the response received is due to an error or not. For example, if the user has Bizum but not Request to Pay, the **'Ds_RtpResponse'** will come with the code **'BIZ00000'**, whereas if the user does not have Bizum, it will contain a different error code.
- **Ds_RtpResponse**: Request to Pay status query request code. If the request is correct, **'BIZ00000'** will be returned. Otherwise, another error code will be returned following the regular expression **'^BIZ[\d]{5}$'**.
- **Ds_RtpDescription**: Description of the Request to Pay status query request. If the 'Ds_RtpResponse' field has a value other than **'BIZ00000'**, this field will indicate a description of the reason the request was not successful.

The possible scenarios that may arise are as follows:

- **Client with RequestToPay**: In this case, the call will return **Ds_RtpStatus=OK**, **Ds_RtpResponse=BIZ00000**, **Ds_RtpDescription=Operacion realizada correctamente**
- **Client without RequestToPay but with Bizum**: In this case, the call will return **Ds_RtpStatus=KO**, **Ds_RtpResponse=BIZ00000**, **Ds_RtpDescription=Operacion realizada correctamente**
- **Client without RequestToPay neither Bizum**: In this case, the call will return **Ds_RtpStatus=KO**, **Ds_RtpResponse=BIZ00009**, **Ds_RtpDescription=Ordenante no encontrado**
- **Some sort of error**: If an error occurs, the call will return **Ds_RtpStatus=KO**, **Ds_RtpResponse=BIZXXXXX** (Where XXXXX is the identifier of the error in question) and

Ds_RtpDescription=YYYYY (YYYYY being the detailed description of the error that occurred)

If the parameter "DS_MERCHANT_COF_TXNID" or "DS_MERCHANT_EXCEP_SCA" has been reported, the query will return additional information.

If "DS_MERCHANT_COF_TXNID" is reported, its validity will be checked, and the same field will be returned with "OK" or "KO" depending on its validity.

If "DS_MERCHANT_EXCEP_SCA" is reported, the exemptions supported by the client will be checked and the "DS_EXCEP_SCA" field will be returned with the supported exemptions, as is the case with cards to check the exemptions allowed by the merchant (Section 9.1 of the REST integration guide).

5.1.3 Example

This allows us to generate the following example:

- Payment data (Ds_MerchantParameters before encoding in base64):

```
{
    "DS_MERCHANT_ORDER": "1612280107",
    "DS_MERCHANT_MERCHANTCODE": "999008881",
    "DS_MERCHANT_TERMINAL": "871",
    "DS_MERCHANT_CURRENCY": "978",
    "DS_MERCHANT_AMOUNT": "145",
    "DS_MERCHANT_BIZUM_MOBILENUMBER": "+34700000000"
}
```

This will produce the following response:

- Decoded response data:

```
{
    "Ds_RtpStatus": "OK",
    "Ds_RtpResponse": "BIZ00000",
    "Ds_RtpDescription": "Operation performed correctly",
    "Ds_MerchantCode": "999008881",
    "Ds_Terminal": "871",
    "Ds_Order": "1612280505",
    "Ds_Currency": "978",
    "Ds_Amount": "145"
}
```

5.2 Start RTP payment request via REST

IMPORTANT! If you decide to integrate Bizum via REST, it is MANDATORY that you support authentication backup. More information in point 10 of the document.

The Bizum integration flow via REST follows the following scheme:

1. The merchant requests the initiation of Request To Pay.
2. An attempt is made to initiate Request To Pay.
    a. **The client has RTP active and communication with the entity has been successful**: We respond with the success of the RTP start attempt by sending the field **Ds_RtpResponse=BIZ00000** and **Ds_RtpDescription=Operation performed successfully**
    b. **Any other situation**: We respond with the RTP initiation attempt error by sending the field **Ds_RtpResponse=BIZXXXXX** (where XXXXX is the identifier of the error in question) and **Ds_RtpDescription=YYYYY** (where YYYYY is the detailed description of the error that occurred).
3. If the RTP start was successful, when the request is confirmed/denied, the system will receive a notification with the result and depending on this and the type of operation to be processed, one action or another will be taken.
    a. Confirmation:
        i. Operation type 0: The operation will be authorised, and the status of the operation will be updated with the result of the authorisation.
        ii. Operation type 7: The operation status will be updated to authenticated and pending authorisation (double-flow Bizum payment).
    b. Refusal:
        i. The status of the operation will be updated as refused.
4. Finally, the merchant will be notified of the result of the operation.

5.2.1 Input parameters

In addition to the basic operation data, the following data is required for this integration:

Field	Description	Type	Format
DS_MERCHANT_OPERATIONTYPE	Operation type	Numeric	^[\d]{1}$
DS_MERCHANT_PAYMENT_METHODS	Payment method	String	^z$
DS_MERCHANT_BIZUM_MOBILENUMBER	Client's phone number with prefix	String	^[\+]?[\d]{5,15}$ E.g.: + 34700000000
DS_MERCHANT_MERCHANTURL	Notification URL	String	URL

The following parameters have specific features:

- **DS_MERCHANT_OPERATIONTYPE**: Only the following values can be specified:
  - **0**: In this case, when correctly notifying the start of Request To Pay, the operation will be authorised directly.
  - **7**: In this case, when the start of Request To Pay is correctly notified, the operation will remain authenticated but pending authorisation, as seen in section 3 of the guide when explaining integration via double flow.
- **DS_MERCHANT_MERCHANTURL**: This parameter is optional if the merchant has configured the Notification URL field in the administration portal. If it is not configured at the terminal level, it must be specified in each call.

As with any type of REST integration, the merchant must send a POST request with the following parameters:

- **Ds_MerchantParameters:** Payment request data encoded in Base64.
- **Ds_SignatureVersion**: Signature algorithm version.
- **Ds_Signature**: Signature generated with the payment data.

The URL where the service is displayed is as follows, depending on the environment:

- **TEST environment**: *https://sis-t.redsys.es:25443/sis/rest/RTP/trataPeticionREST*
- **PRODUCTION environment**: *https://sis.redsys.es/sis/rest/RTP/trataPeticionREST*

5.2.2 Output parameters

In this case, we will differentiate between the response to the RTP initiation request, and the notification sent subsequently when we know the status of the operation if the initiation has been successful.

For the response to the request, the most important parameters are as follows:

Field	Description	Type	Format
Ds_Response	Response to the request	Numeric	^[\d]{4}$
Ds_RtpResponse	RTP start response	String	^BIZ[\d]{5}$
Ds_RtpDescription	RTP start description	String	^.{1,255}$

Regarding the notification response, it will be a normal notification as would be received in a redirection integration.

