Classic Payment API

SOPG (Service Oriented Prepaid Gateway - xml based protocol) Documentation

Version history

Version	Date	Description	Author
0.1	10.03.2013	Initial draft	Paul Kneidinger
0.2	20.03.2013	Added details and restructured	Matthias Vilsecker
0.3	21.03.2013	Added In App payment	Matthias Vilsecker
0.4	24.04.2013	Review and further input	Natasa Jeremic
1.0	02.10.2013	Final document	Natasa Jeremic
1.1	27.01.2014	Minor changes	Natasa Jeremic
1.2	31.10.2017	Refreshed	Eugen Nemecek
1.3	22.01.2020	MCID requirements added	Lucas Hannel
1.4	21.09.2020	Added Logo Endpoint	Techsupport
1.5	08.01.2021	Endpoint updated	Techsupport

For technical questions about implementation please contact integration@paysafecard.com

Table of Contents

1. Introduction

This document gives a detailed overview about the usage and parameters of paysafecard's Service Oriented Prepaid Gateway (SOPG). The gateway is a SOAP XML web service which reveals API client functions that can be used with any SOAP-capable client system. In the first chapter, a functional overview of the payment process is given. Following that, the paysafecard SOPG is presented in detail. Then the systems of paysafecard, as well as the technical details and a complete description of functions and parameters, are provided.

1.1 Product introduction for customers

This section provides an overview of paysafecard customer products relevant in the context of the paysafecard SOPG interface.

1.1.1 paysafecard

paysafecard allows customers to pay safely online without a bank account or credit card. The customer buys paysafecard at a point of sale in the form of a printout with a 16-digit PIN, and pays by entering the PIN in the paysafecard payment panel of your online shop.

1.1.2 my paysafecard

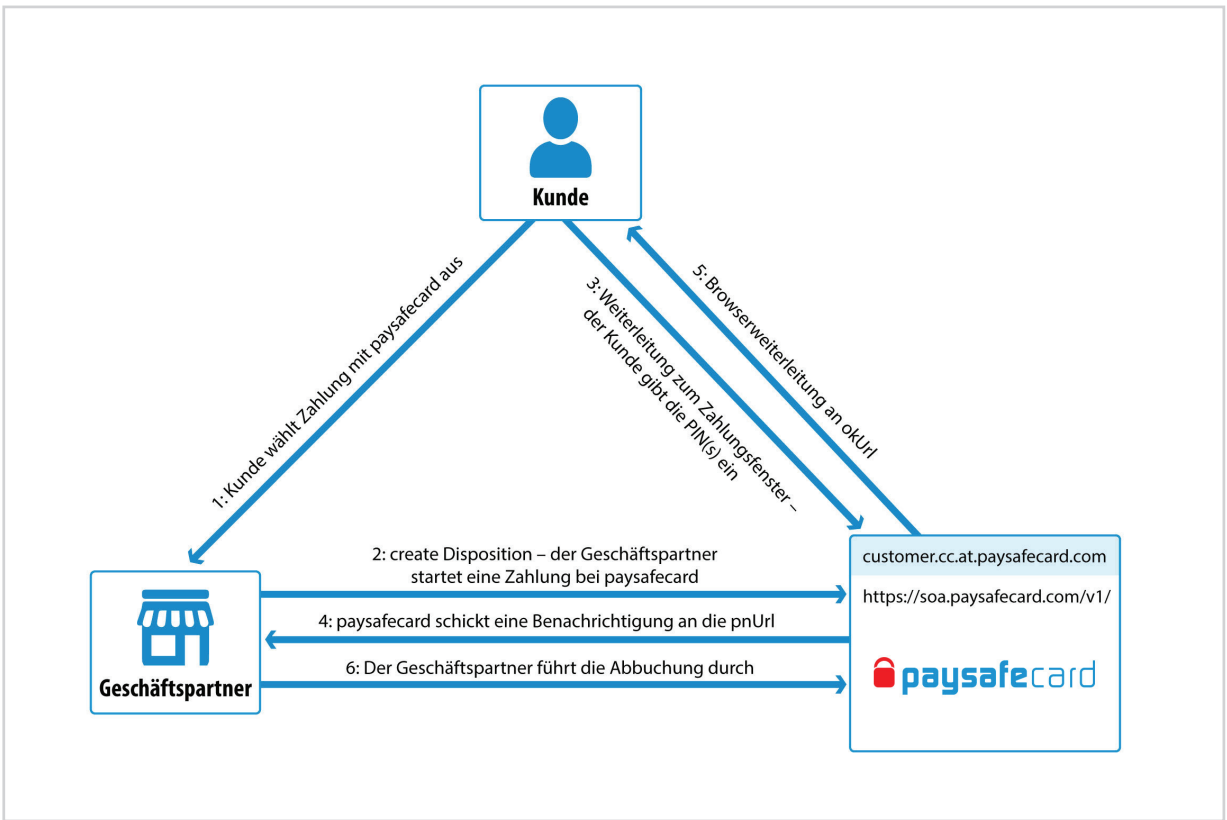
Aside from paying with a PIN, paysafecard also offers users my paysafecard, an account for the customer's PINs. The customer signs up for my paysafecard and tops up the account with purchased paysafecard PINs. The same applies for the my paysafecard administration tool*. The customer can then access the combined value of all paysafecard PINs added to the my paysafecard account. Payments are made simply and safely by entering their username and password.

