Service Oriented Prepaid Gateway (SOPG) Documentation

Table of Contents

Contact:

For technical questions about the implementation please contact

techsupport@paysafecard.com

1. System overview

This Document gives detailed information about the usage and parameters of cash-ticket`s Service Oriented Prepaid Gateway (SOPG).

The Gateway is a "Simple Object Access Protocol (SOAP) XML Web Service" which exposes API Client functionalities that can be used with any SOAP capable client system.

1.1. Prerequisites for using SOPG:

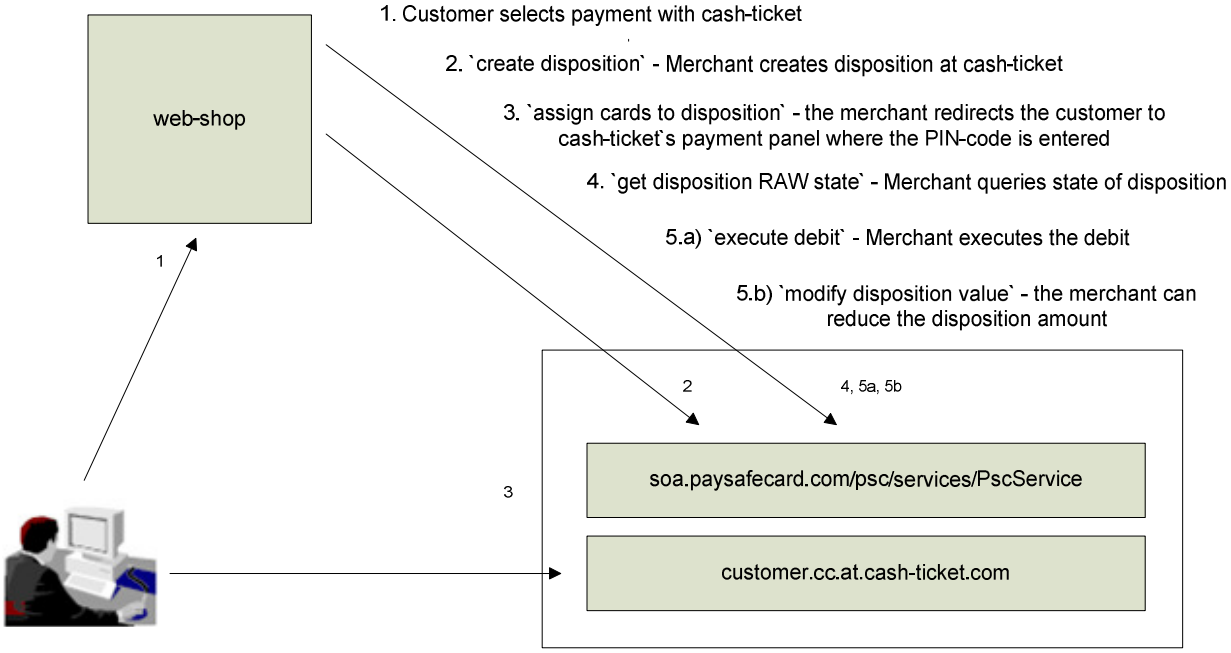
- login credentials (user/password)
- authorization of **your** IP address (if you receive a 403 error when trying to access the service, it is very likely because your IP address is not yet allowed to access)
- Service Endpoint URL
  https://soa.paysafecard.com/psc/services/PscService
- Web Service Definition Language (WSDL) URL
  https://soa.paysafecard.com/psc/services/PscService?wsdl

2. Classic Payment overview

In each payment there are three parties involved: a customer, a web-shop (whom we refer to as 'merchant') and the Cash-Ticket company.

Payments take place in 'payment transactions' or 'dispositions', which are uniquely identified by a 'merchant transaction ID' and hold a value called 'amount' that is typically the amount of money for which a customer buys something

2.1. Life cycle of a normal payment transaction



1. When a customer indicates that they want to buy, this is typically done by activating a button in the merchant's web shop.