5.2.3 Example

This allows us to generate the following example:

- Payment data (Ds_MerchantParameters before encoding in base64):

```
{
    "DS_MERCHANT_ORDER": "1614092745",
    "DS_MERCHANT_MERCHANTCODE": "999008881",
    "DS_MERCHANT_TERMINAL": "871",
    "DS_MERCHANT_CURRENCY": "978",
    "DS_MERCHANT_AMOUNT": "145",
    "DS_MERCHANT_OPERATIONTYPE": "0",
    "DS_MERCHANT_PAYMENTMETHODS": "z",
    "DS_MERCHANT_MERCHANTURL": "https:\/\/sis-d.redsys.es\/sis-simulador-
web\/notificacion\/reciboNotif.jsp",
    "DS_MERCHANT_BIZUM_MOBILENUMBER": "+34700000000"
}
```


This will produce the following response:

- Decoded data received:


```
{
    "Ds_Amount": "145",
    "Ds_Currency": "978",
    "Ds_Order": "1614092745",
    "Ds_MerchantCode": "999008881",
    "Ds_Terminal": "871",
    "Ds_Response": "9998",
    "Ds_AuthorisationCode": "",
    "Ds_OperationType": "0",
    "Ds_SecurePayment": "0",
    "Ds_Language": "1",
    "Ds_MerchantData": "",
    "Ds_ProcessedPayMethod": "68",
    "Ds_RtpResponse": "BIZ00000",
     "Ds_RtpDescription": "Operacion realizada correctamente"
}
```


When the RTP is complete, the following notification will be sent:

- Decoded data received:


```
{
    "Ds_Date": "23/02/2021",
    "Ds_Hour": "16:07",
    "Ds_SecurePayment": "1",
    "Ds_Amount": "145",
    "Ds_Currency": "978",
    "Ds_Order": "1614092745",
    "Ds_MerchantCode": "999008881",
    "Ds_Terminal": "871",
    "Ds_Response": "0000",
    "Ds_MerchantData": "",
    "Ds_OperationType": "0",
```

```
    "Ds_ConsumerLanguage": "1",
    "Ds_AuthorisationCode": "123456",
    "Ds_ProcessedPayMethod": "68"
}
```

5.3 Associated error codes

Code	DESCRIPTION
BIZ00000	Operation performed correctly.
BIZ00001	Required input parameter not completed.
BIZ00002	The format of a parameter is incorrect.
BIZ00003	The item was not found.
BIZ00005	Internal system error.
BIZ00006	3DES or MAC security error X9.19
BIZ00007	Operation not permitted.
BIZ00008	Beneficiary not found.
BIZ00009	Sender not found.
BIZ00202	Functionality not yet implemented.
BIZ00213	Authentication error in the request received. Security sequence failure.
BIZ00233	Authentication error. The operation can be backed up.
BIZ00224	The entity's response to RTP authentication is KO.
BIZ00225	Request to pay authentication has not been successfully completed.

6. Testing – REST integration

To assess the different scenarios, you can use the test telephone number +34700000000, conducting operations with different amounts.

Amount	Client with Bizum	Client with RTP	Operation result	Description in RTP query	Description in REST payment
Less than €5	OK	OK	OK	The user's telephone has RTP active. In this range, the client has exemptions and all valid TIDs.	In the initial request, we received a response that it went OK. When reviewing the operation, we will see that it is authenticated.
Between €5 and €10	OK	OK	KO	The user's telephone has RTP active. In this range, the client has no exemptions and all TIDs are invalid.	In the initial request, we receive a response that it went OK. When reviewing the operation, we will see that it is denied.
Between €10 and €15	OK	KO	-	The user's telephone does not have RTP active.	In the initial request, we received a response that it has gone KO. The error reason is BIZ00202.
Over €15	KO	KO	-	The user's phone number does not have BIZUM activated	In the initial request, we received a response that it has gone KO. The error reason is BIZ00009.
Between €75 and €80	OK	OK	OK		In the initial request, we received a response stating that the operation can be backed up. Upon completion of the flow, the operation will remain authorised.
Between €80 and €85	OK	OK	KO		In the initial request, we received a response stating that the operation can be backed up. At the end of the flow, the operation will be denied.

7. Other REST functionalities

7.1 Return without original

This type of operation will allow the merchant to make the refund without the need for an original operation, such as a rebate, a bonus payment, etc., or in special situations such as when the deadline for a refund has passed.

7.1.1 Input parameters

In addition to the basic operation data, the following data is required for this integration:

Field	Description	Type	Format
DS_MERCHANT_OPERATIONTYPE	Operation type	Numeric	^34$
DS_MERCHANT_PAYMETHODS	Payment method	String	^z$
DS_MERCHANT_BIZUM_MOBILENUMBER	Client's phone number with prefix	String	^[\+]?[\d]{5,15}$ E.g.: + 34700000000

As with any type of REST integration, the merchant must send a POST request with the following parameters:

- **Ds_MerchantParameters:** Payment request data encoded in Base64.
- **Ds_SignatureVersion**: Version of the signature algorithm.
- **Ds_Signature**: Signature generated with the payment data.

The URL where the service is displayed is as follows, depending on the environment:

- **TEST environment**: *https://sis-t.redsys.es:25443/sis/rest/RTP/trataPeticionREST*
- **PRODUCTION environment**: *https://sis.redsys.es/sis/rest/RTP/trataPeticionREST*

7.1.2 Output parameters

For the response to the request, the standard response from a REST request to the Virtual POS will be obtained, where the most important parameter is the following:

Field	Description	Type	Format
Ds_Response	Response to the request. The value "0900" indicates that the operation has been authorised.	Numeric	^[\d]{4}$

7.1.3 Example

This allows us to generate the following example:

- Payment details (Ds_MerchantParameters before encoding in base64):


```
{
    "DS_MERCHANT_ORDER":1724928574,
    "DS_MERCHANT_MERCHANTCODE":"999008881",
    "DS_MERCHANT_TERMINAL":"871",
    "DS_MERCHANT_CURRENCY":"978",
    "DS_MERCHANT_OPERATIONTYPE":"34",
    "DS_MERCHANT_AMOUNT":"145",
    "DS_MERCHANT_PAYMENT_METHODS":"z",
    "DS_MERCHANT_BIZUM_MOBILENUMBER":"34700000000"
}
```


This will produce the following response:

- Decoded data received:


```
{
    "Ds_Amount":"145",
    "Ds_Currency":"978",
    "Ds_Order":"1724928574",
    "Ds_MerchantCode":"999008881",
    "Ds_Terminal":"871",
    "Ds_Response":"0900",
    "Ds_AuthorisationCode":"",
    "Ds_OperationType":"Y",
    "Ds_SecurePayment":"1",
    "Ds_Language":"1",
    "Ds_MerchantData":"",
    "Ds_Bizum_IdOper":"66569849-91c9-41b2-97b2-83cb96718bf",
    "Ds_ProcessedPayMethod":"68",
    "Ds_Control_1724928579058":"1724928579058",
    "Ds_RtpResponse":"BIZ00000",
    "Ds_RtpDescription":"Operation completed successfully"
}
```

8. PSD2 exemptions

At this point, the use of the several types of exemptions in Bizum payments will be explained in different sections.