- The my paysafecard administration tool is an alternative to the classic my paysafecard account, and can only be used in Australia, Canada, Lithuania, Mexico, New Zealand, and Uruguay. The my paysafecard administration tool cannot be used for product payouts or refunds.

2. Payment function overview

Every payment involves three parties: A customer, a business partner, and the paysafecard company („PSC“). Payments take place in „payment transactions“ or „dispositions“, which are uniquely identified by a „merchant transaction ID“ („MTID“) and which carry a value („amount“) that is typically the amount of money for which a customer buys something.

2.1 Payment process



3. About SOPG

The Service Orientated Prepaid Gateway (SOPG) allows paysafecard to offer payment functions as a web service. The web service is based on SOAP protocol and can, regardless of programming language, be used by all SOAP clients. The complete payment procedure is processed between the business partner‘s system and the paysafecard SOPG.

3.1 Prerequisites

A business partner can only connect to paysafecard‘s systems if the following prerequisites are fulfilled:

- They possess SOPG user details (username/password) assigned by paysafecard
- The IP address of the payment server has been authorised (if, when attempting to access the server, an „Error 403“ is displayed, assume that the IP address has not yet been unlocked).

3.2 Error handling

All SOPG operations will return an „Errorcode“ and a „ResultCode“. The „ResultCode“ can contain the following values: „0“ (successful), „1“ (logical problem), or „2“ (technical problem).
In general, the following rules apply:

- „1“ indicates that there is a problem with the submitted data (e.g. wrong login details, transaction expired, etc.). A new attempt using the same information will not be successful.
- „2“ (technical problem) indicates that the service is temporarily unavailable – the request can be reattempted.