2. The merchant now initiates the payment process by sending a 'create disposition' request to Cash-Ticket that creates a disposition at the server. It is the responsibility of the merchant to provide the unique merchant

transaction ID and remember it for future reference. Now that Cash-Ticket knows about the payment, the merchant has to forward the customer to the Cash-Ticket payment site (this is typically done automatically by redirecting the customer's web browser to this site). Along with the forwarding HTTPS request the merchant sends the merchant transaction ID and some other parameters so that Cash-Ticket can associate the incoming customer with the previously created disposition.

3. The customer is now presented with a page where they can enter the PIN (a secret number of a Cash-Ticket) and optionally a password, in case the customer has set a password at the Cash-Ticket Website. Upon completion Cash-Ticket sends the customer back to the merchant's site (the exact URL ("OK-URL") is specified by the merchant during the creation of the payment transaction). The amount shall not be transferred in the OK-URL.

4. The merchant checks the correct payment by the request "get disposition state". After the successful check, the merchant typically thanks the customer for the payment or invites them to shop further. At this stage Cash-Ticket reserves the amount for the merchant, but the amount is not yet captured.

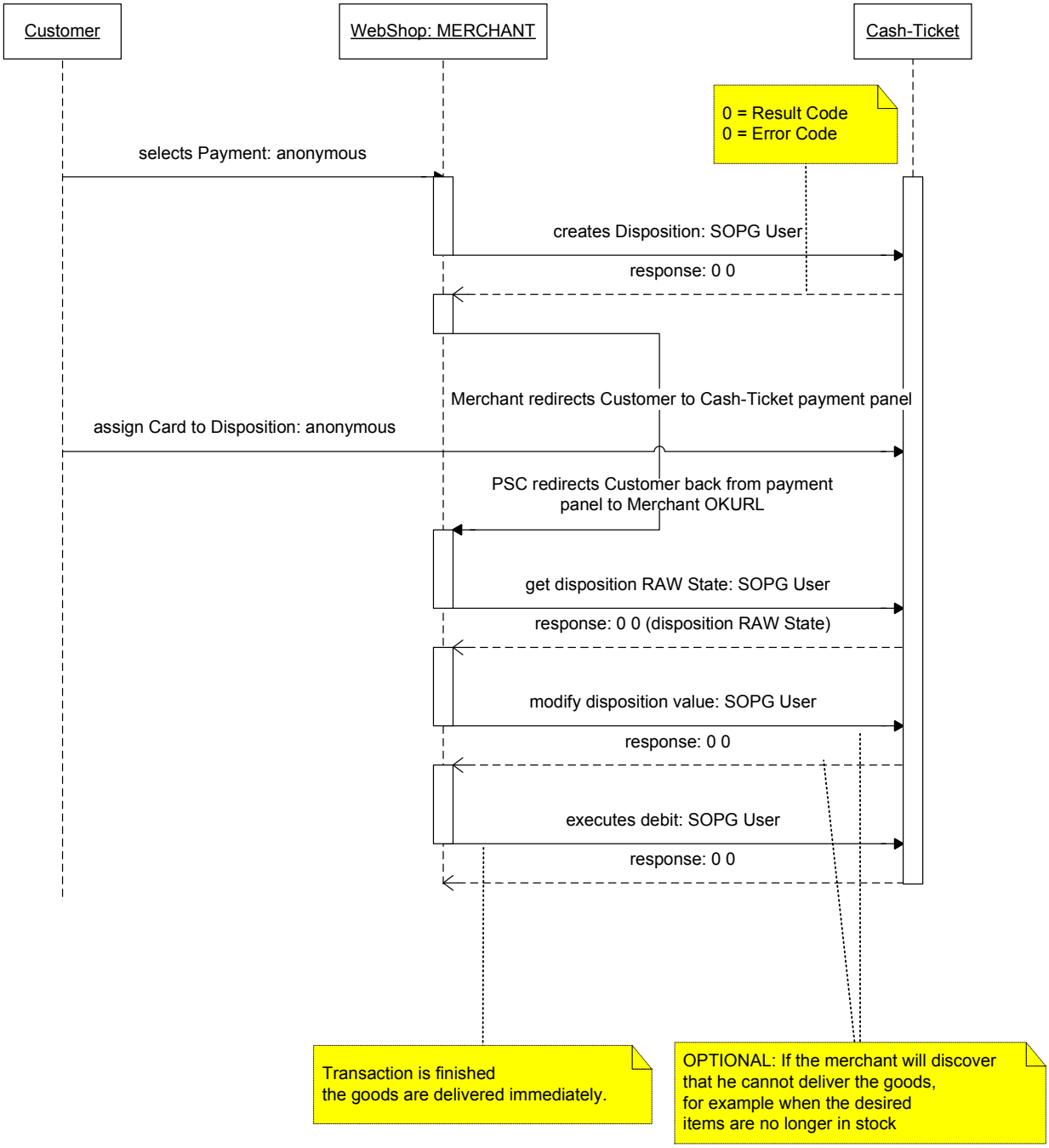
5. The merchant can prepare shipment of the goods. Immediately before shipment the merchant confirms that the payment transaction is completed by issuing an 'execute debit' request to Cash-Ticket. Only after the successful 'execute debit' request Cash-Ticket transfers the money to the merchant. In case the merchant does not call the 'execute' debit request within a certain time period (disposition time[1], which has to be set by Cash-Ticket) the reserved amount will be credited back to the card and can be reused by the customer. The merchant has lost the amount and will never be able to capture it. Unless otherwise noted Cash-Ticket automatically sets the disposition time to 1 hour. Usually the merchant will send the 'execute debit' request with a 'close flag' set to indicate that no more activity will take place for this disposition. Sometimes the merchant may discover that he cannot deliver the goods, for example when the desired items are no longer in stock. In this case the merchant can send a 'modify disposition' request to Cash-Ticket to reduce the amount of the payment transaction or even set it to zero.