8.1 MIT operation

To make MIT payments, it will be necessary to subscribe the user to the product and store the resulting TID (operation ID) for later use.

The rules published by issuers require that all operations (Obtain TID & MIT) that are COF be identified, as well as the use (COF type) being made of the credentials. The aim of this new regulation is to reduce the number of refusals to issuers in this type of operation.

8.1.1 TID generation

First, an initial operation (which will always be authenticated) must be conducted to generate the TID to be used in future MIT operations. To do this, the following parameter must be included in the operation (regardless of whether it is conducted via REST or redirection):

Field	Description	Type	Format
DS_MERCHANT_COF_TYPE	COF operation type. Values: "I": Instalments "R": Recurring “H”: Reauthorisation “E”: Resubmission "D": Delayed “M”: Incremental “N”: No Show “C”: Other	String	^(I|R|H|E|D|M|N|C)$
DS_MERCHANT_COF_INI	First use indicator for COF	String	^S$

The response to this operation will include a new field indicating the TID generated for use in future MIT operations.

Field	Description	Type	Format
DS_MERCHANT_COF_TXNID	COF operation identifier. Optional. The merchant will receive this identifier as a response in the initial credential storage (CIT) operations. This value must be stored so that it can be sent in subsequent COF authorisations, linking all subsequent operations to their corresponding original. Optionally, it is also possible to receive this value in the response to a subsequent MIT operation. In this case, if the value received is different from the one stored by the merchant, the merchant must store this new value for use in the following linked MITs.	String	^[a-zA-Z0-9]{1,15}$

8.1.2 Example

- Payment details (Ds_MerchantParameters before encoding in base64):


```
{
    "DS_MERCHANT_ORDER": "1675155813",
    "DS_MERCHANT_MERCHANTCODE": "999008881",
    "DS_MERCHANT_TERMINAL": "871",
    "DS_MERCHANT_CURRENCY": "978",
    "DS_MERCHANT_AMOUNT": "100",
    "DS_MERCHANT_PRODUCTDESCRIPTION": "Test",
    "DS_MERCHANT_OPERATIONTYPE": "0",
    "DS_MERCHANT_PAYMENT_METHODS": "z",
    "DS_MERCHANT_BIZUM_MOBILENUMBER": "+34700000000",
    "DS_MERCHANT_COF_TYPE": "I",
    "DS_MERCHANT_COF_INI": "S",
    "DS_MERCHANT_MERCHANTURL": "https://sis-d.redsys.es/sis-simulador-web/notificacion/reciboNotif.jsp
}
```


This will produce the following response:

- Decoded data received:


```
{
    "Ds_Amount": "100",
    "Ds_Currency": "978",
    "Ds_Order": "1675155813",
    "Ds_MerchantCode": "999008881",
    "Ds_Terminal": "871",
    "Ds_Response": "9998",
    "Ds_AuthorisationCode": "",
    "Ds_OperationType": "0",
    "Ds_SecurePayment": "0",
    "Ds_Language": "1",
    "Ds_MerchantData": "",
    "Ds_ProcessedPayMethod": "68",
    "Ds_Control_1675155876933": "1675155876933",
    "Ds_RtpResponse": "BIZ00000",
    "Ds_RtpDescription": "SIS-T Operation performed correctly"
}
```


When the RTP is complete, the following notification will be sent:

- Decoded data received:


```
{
    "Ds_Date": "31%2F01%2F2023",
    "Ds_Hour": "10:04",
```

```
    "Ds_SecurePayment": "1",
    "Ds_Amount": "100",
    "Ds_Currency": "978",
    "Ds_Order": "1675155813",
    "Ds_MerchantCode": "999008881",
    "Ds_Terminal": "871",
    "Ds_Response": "0000",
    "Ds_MerchantData": "",
    "Ds_OperationType": "0",
    "Ds_ConsumerLanguage": "1",
    "Ds_AuthorisationCode": "000000",
    "Ds_Bizum_TruncatedAccount": "XXXX4000XXXXXXXXXXX4000",
    "Ds_Bizum_IdOper": "938275551143520623001217181919669 11",
    "Ds_Merchant_Cof_Txnid": "121718191966911",
    "Ds_ProcessedPayMethod": "68",
    "Ds_Control_1675155882558": "1675155882558"
}
```


8.1.3 MIT Operations

Once the TID has been generated, the merchant can initiate operations without requiring the client to be present.

For these operations, it is not necessary to provide the buyer's phone number, as the TID is sufficient. The following parameters must be provided:

Field	Description	Type	Format
DS_MERCHANT_COF_TXNID	Operation ID	String	^[a-zA-Z0-9]{1,15}$
DS_MERCHANT_COF_TYPE	COF operation type. Values: “I”: Instalments “R”: Recurring “H”: Reauthorisation “E”: Resubmission "D": Delayed “M”: Incremental “N”: No Show “C”: Other	String	^(I|R|H|E|D|M|N|C)$

DS_MERCHANT_COF_INI	First use indicator for COF	String	^N$
DS_MERCHANT_EXCEP_SCA	Exemption type (MIT)	String	^MIT$
DS_MERCHANT_DIRECTPAYMENT	Non-secure payment indicator	String	^true$

NOTE: No validation will be performed on the COF type used between the operation to obtain the TID and the MIT operation. If a different COF type is sent in future MIT payments, the operation may be rejected. The cancellation is made by the issuer. Each issuer may have its own criteria and perform its own validations. In addition, these may evolve over time and start without validation and then begin to validate. This already occurs in card payments.