Below is an example of a failed request:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns1="urn:pscservice">
  <soap:Body>
    <ns1:createDispositionResponse>
      <ns1:createDispositionReturn>
        <ns1:>transactionID1234</ns1:>
        <ns1:subId>Merchant1234</ns1:subId>
        <ns1:mid xsi:nil="true" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"/>
        <ns1:resultCode>1</ns1:resultCode>
        <ns1:errorCode>10008</ns1:errorCode>
      </ns1:createDispositionReturn>
    </ns1:createDispositionResponse>
  </soap:Body>
</soap:Envelope>
```

3.2.1 Error message descriptions

Please bear in mind that error messages to the customer will be displayed only on the paysafecard payment window. For other messages, only the error code will be provided. %1, %2, ... are placeholders for different values, e.g. MID. The most common error codes can be found in Chapter 10.

3.3 Content type and character set

Please make sure that the content type in the http header, when submitting requests, is set to: Content-Type: text/xml;charset=UTF-8

4. Definition of paysafecard systems

4.1 API service environment

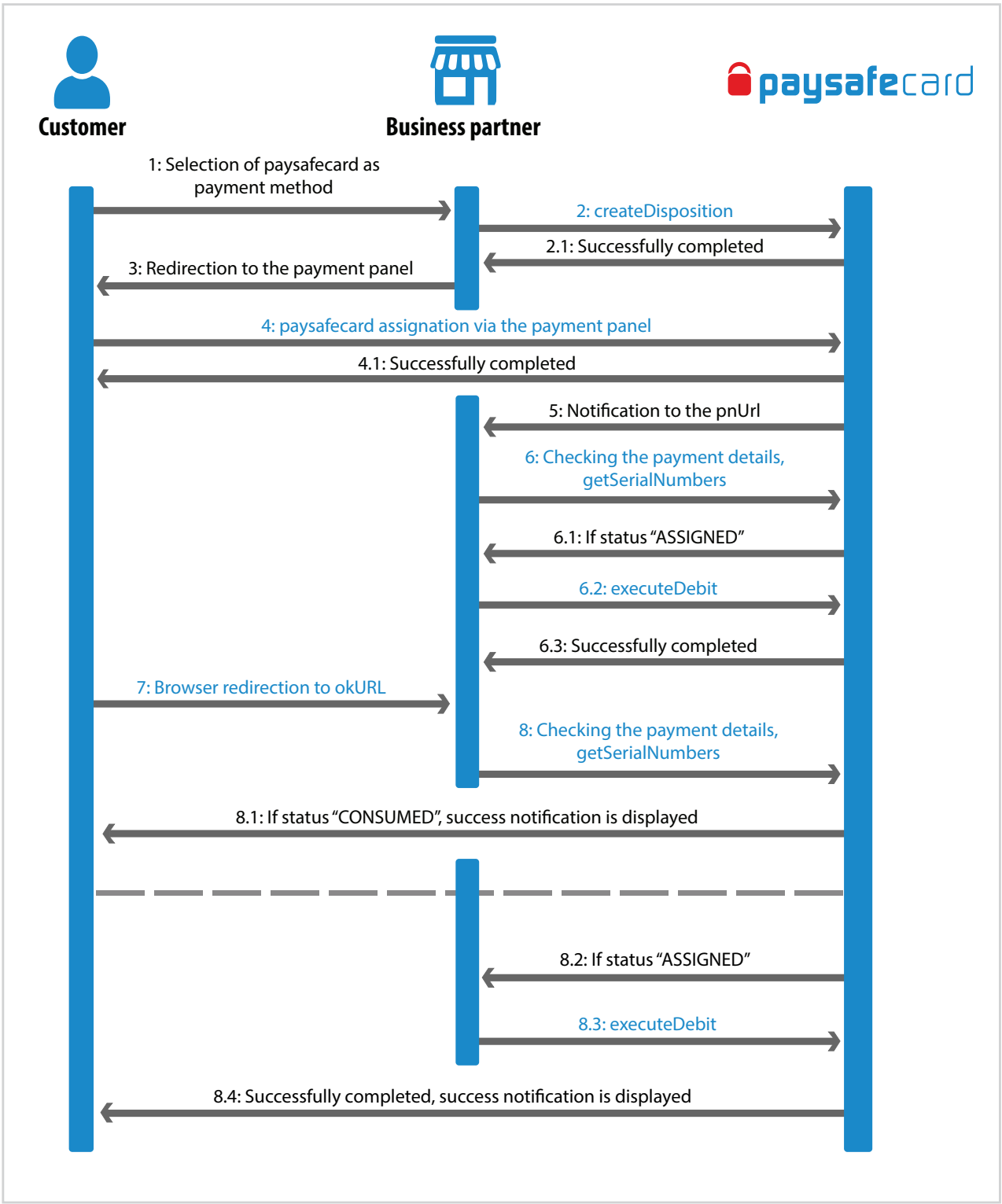
paysafecard provides the „paysafecard test system“ (SOATEST), a test environment for the integration of new business partners. The integration of new business partners initially occurs within this test system, for which no IP addresses are added to the whitelist. In the Merchant Service Center, each new on-boarding process is accompanied by an automated integration test in which, following the SOPG payment process, the code in the backend of the website is checked. Once the integration test is passed, the business partner can be moved to the productive systems which, for the business partner, consists of the following steps:

- Changing to productive login details (all assigned by paysafecard)
- Replacing the service endpoint URL
- Replacing the WSDL URL

All details are made available by the Merchant Service Center.

- The endpoint for the test environment is: https://soatest.paysafecard.com/psc/services/PscService
- The endpoint for the productive environment is: https://soa.paysafecard.com/psc/services/PscService

5. Technical Overview



6. Function details and WSDL structure

This document describes all the necessary functions required for the basic payment process of the SOPG WSDL structure. Please note that the contract includes many more features. All necessary payment parameters are required, and a transmission is obligatory although the value remains NULL. If the web service framework requires a WSDL in real time, the SOPG WSDL of the WSDL URL needs to be downloaded and provided in the local environment.
NOTE: It is not possible to retrieve the WSDL from the paysafecard servers in real time.

6.1 createDisposition functions

The business partner initiates the payment process by sending a „createDisposition" request to paysafecard, so as to create a disposition at the server. The maximum allowed amount is 1,000.00 EUR (or the equivalent in other currencies).

Request – Elements	Response – Elements
SOPG-Username SOPG-Password MerchantTransactionID MerchantclientID SubID Amount Currency okUrl nokUrl pnUrl ClientIp Dispositionrestrictions ShopId Shoplabel	MID ResultCode ErrorCode SubID

6.2 getSerialNumbers function

Calls up the disposition status and checks if it matches the expected state before calling upon the next function.

Request – Elements	Response – Elements
SOPG-Username SOPG-Password MerchantTransactionID SubID Currency	MTID SubID ResultCode ErrorCode Amount Currency DispositionState SerialNumbers

6.3 executeDebit function

Withdraws the money from the customer's paysafecard. This step concludes the payment if the „close" flag is set to „1". The business partner can keep the transaction open (while the full amount is reserved) until the end of the disposition time. If the business partner wants to close the disposition without executing anything, the function must be called upon with „amount=0.00".

Request – Elements	Response – Elements
SOPG-Username SOPG-Password MerchantTransactionID SubID Currency Close	MTID SubID ResultCode ErrorCode

6.4 Payment notification

The payment notification is used to notify the business partner of the PIN assignation in the payment window, independent of the customer's behavior. This service ensures that dispositions can be completed before topping up the customer's account, and is therefore highly recommended in order to avoid transaction expiration. When a customer enters a PIN in the payment window, our API sends a POST request to the pnUrl defined by the business partner in the createDisposition request. To confirm successful delivery, our Payment Notification Server awaits an HTTP-200 response. For transfer to a secured pnUrl (https), the most common certificate authorities are stored in our certificate root store. A pnUrl with a registered port (:40) does not work.

A payment notification will only be sent after successful paysafecard assignation. This notification will also be forwarded to the „okUrl". In the case of technical application errors on the part of the business partner, the payment notification is transmitted again.

Output parameters	Merchant response elements
Mtid eventType (ASSIGN_CARDS is returned) serialNumbers currency disposition amount cardTypeId	HTTP 200

6.4.1 Repeating payment notification

In the case of technical errors (e.g. socket timeout), or application errors (e.g. HTTP 500 response), the payment notification is resubmitted at regular intervals until one of the following criteria is fulfilled:

- The payment notification is successfully delivered (i.e. HTTP 200 response from payment server).
- The maximum number of attempts has been reached (currently configured at five attempts).

The payment notification is repeated at the following intervals:

- 1. Delivery attempt immediately after PIN input
- 2. Delivery attempt one (1) second after PIN input
- 3. Delivery attempt sixty (60) seconds after PIN input
- 4. Delivery attempt two (2) minutes after PIN input
- 5. Delivery attempt three (3) minutes after PIN input

The period between PIN assignment and PIN validation is configured as being 1-10 minutes, as per the T&Cs. Depending on the configuration of the disposition time frame (chapter 8.3.1), the last three delivery attempts may not be executed because the transaction has already expired.

7. Payment process

1. Initiate payment with the createDisposition function.

   1.1 If the answer is correct (errorcode & resultcode = 0) -> Take the customer to our payment window.

   1.2 If the answer is wrong (errorcode & resultcode not 0) -> Show the customer the following error message: **„The transaction could not be completed due to technical problems. If this problem persists, please contact (business partner) support."**

2. The customer is redirected to the payment window.

3. The customer enters a valid paysafecard PIN in the payment window and clicks „Pay".

   3.1. The PIN will be validated on the paysafecard payment window. If the customer aborts the transaction at the payment window, please display the following error message: **„Payment cancelled by customer".**

4. Since the card is assigned to the transaction (payment status S for „Disposed"), we will send the payment notification to your pnURL

5. After receiving the payment notification, respond with HTTP code 200 and check the status of the transaction by requesting the getSerialNumbers function.

   5.1 If the disposition status returns the getSerialNumbers function with a status O value, the transaction has already been completed.

   5.2. If the disposition status returns the getSerialNumbers function with a status S value, the executeDebit function should be performed.

      5.2.1. If the answer is correct (errorcode & resultcode = 0) -> Please top up the end customer account.

      5.2.2. If the answer is wrong (errorcode & resultcode not 0) -> No action required.

6. The end customer is forwarded to the okUrl.

7. You can check the transaction by requesting the getSerialNumbers function.

   7.1 If the disposition status returns the getSerialNumbers function with a status „R" or „X", display the following error message to the end user: „The payment could not be completed due to a temporary connection problem. If this problem persists, please contact (business partner) support."

   7.2. If the disposition status returns the getSerialNumbers function with a status O value, the transaction has already been completed. Please display a success message to the end customer.

   7.3 If the disposition status returns the getSerialNumbers function with a status S value, the executeDebit function should be performed.

7.3.1 If the answer is correct (errorcode & resultcode = 0) -> Please top up the end customer account and display a success message to the end customer.

7.3.2 If the answer is wrong (errorcode & resultcode not 0) - > Please display an error message to the end user. The error message reads: **„The payment could not be completed due to a temporary connection problem. If this problem persists, please contact (business partner) support."**

8. Parameters

8.1. Description of parameters

username – individual account username
- provided by paysafecard for authentication

password – individual account password
- provided by paysafecard for authentication

mtid – (unique) transaction ID, unique identifier for each disposition.
- Max. length: 60 characters
- recommended: up to 20 characters
- provided by business partner
- Only the following characters are allowed: A-Z, a-z, 0-9, as well as - (hyphen) and _ (underline)
- Example: 3516-6s4dfsad41

subId – mandatory parameter, value must be left empty if nothing else is agreed.
- So-called „reporting criteria", offering the possibility to classify transactions.
- Max. length: 8 characters max. (capital and lower-case letters)
- In agreement with paysafecard.
- Required example : shop1

amount – disposition amount
- Requested amount is not able to exceed a value of 1,000.00 EUR (or equivalent in other currency)
- Max. 11 digits before – exactly 2 digits after decimal point
- using a full-stop/period as a decimal separator
- Example: 100.00

currency – disposition currency
- Max. length: 3 upper-case letters,
- ISO currency code
- Example: EUR

pnUrl – payment notification URL of which paysafecard notifies the business partner as soon as a
- paysafecard disposition has been successfully allocated (more details in Chapter 6.2).
- URL must be absolute and URL-encoded,
- as they must be defined by the business partner and sent as a parameter
- Max. length: 765 characters

okUrl – the URL to which the customers are forwarded following the successful redemption of their paysafecard PINS.
- The business partner may include some information in the URL.
- URL must be absolute and URL-encoded as they are sent as a parameter.
- Max. length: 765 characters.

nokUrl – this is the URL to which customers are forwarded by paysafecard when they
- click the ‚cancel' button on the paysafecard payment window.
- URL must be absolute and URL-encoded as they are sent as a parameter.
- Max. length: 765 characters.

It is crucial to send the „okURL", „nokURL" and optional „pnUrl" in a URL-encoded (also called percent encoding) format. Otherwise, the result will be an incorrect redirection of the customer to the confirmation page, in addition to a possible failure of the payment.

merchantClientID (MCID) – Also known as „MCID", the merchantClientId is an important parameter for the integration of paysafecard. The merchantClientId identifies the Customer on our business partners side.
The most optimal merchantClientId is a completely random value. A value that uniquely identifies the customer and is disconnected from any personal information. This merchantClientId value should be the same for all transactions of the customer.

Here are Guidelines for possible merchantClientIds:
Valid Values:

Value	Type
• 22c3be0b50c7a5f1964a63d78f38a6ffc41c027e9	SHA1 - test@123.com
• 742f2b1a55cd5d606ea44b4fcb54646a	MD5 - test@123.com
• 3a5b0d0777dead9df93d502df85c8180e53804eb	SHA1 - UsernameValue1
• 3192481752123	Random Customer Identifier
• CustomerID1	Customer Identifier free of personal information

Invalid Values:
- test@123.com
- Username_1
- FirstName123
- LastName123
- Timestamp
- IP Address

Please note that sending any form of the invalid values will not be accepted.
If you intend to process paysafecard transactions on multiple brands, please inquire about the possibilities of separating multiple entities for your account.

clientIp – the IP address of the paysafecard customer.
shopId – identification of the shop originating the request. This is mostly used by payment service providers who act as a proxy for other payment methods as well.
- Max. length: 60 characters
- Recommended: up to 20 characters
- provided by business partner
- Characters permitted are: A-Z, a-z, 0-9, as well as - (hyphen) and _ (underline)