8.1.3.1 Example

Based on the example of TID generation,

- Payment data (Ds_MerchantParameters before encoding in base64):

```
{
    "DS_MERCHANT_ORDER": "1675156044",
    "DS_MERCHANT_MERCHANTCODE": "999008881",
    "DS_MERCHANT_TERMINAL": "871",
    "DS_MERCHANT_CURRENCY": "978",
    "DS_MERCHANT_AMOUNT": "100",
    "DS_MERCHANT_PRODUCTDESCRIPTION": "Test",
    "DS_MERCHANT_OPERATIONTYPE": "0",
    "DS_MERCHANT_PAYMENT_METHODS": "z",
    "DS_MERCHANT_COF_TYPE": "I",
    "DS_MERCHANT_COF_INI": "N",
    "DS_MERCHANT_COF_TXNID": "121718191966911",
    "DS_MERCHANT_DIRECTPAYMENT": "true",
    "DS_MERCHANT_EXCEP_SCA": "MIT",
    "DS_MERCHANT_MERCHANTURL": "https://sis-d.redsys.es/sis-simulador-
web/notificacion/reciboNotif.jsp
}
```

This will produce the following response:

- Decoded data received:

```
{
    "Ds_Amount": "100",
    "Ds_Currency": "978",
    "Ds_Order": "1675156044",
    "Ds_MerchantCode": "999008881",
```

```
    "Ds_Terminal": "871",
    "Ds_Response": "0000",
    "Ds_AuthorisationCode": "",
    "Ds_OperationType": "0",
    "Ds_SecurePayment": "0",
    "Ds_Language": "1",
    "Ds_MerchantData": "",
    "Ds_Merchant_Cof_Txnid": "121718191966911",
    "Ds_ProcessedPayMethod": "68",
    "Ds_Control_1675156081359": "1675156081359"
}
```


8.2 Other exemptions (TRA and LWV)

To enable a payment in which the client does not perform the strong authentication process, the use of exemptions is enabled, which may be proposed by the merchant.

The following exemptions are permitted:

- **TRA** (Operation Risk Analysis): this exemption is based on an operation risk analysis by the acquirer/merchant.
- **LWV** (Low value operation): exemption for low amounts (up to €30, with a maximum of 5 operations or €100 accumulated per client; these counters are controlled at the issuing entity level).

8.2.1 First exemption request

Upon the first exemption request for a client at a merchant (1:1 ratio), the proposal will not be considered, and a standard payment request will be sent.

If the operation is completed successfully and the operation is conducted, the information will be stored for subsequent checks.

To request the use of exemptions, the merchant must include the following parameter in the payment request:

Field	Description	Type	Format
DS_MERCHANT_EXCEP_SCA	Exemption type	String	^(TRA|LWV)$

8.2.2 Subsequent requests

Once it has been identified that the user has made a non-exempt purchase at the merchant making the request, the relevant merchant validations will be conducted and once passed, the exemption proposal submitted by the merchant will be forwarded to the client's financial entity.

The financial entity must complete the authentication process, being free to accept or reject the exemption, and will send the completion of authentication through the usual channel, where it will be transferred if the proposed exemption has been applied or not.

To request the use of exemptions, the merchant must include the following parameter in the payment request:

Field	Description	Type	Format
DS_MERCHANT_EXCEP_SCA	Exemption type	String	^(TRA|LWV)$

Depending on whether the exemption is accepted or not, the following parameter will be transferred to the merchant in the response:

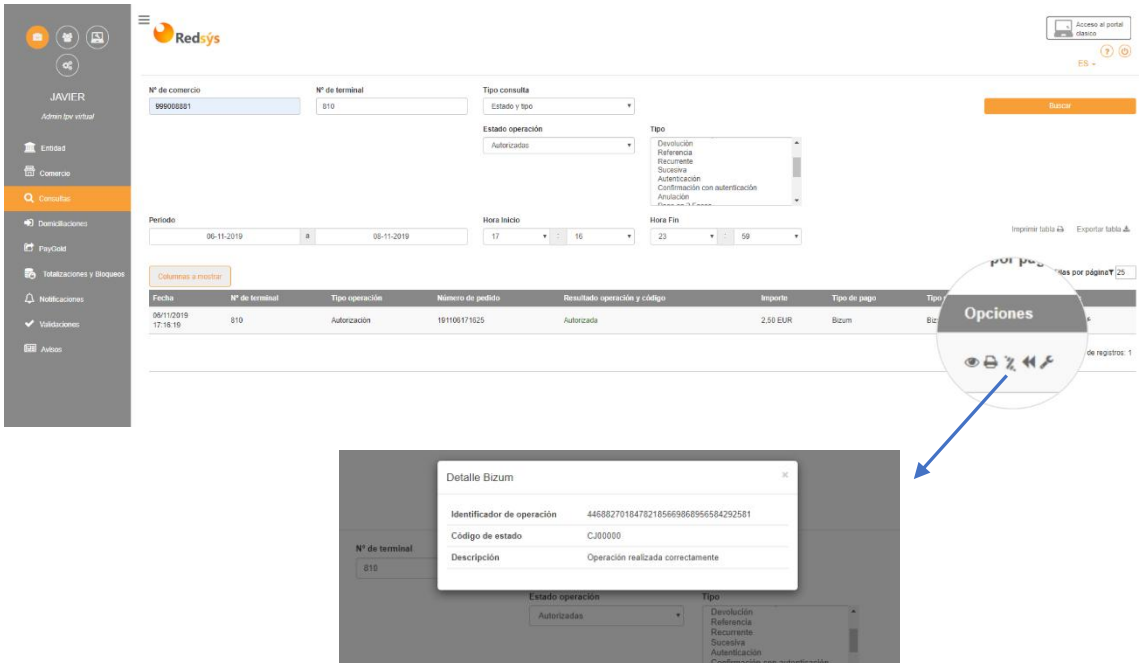
Field	Description	Type	Format
Ds_SecurePayment	Indicates responsibility for the operation: - 1: Secure operation, issuer responsibility - 0: Non-secure operation, merchant liability (Exemption accepted)	String	^(0|1)$

9. Management of Bizum operations from the Virtual POS Administration Portal

The Virtual POS Administration Portal allows you to query and refund Bizum operations.