- Example: 2568-B415rh_785

shopLabel – label or URL of the shop which is the originator of the request, related to the „shopId".
„shopId". This is most likely used by payment service providers who act as a proxy for other payment methods as well.
- Max. length: 60 characters
- Example: www.foodstore.com

mid – merchant ID, unique ID of the merchant/currency pair.
- 10 digits long
- provided by paysafecard
- Example: 1000001234

dispositionState – current state of the disposition (see Chapter 8.3 for more details).

serialNumbers – serial number(s) of assigned paysafecard(s) by the customer, after entering the PIN at the paysafecard payment window (values separated by semicolons).

- currency: ISO currency code
- disposition amount: amount reserved for this disposition on the paysafecard of the customer
- cardTypeId: paysafecards are grouped into card types: e.g. junior_paysafecard;
- adult_paysafecard; inhouse_paysafecard
- Example:
  0000000001200000;EUR;7.50;00002;
  0000000001300000;EUR;5.50;00002;

close – the close flag of the disposition can be set to „0" or „1" to indicate if further actions will be executed or not.

- „0" [don't close transaction]
- „1" [close transaction, this is the last debit]

resultCode – result code of the operation (see the result codes chapter for details).
errorCode – error code of the operation (see the error codes chapter for details).
dispositionRestrictions – disposition restrictions can be set by the business partner in order to restrict a payment transaction, according to their individual needs. Details in Chapter 8.2.

- Multiple repeats possible
- Each restriction consists of a „key" and a „value":
- „key" - the key of the restriction
- „value" - the value of the restriction

8.2. Restrictions

Please transmit a country code (ISO 3166-1) via the createDisposition API request to restrict payments to the specified country. Please do not define a country restriction in the „create disposition" API Request (COUNTRY) during the integration test in the Merchant Service Center. The integration test uses only one card type (AT_ Classic).

Key	Example value	possible values	Description
COUNTRY	DE	All countries in which paysafecard is available (e.g. FR, ES, etc.)	Restricts the processing of payments to Germany only. The value accepts ISO 3166-1 country codes.

The following restrictions are available for a paysafecard payment with a paysafecard account (my paysafecard):

Key	Example value	possible values	Description
MIN_AGE	18	must be a positive number value	Restricts my paysafecard account holders to only those aged 18 years or older.
MIN_KYC_LEVEL	FULL	SIMPLE or FULL	Restricts my paysafecard account holders to only those of a given status, here FULL.

8.3.Disposition status

One letter code	Meaning	Description
R	Created	The disposition has been successfully created. If nothing happens within 30 minutes, the disposition will be set to „X“ status by paysafecard.
S	Disposed	The customer‘s paysafecard has been successfully assigned to the disposition. The business partner can use „executeDebit“; no debits have taken place so far.
O	Consumed	The final debit with close=1 has been processed; no further debits are possible.
L	Cancelled	The disposition has been actively cancelled by the customer.
X	Expired	The time frame for this disposition has ended (either before a paysafecard was assigned or before „executeDebit“ was called up).

8.3.1 Disposition time frame

Once a disposition is in status „S“ („DISPOSED“), the business partners must carry out their debits as per their settings. Fundamentally, the disposition time window is 60 seconds. In consultation with the paysafecard integration team, this setting can be changed anywhere up to 10 minutes, as per the T&Cs. If this time frame is exceeded, the disposition will automatically expire, and the amount will be available again on the customer‘s paysafecard. Furthermore, all dispositions which have been created, but not successfully debited, will be set to „EXPIRED“.

NOTE: These jobs are only active on the paysafecard productive server. On the test system, transactions and reserved amounts in the „S“ („DISPOSED“) state are committed until the transaction completes with executeDebit.

8.4. Payment expiration window

Each initiated transaction remains in „R“ („CREATED“) status for 30 minutes. If the transaction does not receive a valid paysafecard PIN, the transaction is set to „X“ („EXPIRED“) and has therefore expired.

8.5. Payment panel details

8.5.1 Desktop payment panel

The paysafecard payment panel can be presented in a popup window or, alternatively, in an iFrame.

To make sure the whole payment panel is visible to the user, please always allow vertical scrolling or dynamic sizing.

- The width is fixed to 600px
- The height is fixed to 840px

8.5.2 Mobile device payment panel

paysafecard‘s payment panel is optimized for mobile devices. If a customer is using a device with a resolution lower than 600px, an optimized payment panel will be shown. This is also the case if the embedded iFrame has a width less than 600px.

NOTE: The iFrame for embedding the desktop payment panel always needs to have a width of at least 600px. Otherwise, the mobile version of the payment panel will be displayed.

8.6 Location and language settings

For backward compatibility, all existing language parameters still yield the same results as in former versions of the API, but every language will be automatically transformed into regional schema.

Basically, the language and location of the payment panel are determined by the following rules:

1. Has the customer already visited the payment panel? Call up the regional schema from the seeded cookie.
2. Determine the location from the IP address of the customer.
3. Use the value from the location parameters.
4. Use the value from the language parameters.
5. Call up the locations from the browser header.
6. Use the fallback schema (de_de). Therefore, it is not obligatory to set a regional schema parameter.

9. Example payment

In this chapter, a test scenario is presented using example data. In practice, it will differ from business partner to business partner whether one or more debits or status-requests will be executed. Do not use the enclosed example data! Each business partner will receive a consistent set of test data for testing purposes.

9.1 createDisposition

The business partner initiates the payment process by sending a ‚createDisposition' request.

Example request:

```
<urn:createDisposition>
  <urn:username>USER</urn:username>
    <urn:password>PASSWORD</urn:password>
      <urn:>18b02d230-a6822f-4cbb-aee9-0bc07d90cfa4</urn:>
        <!--Zero or more repetitions:-->
          <urn:subId></urn:subId>
            <urn:amount>10.00</urn:amount>
              <urn:currency>EUR</urn:currency>
                <urn:okUrl>http%3a%2f%2fwww%2epaysafecardokURL%2ecom</urn:okUrl>
                  <urn:nokUrl>http%3a%2f%2fwww%2epaysafecardnokURL%2ecom</urn:nokUrl>
                    <urn:merchantclientid>cID_919191</urn:merchantclientid>
                  <urn:pnUrl> http%3a%2f%2fwww%2emerchantpnURL%2ecom </urn:pnUrl>
                <!--Zero or more repetitions:-->
              <urn:dispositionRestrictions>
            <urn:value>FR</urn:value>
          <urn:dispositionRestrictions>
        <urn:key>MIN_AGE</urn:key>
      <urn:value>18</urn:value>
    </urn:dispositionRestrictions>
  <!--Optional:-->
</urn:createDisposition>
```

Example response:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <ns1:createDispositionResponse xmlns:ns1="urn:pscservice">
      <ns1:createDispositionReturn>
        <ns1:>18b02d230-a6822f-4cbb-aee9-0bc07d90cfa4</ns1:>
          <ns1:mid>1000001234</ns1:mid>
            <ns1:resultCode>0</ns1:resultCode>
            <ns1:errorCode>0</ns1:errorCode>
          </ns1:createDispositionReturn>
        </ns1:createDispositionResponse>
      </soapenv:Body>
    </soapenv:Envelope>
  </soapenv:Body>
</soapenv:Envelope>
```