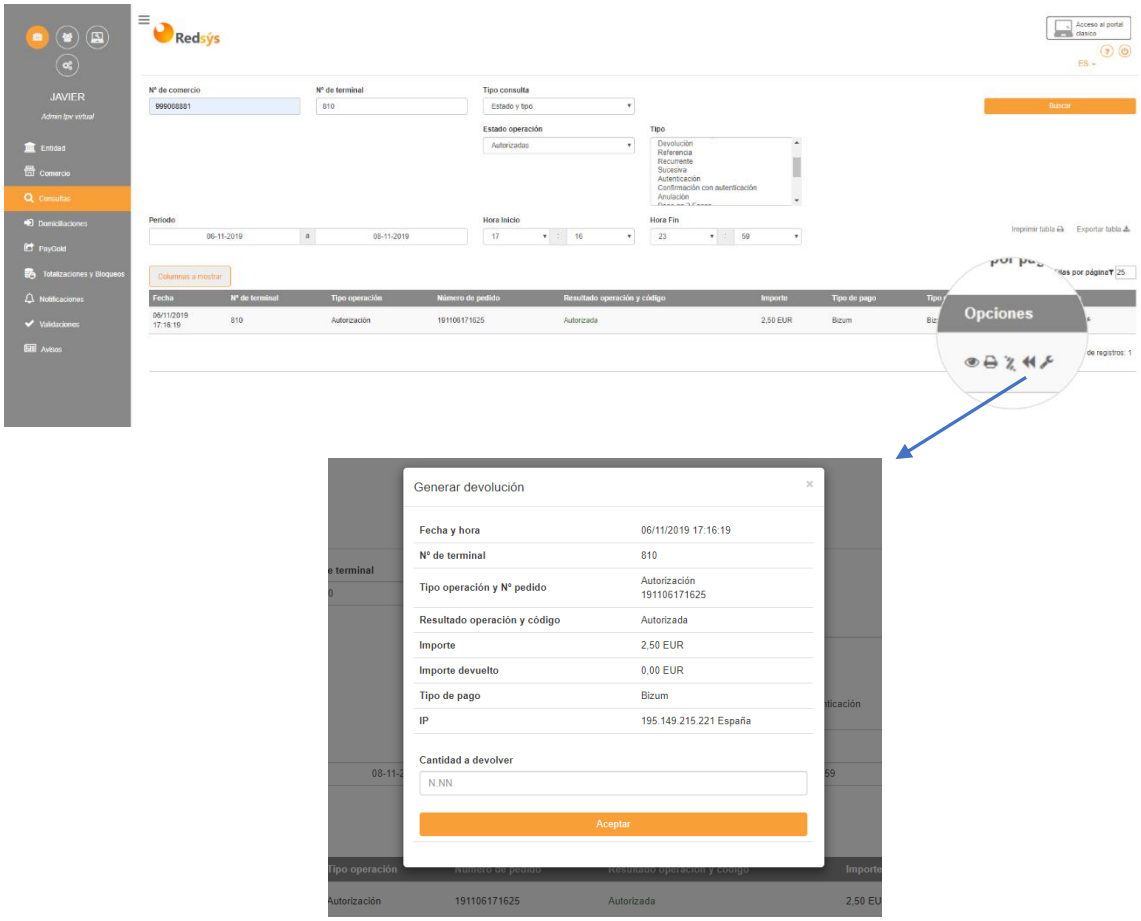
9.1 Bizum Operation Query

From the operation query option in the Virtual POS Administration Portal, you can obtain details of Bizum operations. In the "Options" section, clicking on the Bizum icon will display the operation details, as shown in the following image:

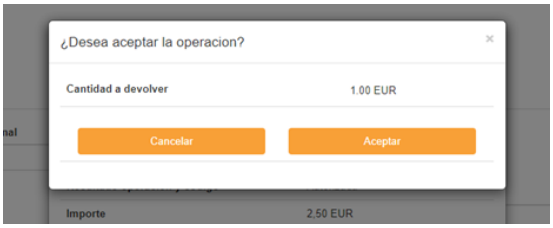


9.2 Bizum Operation Returns

From the operation query option in the Virtual POS Administration Portal, you can return Bizum operations, provided that the Portal user has permission to make refunds. In the "Options" section, clicking on the "Double Arrow" icon opens a pop-up window where you can specify the refund details, as shown in the following image:



Once you have specified the amount to be refunded, an operation confirmation box will appear, as shown in the following image:



10. Backup

At this point, we will explain the different steps for backing up an operation with Bizum payment.

For a Bizum payment operation to be backed up, we must obtain a response like this:

- Data received:


```
{
    "Ds_SignatureVersion": "HMAC_SHA512_V2",
    "Ds_MerchantParameters":
"eyJEc19BbW91bnQiOiI3NjAwIiwiRHNfQ3VycmVuY3kiOiI5NzgiLCJEc19PcmRlciI6IjE3NjU
3OTYwMzciLCJEc19NZXJjaGFudENvZGUiOiI5OTkwMDg4ODEiLCJEc19UZXJtaW5hbCI6IjgxMCI
sIkRzX1Jlc3BvbnNlIjoiOTk5OCIsIkRzX0F1dGhvcmlzYXRpb25Db2RlIjoiIiwiRHNfVHJhbnN
hY3Rpb25UeXBlIjoiMCIsIkRzX0xhbmd1YWdlIjoiMSIsIkRzX01lcmNoYW50RGF0YSI6IiIsIkR
zX0JpenVtX0lkT3BlciI6IjE4NDQ2ODQ4MjcyNjI3ODM4NjMyMjg5MzYzMTM2MDg3NTg3IiwiRHN
fUHJvY2Vzc2VkUGF5TWV0aG9kIjoiNjgiLCJEc19Db250cm9sXzE3NjU3OTYwNDE3MDgiOiIxNzY
1Nzk2MDQxNzA4IiwiRHNfUnRwUmVzcG9uc2UiOiJCSVowMDIzMyIsIkRzX1J0cERlc2NyaXB0aW9
uIjoiRXJyb3IgZW4gbGEgYXV0ZW50aWNhY2lvbi4gTGEgb3BlcmFjaW9uIHB1ZWRlIHJlc3BhbGR
hcnNlIiwiRHNfUnRwQmFja3VwSHRtbCI6IlBDRkVUME5VV1ZCRklFaFVUVXcrUEdoMGJXd2diR0Z
1WnowaUpIdHNZVzVuZFdGblpYMGlQanhvWldGa1BqeHRaWFJoSUdOb1lYSnpaWFE5SW5WMFppMDR
JajQ4YldWMFlTQm9kSFJ3TFdWeGRXbDJQU0pZTFZWQkxVTnZiWEJoZEdsaWJHVWlJR052Ym5SbGJ
uUTlJa2xGUFdWa1oyVWlQaUk4YldWMFlTQnVZVzFsUFNKMmFXVjNjRzl5ZENJZ1kyOXVkR1Z1ZEQ
waWQybGtkR2c5WkdWMmFXTmxMWGRwWkhSb0xDQnBibWwwYVdGc0xYTmpZV3hsUFRFaVBqeHRaWFJ
oSUdoMGRIQXRaWEYxYVhZOUlrTmhZMmhsTFVOdmJuUnliMndpWTI5dWRHVnVkRDBpYm04dFkyRmp
hR1VzSUc1dkxYTjBiM0psTENCdGRYTjBMWEpsZG1Gc2FXUmhkR1VzSUcxaGVDMWhaMlU5TUNJdlB
qeHRaWFJoSUdoMGRIQXRaWEYxYVhZOUlsQnlZV2R0WVNJZ1kyOXVkR1Z1ZEQwaWJtOHRZMkZqYUd
VaUx6NDhiV1YwWVNCb2RIUndMV1Z4ZFdsMlBTSkZlSEJwY21WeklpQmpiMjUwWlc1MFBTSXdJaUF
2UGp4c2FXNXJJSEpsYkQwaWFXTnZiaUlnYUhKbFpqMGlabUYyYVdOdmJpNXdibWNpUGp4MGFYUnN
aVDVRWVdkdklFSnBlblZ0UEM5MGFYUnNaVDQ4YzJOeWFYQjBJSE55WXowaWFIUjBjSE02THk5aGF
tRjRMbWR2YjJkc1pXRndhWE11WTI5dEwyRnFZWGd2YkdsaWN5OXFjWFZsY25rdk1TNHhNUzR6TDJ
weGRXVnllUzV0YVc0dWFuTWlQand2YzJOeWFYQjBQanhzYVc1cklHaHlaV1k5SW1oMGRIQnpPaTh
2Y0hCcGFTMXBMb...",
    "Ds_Signature": "199rd39HzwmurheI9IC-
EKS4PLOX11wMqvA9icN5okZgwkJWV2N3xXOY-_Osy3By7DlFEnOyty3RGdAxDA7EtQ"
}
```