9.2 getCustomerPanel

The „createDisposition" command was successfully executed. Thus, the customer can be forwarded to the paysafecard payment panel for assigning cards to the disposition.

Example URL test system:

```
https:// customer.test.at.paysafecard.com/psccustomer/GetCustomerPanelServlet
```

Example URL productive system:

```
https:// customer.cc.at.paysafecard.com/psccustomer/GetCustomerPanelServlet
?mid=1000001234
&=18b02d230-a6822f-4cbb-aee9-0bc07d90cfa4
&amount=10.00
&currency=EUR
```

Input parameters example:

```
PIN:          0000 0000 1234 5678
Terms of Use: <checkbox, default unchcecked>
```

9.3 pnUrl request

The paysafecard system sends an „HTTP POST" request to the business partner's system („pnUrl") in order to give notice of the successful assignation of the customer's paysafecard's.

Example URL:

```
http://www.merchantpnURL.com/notifyME
?=3516-6s4dfsad41
&eventType=ASSIGN_CARDS
&serialNumbers=0000000001200000;EUR;100.00;DE00002
```

Response:

```
HTTP 200
```

9.3.1 Payment notification supported country codes

As an additional information parameter, the country code (country a paysafecard is sold in) is part of the standard payment notification API request. The parameter „cardTypeId“ in the payment notification provides a combination of default ISO country code and cardtype ID. Exception: Some cards are not assigned to any specific country. Therefore, no country code will be provided.

Basic pnUrl response

```
mtid=<Mtid>
&eventType=<eventType>
&serialNumbers=<serialNr1>;<currency1>;<amount1>;<cardTyp1>;
<serialNr2>;<currency2>;<amount2>;<cardType2>;
Example pnUrl
```

Example - pnUrl response

```
mtid=123456
&eventType=ASSIGN_CARDS
&serialNumbers=0000000001200000; EUR; 50.00; XX00004;
0000000001200001; EUR; 50.00; XX00004
```

9.4 getSerialNumbers

The business partner checks the transaction status using the „HTTP-GET“ request.

Example URL:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:pscservice">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:getSerialNumbers>
      <urn:username>USER</urn:username>
      <urn:password>PASSWORD</urn:password>
      <urn:mtid>transactionID123456</urn:mtid>
      <!-Zero or more repetitions:->
      <urn:subId></urn:subId>
      <urn:currency>EUR</urn:currency>
    </urn:getSerialNumbers>
  </soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="urn:pscservice">
  <soap:Body>
    <ns1:getSerialNumbersResponse>
      <ns1:getSerialNumbersReturn>
        <ns1:mtid>transactionID123456</ns1:mtid>
        <ns1:subId/>
        <ns1:resultCode>0</ns1:resultCode>
        <ns1:errorCode>0</ns1:errorCode>
        <ns1:amount>1.0</ns1:amount>
        <ns1:currency>EUR</ns1:currency>
        <ns1:dispositionState>R</ns1:dispositionState>
        <ns1:serialNumbers/>
      </ns1:getSerialNumbersReturn>
    </ns1:getSerialNumbersResponse>
  </soap:Body>
</soap:Envelope>
```

9.5 executeDebit

After the customer successfully assigns the cards to the disposition, the business partner executes the debit to withdraw the money from the customer's paysafecard.

Example request:

```
<urn:executeDebit>
  <urn:username>USER</urn:username>
    <urn:password>PASSWORD</urn:password>
      <urn:>18b02d230-a6822f-4cbb-aee9-0bc07d90cfa4</urn:>
        <urn:subId></urn:subId>
      <urn:amount>10.00</urn:amount>
    <urn:currency>EUR</urn:currency>
  <urn:close>1</urn:close>
</urn:executeDebit>
```

Example response:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <ns1:executeDebitResponse xmlns:ns1="urn:pscservice">
      <ns1:executeDebitReturn>
        <ns1:>18b02d230-a6822f-4cbb-aee9-0bc07d90cfa4</ns1:>
          <ns1:subId/>
            <ns1:resultCode>0</ns1:resultCode>
            <ns1:errorCode>0</ns1:errorCode>
          </ns1:executeDebitReturn>
        </ns1:executeDebitResponse>
      </soapenv:Body>
  </soapenv:Envelope>
```

10. Error codes

If an error code appears that is not listed here, please contact integration@paysafecard.com

Description of result codes:

Result name	Description
resultCode	0 : successful 1 : logical problem 2 : technical problem
errorCode	Contains an error number if the resultcode is not equal to 0.

The most common error codes are:

2001=Transaction (%1/%2) already exists. Please contact your Webshop.

2017=Transaction (%1/%2) is in invalid %3 status; %4 or %5 was expected.

3001=Merchant %1 is not active. Please contact your Webshop.

3007=Debit attempt after expiry of disposition time window.

3014=Reporting criterion %1 for merchant %2 doesn't exist.

10007= General technical error.

10008= Authentication failed.

10015= Currency not valid for SOPG user.

10028= One of the requested parameters could not be validated.

3017= It is mandatory to send an MCID.

3019= MCID contains invalid values.

11. paysafecard brand guidelines and logos

The paysafecard integration must be done accordingly to the brand guidelines *(https://www.paysafecard.com/fileadmin/Website/Dokumente/B2B/2018-paysafecard-brand-guidelines-partners.pdf).*

The svg paysafecard logos can be accessed via the following endpoints:

- During the test phase: https://customer.test.at.paysafecard.com/rest/payment/logo.svg?mid=MID&submerchant_id=SUBMERCHANT_ID&country=COUNTRY
- Pre-production and live: https://customer.cc.at.paysafecard.com/rest/payment/logo.svg?mid=MID&submerchant_id=SUBMERCHANT_ID&country=COUNTRY

To get the correct logo for the type of integration and country, the following parameters must be correctly specified when calling the endpoint:

Parameter	Description	Format
MID (mandatory)	The Merchant ID is the unique merchant identifier and also defines the currency being used for a transaction.	default value of 10 digits
SUBMERCHANT_ID (mandatory - only if any is set for the MID)	The Reporting Criteria (or submerchant Id) is used to classify sub-merchants. The setup of RCs must be agreed with paysafecard.	max. value of 8 alphanumeric characters
COUNTRY (mandatory)	The target country.	2-digit ISO 3166-1

Example request: https://customer.test.at.paysafecard.com/rest/payment/logo.svg?mid=1090002661&submerchant_id=1&country=AT