- Decoded data:


```
{
    "Ds_Amount": "7600",
    "Ds_Currency": "978",
    "Ds_Order": "1765796037",
    "Ds_MerchantCode": "999008881",
```

```
    "Ds_Terminal": "810",
    "Ds_Response": "9998",
    "Ds_AuthorisationCode": "",
    "Ds_OperationType": "0",
    "Ds_Language": "1",
    "Ds_MerchantData": "",
    "Ds_Bizum_IdOper": "184468482726278386322893631360875 87",
    "Ds_ProcessedPayMethod": "68",
    "Ds_Control_1765796041708": "1765796041708",
    "Ds_RtpResponse": "BIZ00233",
    "Ds_RtpDescription": "Authentication error. The operation can be backed
up",
    "Ds_RtpBackupHtml":
"PCFET0NUWVBFIEhUTUw+PGh0bWwgbGFuZz0iJHtsYW5ndWFnZX0iPjxoZWFkPjxtZXRhIGNoYXJ
zZXQ9InV0Zi04Ij48bWV0YSBodHRwLWVxdWl2PSJYLVVBLUNvbXBhdGlibGUiIGNvbnRlbnQ9Ikl
FPWVkZ2UiPiI8bWV0YSBuYW1lPSJ2aWV3cG9ydCIgY29udGVudD0id2lkdGg9ZGV2aWNlLXdpZHR
oLCBpbml0aWFsLXNjYWxlPTEiPjxtZXRhIGh0dHAtZXF1aXY9IkNhY2hlLUNvbnRyb2wiY29udGV
udD0ibm8tY2FjaGUsIG5vLXN0b3JlLCBtdXN0LXJldmFsaWRhdGUsIG1heC1hZ2U9MCIvPjxtZXR
hIGh0dHAtZXF1aXY9IlByYWdtYSIgY29udGVudD0ibm8tY2FjaGUiLz48bWV0YSBodHRwLWVxdWl
2PSJFeHBpcmVzIiBjb250ZW50PSIwIiAvPjxsaW5rIHJlbD0iaWNvbiIgaHJlZj0iZmF2aWNvbi5
wbmciPjx0aXRsZT5QYWdvIEJpenVtPC90aXRsZT48c2NyaXB0IHNyYz0iaHR0cHM6Ly9hamF4Lmd
vb2dsZWFwaXMuY29tL2FqYXgvbGlicy9qcXVlcnkvMS4xMS4zL2pxdWVyeS5taW4uanMiPjwvc2N
yaXB0PjxsaW5rIGhyZWY9Imh0dHBzOi8vcHBpaS1pLnJlZHN5cy5lcy9jMmVyLWF1dGgvc3RhdGl
jL2Nzcy9ib290c3RyYXAubWluLmNzcyIgcmVsPSJzdHlsZXNoZWV0IiAvPjxsaW5rIGhyZWY9Imh
0dHBzOi8vcHBpaS1pLnJlZHN5cy5lcy9jMmVyLWF1dGgvc3RhdGljL2Nzcy9tYWluLmNzcz92PTF
fMiIgcmVsPSJzdHlsZXNoZWV0IiAvPjxzdHlsZT5AaW1wb3J0IHVybCgnaHR0cHM6Ly9mb250cy5
nb29nbGVhcGlzLmNvbS9jc3MyP2ZhbWlseT1NdWxpc2g6aXRhbCx3Z2h0QDAsMjAwLi4xMDAwOzE
sMjAwLi4xMDAwJmRpc3BsYXk9c3dhcCcpO0BpbXBvcnQgdXJsKCdodHRwczovL2ZvbnRzLmdvb2d
sZWFwaXMuY29tL2NzczI/ZmFtaWx5PUxhdG86aXRhbCx3Z2h0QDAsMTAwOzAsMzAwOzAsNDAwOzA
sNzAwOzAsOTAwOzEsMTAwOzEsMzAwOzEsNDAwOzEsNzAwOzEsOTAwJmRpc3BsYXk9c3dhcCcpO2h
0bWwsIGJvZHksIGRpdiwgc3BhbiwgYXBwbGV0LCBvYmplY3QsIGlmcmFtZSwgaDEsIGgyLCBoMyw
gaDQsIGg1LCBoNiwgcCxibG9ja3F1b3RlLCBwcmUsIGEsIGFiYnIsIGFjcm9ueW0sIGFkZHJlc3M
sIGJpZywgY2l0ZSwgY29kZSwgZGVsLCBkZm4sZW0sIGltZywgaW5zLCBrYmQsIHEsIHMsIHNhbXA
sIHNtYWxsLCBzdHJpa2UsIHN0cm9uZywgc3...",
    "Ds_RtpBackupTemplate":
"eyJjaGFsbGVuZ2VJbmZvVGV4dCI6IkludHJvZHV6Y2EgZWwgY8OzZGlnbyBxdWUgaGEgbGxlZ2F
kbyBhIHN1IHRlbMOpZm9ubyIsInN1Ym1pdEF1dGhlbnRpY2F0aW9uTGFiZWwiOiJFbnZpYXIiLCJ
pc3N1ZXJJbWFnZSI6Imh0dHBzOi8vcHBpaS1pLnJlZHN5cy5lcy9jMmVyLXdlYi9zdGF0aWMvaW1
nL2xvZ29zLzk5OTkucG5nIiwiaWRPcGVyYWNpb24iOiIxODQ0Njg0ODI3MjYyNzgzODYzMjI4OTM
2MzEzNjA4NzU4NyIsImJpenVtSW1hZ2UiOiJodHRwczovL3BwaWktaS5yZWRzeXMuZXMvYzJlci1
3ZWIvc3RhdGljL2ltZy9sb2dvLWJpenVtLnBuZyJ9"
}
```


As we can see, **BIZ00233** indicates that this is an operation that can be backed up. There are two ways to do this: using a template or HTML.

HTML integration consists of including a complete HTML form provided by Bizum within the merchant's flow. The merchant must present the client with a standard form defined by Bizum. In this form, the client will enter the received OTP to authenticate the operation.

In template integration, Bizum provides the necessary elements for the merchant to build their own form. Fields included:

- **challengeInfoText:** Code received on the mobile phone.
- **submitAuthenticationLabel:** Text that will be displayed on the form submission button.
- **IssuerImage:** Issuer's logo.
- **idOperacion:** Operation identifier.
- **bizumImage:** Bizum logo.

10.1 Invoke backup

Regardless of whether the merchant decides to integrate the backup via HTML or template, they must call this service so that the OTP is sent to the client, which is necessary to validate the backed-up authentication.

The flow for invoking backup in Bizum follows this pattern:

1. The merchant requests the invocation of the backup.
2. An attempt is made to invoke the backup.
    - We respond with the success of the backup invocation by sending the field **Ds_RtpResponse=BIZ00000** and **Ds_RtpDescription=Operacion realizada correctamente.**
    - We respond with the backup invocation error by sending the field **Ds_RtpResponse=BIZXXXXX** (where XXXXX is the identifier of the error concerned) and **Ds_RtpDescription=YYYYY** (where YYYYY is the detailed description of the error that occurred).
3. If the backup invocation was successful, there will be two bifurcations, depending on whether the merchant decided to back up the operation using the template or HTML.
    - **Template:** If the backup is supported using the template integration, the merchant must ask the client for the OTP that they should have received after the merchant invoked the backup. The merchant must then call the **backup validation** service to complete the client's authentication.
    - **HTML:** If backup is performed using HTML integration, the merchant must display the OTP receipt form included in the Ds_RtpBackupTemplate field, and when the client enters the OTP in that form, the operation will be automatically authenticated without the merchant having to call the backup validation service.

10.1.1 Input parameters

The input parameters for initiating backup are the basic data of an operation, in addition to the following:

Field	Description	Type	Format
DS_MERCHANT_BIZUM_MOBILENUMBER	Client's phone number with prefix	String	^[\+]?[\d]{5,15}$ E.g.: + 34700000000

Please note that the **DS_MERCHANT_ORDER** must be the same as the one for the operation that can be backed up.

The URL where the service is displayed is as follows, depending on the environment:

- **TEST environment**: *https://sis-t.redsys.es:25443/sis/rest/RTP/respaldarUsuarioRTP*
- **PRODUCTION environment**: *https://sis.redsys.es/sis/rest/RTP/respaldarUsuarioRTP*

10.1.2 Output parameters

- **Ds_RtpStatus**: Indicates whether the backup has been invoked. Values are 'OK' if possible and 'KO' if not possible. If 'KO' is returned, the 'Ds_RtpResponse' field must be consulted to verify whether the response received is due to an error or not.
- **Ds_RtpResponse**: Backup invocation request code. If the request is correct, 'BIZ00000' will be returned. Otherwise, another error code will be returned following the regular expression '**^BIZ[\d]{5}$**'.
- **Ds_RtpDescription**: Description of the backup invocation request. If the 'Ds_RtpResponse' field has a value other than 'BIZ00000', this field will indicate a description of the reason the request was not successful.

10.1.3 Example

- Input example

- Encoded data:


```
{
    "Ds_SignatureVersion": "HMAC_SHA512_V2",
    "Ds_MerchantParameters":
"eyJEU19NRVJDSEFOVF9BTU9VTlQiOiI3NjAwIiwiRFNfTUVSQ0hBTlRfT1JERVIiOiIxNzY1Nzk
2MDM3IiwiRFNfTUVSQ0hBTlRfTUVSQ0hBTlRDT0RFIjoiOTk5MDA4ODgxEU19NRVJDSEFOVF9QQV
lNRVRIT0RTIjoieiIsIkRTX01FUkNIQU5UX0JJWlVNX01PQklMRU5VTUJFUiI6IiszNDcwMDAwMD
AwMCJ9",
    "Ds_Signature": "WIFG3-
kUOatw5eSsWktWsXF2G6rXVjhiRBGIAT15c6iNRAHfGAPLt7SgrWn420oHDQY2RXwS3xJxu5ugcX
kqww"
}
```


- Decoded data:

```
{
```

```
    "DS_MERCHANT_AMOUNT": "7600",
    "DS_MERCHANT_ORDER": "1765796037",
    "DS_MERCHANT_MERCHANTCODE": "999008881",
    "DS_MERCHANT_CURRENCY": "978",
    "DS_MERCHANT_OPERATIONTYPE": "0",
    "DS_MERCHANT_TERMINAL": "810",
    "DS_MERCHANT_PAYMENTMETHODS": "z",
    "DS_MERCHANT_BIZUM_MOBILENUMBER": "+34700000000"
}
```


- Output example

- Encoded data:


```
{
    "Ds_SignatureVersion": "HMAC_SHA512_V2",
    "Ds_MerchantParameters":
"eyJEc19SdHBTdGF0dXMiOiJPSyIsIkRzX1J0cFJlc3BvbnNlIjoiQklaMDAwMDAiLCJEc19SdHB
EZXNjcmlwdGlvbiI6Ik9wZXJhY2lvbiByZWFsaXphZGEgY29ycmVjdGFtZW50ZSIsIkRzX01lcmN
oYW50Q29kZSI6Ijk5OTAwODg4MSIsIkRzX1Rlcm1pbmFsIjoiODEwIiwiRHNfT3JkZXIiOiIxNzY
1Nzk2MDM3IiwiRHNfQ3VycmVuY3kiOiI5NzgiLCJEc19BbW91bnQiOiI3NjAwIn0=",
    "Ds_Signature": "iAfK9dwoUl7i3HiPiBZlosKGRnY5kbMPj_DG-
ivZBWZAHB_ZQZGiQKb69kb0DX0cxcx1DcVy7hNmNpLD6teMYw"
}
```


- Decoded data:


```
{
    "Ds_RtpStatus": "OK",
    "Ds_RtpResponse": "BIZ00000",
    "Ds_RtpDescription": "Operation performed correctly",
    "Ds_MerchantCode": "999008881",
    "Ds_Terminal": "810",
    "Ds_Order": "1765796037",
    "Ds_Currency": "978",
    "Ds_Amount": "7600"
}
```


10.2 Validate backup

If we use template backup, we must send a backup validation request with the OTP provided by the client.

A fictional OTP can be used for the test environment; in this case, 1234 has been chosen as the OTP.

10.2.1 Input parameters

The input parameters will be the same as in the backup invocation, with the addition of the OTP.

Field	Description	Type	Format
DS_MERCHANT_BIZUM_MOBILENUMBER	Client's phone number with prefix	String	^[\+]?[\d]{5,15}$ E.g.: + 34700000000
DS_MERCHANT_OTP	Code sent to client	Numeric	

Please note that the **DS_MERCHANT_ORDER** must be the same as the one used in the backup call operation.

The URL where the service is displayed is as follows, depending on the environment:

- **TEST environment**: *https://sis-t.redsys.es:25443/sis/rest/RTP/validarRespaldoUsuarioRTP*
- **PRODUCTION environment**: *https://sis.redsys.es/sis/rest/RTP/validarRespaldoUsuarioRTP*

10.2.2 Output parameters

- **Ds_RtpStatus**: Indicates whether the backup has been invoked. Values are 'OK' if successful and 'KO' if unsuccessful. If 'KO' is returned, the 'Ds_RtpResponse' field should be consulted to verify whether the response received is due to an error or not.
- **Ds_RtpResponse**: Backup invocation request code. If the request is correct, 'BIZ00000' will be returned. Otherwise, another error code will be returned following the regular expression **'^BIZ[\d]{5}$'**.
- **Ds_RtpDescription**: Description of the backup invocation request. If the 'Ds_RtpResponse' field has a value other than 'BIZ00000', this field will indicate a description of the reason the request was not successful.

10.2.3 Example

- Input example
- Encoded data:


```
{
    "Ds_MerchantParameters":
"eyJEU19NRVJDSEFOVF9NRVJDSEFOVENPREUiOiI5OTkwMDg4ODEiLCJEU19NRVJDSEFOVF9URVJ
NSU5BTCI6IjgxMCIsIkRTX01FUkNIQU5UX09SREVSIjoiMTc2NTc5NjAzNyIsIkRTX01FUkNIQU5
UX0NVUlJFTkNZIjoiOTc4IiwiRFNfTUVSQ0hBTlRfQU1PVU5UIjoiNzYwMCIsIkRTX01FUkNIQU5
UX0JJWlVNX01PQklMRU5VTUJFUiI6IiszNDcwMDAwMDAwMCIsIkRTX01FUkNIQU5UX09UUCI6MTI
zNH0",
    "Ds_Signature": "gIiiPvyXmN1ZQcDW63Nt98PmGAEtXZeulpg_-
W_OnVCCx9fExT6I946ZCbQY9ZRl7lzFAkgfKuL4xShLas1XUA",
    "Ds_SignatureVersion": "HMAC_SHA512_V2"
}
```


- Decoded data:

```
{
    "DS_MERCHANT_MERCHANTCODE": "999008881",
    "DS_MERCHANT_TERMINAL": "810",
    "DS_MERCHANT_ORDER": "1765796037",
    "DS_MERCHANT_CURRENCY": "978",
    "DS_MERCHANT_AMOUNT": "7600",
    "DS_MERCHANT_BIZUM_MOBILENUMBER": "+34700000000",
    "DS_MERCHANT_OTP": 1234
}
```


- Output example

- Encoded data:


```
{
    "Ds_SignatureVersion": "HMAC_SHA512_V2",
    "Ds_MerchantParameters":
"eyJEc19SdHBTdGF0dXMiOiJPSyIsIkRzX1J0cFJlc3BvbnNlIjoiQklaMDAwMDAiLCJEc19SdHB
EZXNjcmlwdGlvbiI6Ik9wZXJhY2lvbiByZWFsaXphZGEgY29ycmVjdGFtZW50ZSIsIkRzX01lcmN
oYW50Q29kZSI6Ijk5OTAwODg4MSIsIkRzX1Rlcm1pbmFsIjoiODEwIiwiRHNfT3JkZXIiOiIxNzY
1Nzk2MDM3IiwiRHNfQ3VycmVuY3kiOiI5NzgifQ==",
    "Ds_Signature":
"7h8Muy2BDKex0h_9ua85bRydbwO4PnS_ArjLYLLS_24Mev8DLSYb84cJRnxvdM0_pvg85l55BNF
Te3abpfjpvg"
}
```


- Decoded data:


```
{
    "Ds_RtpStatus": "OK",
    "Ds_RtpResponse": "BIZ00000",
    "Ds_RtpDescription": "Operation performed correctly",
    "Ds_MerchantCode": "999008881",
    "Ds_Terminal": "810",
    "Ds_Order": "1765796037",
    "Ds_Currency": "978"
}
```


11. Operational and functional queries

If you have any queries regarding the Virtual POS, you should contact the financial entity with which you operate.

Enquiries regarding requests for Virtual POS documentation, registration data requests, definition of payment methods, enquiries about merchant configuration data, merchant user enquiries, etc. should always be directed to your financial entity.

Furthermore, the REDSYS client service centre does not provide consulting services (enquiries about the code to be developed to connect to the Virtual POS).