---

[1] If not specified on the contract, the disposition time will be set to 1 hour. If needed, it can be changed by Cash-Ticket any time.

The amount, which was up to this time reserved for the transaction, is freed on the customer's card and they can use it for other payments. Another possible scenario is that part of the order can be shipped immediately, while the rest, which for example has to be fetched from a central stock, will take longer. In this case the merchant can decide to ship in two parts and send an 'execute debit' call for only a part of the amount, where the 'close flag' is not set so that Cash-Ticket knows that further debits will take place for the transaction.

The following sequence diagram below shows the sequence followed for payments.



3. SOPG Operations

The complete payment process is handled between the merchant-side system and Cash-Ticket SOPG **except** the redirection of the customer to the payment panel, please check Chapter 5.1.3. for details.

The provided WSDL includes much more Operations, please notice that all required Operations for the basic payment process are described in this document.

3.1. Operations Overview

Operation Name	Request Elements	Response Elements
createDisposition	username, password, mtid, amount, currency, okUrl, nokUrl	mtid, mid, resultCode, errorCode
getDispositionRawState	username, password, mtid, currency	mtid,resultCode, errorCode, currency, amount, dispositionState
getSerialNumbers	username, password, mtid, currency	mtid, resultCode, ErrorCode, currency, amount, dispositionState, serialNumbers
modifyDispositionValue [1]	username, password, mtid, amount, currency	mtid, resultCode, errorCode,
execute debit	username, password, mtid, amount, currency, close	mtid, resultCode, errorCode
getMid [1]	username, password, currency	currency, mid, resultCode, errorCode

[1] Optional Operations

3.2.1 Create Disposition

The merchant initiates the payment process by sending a 'create disposition' request to Cash-Ticket that creates a disposition on the server. The following data must be passed with the request:

- mtid: the 'merchantTransactionID', has to be generated by the merchant and has to be unique for each merchant (typically it includes a customer ID and the timestamp e.g. user321-2006082413246789)
- amount: the value of the goods ordered by the customer; max. 11 digits before the decimal point, exactly two digits after the decimal point
- currency is the merchant currency (3 characters ISO-Code)
- 'OKURL' this is the URL to which the customers are forwarded by Cash-Ticket after they successfully assigned their cards and completed the payment. The merchant may include some information in the URL (e.g. the merchantTransactionID) to get back the context of the payment transaction, e.g. to show a confirmation panel to the customer.
- 'NOKURL' this is the URL to which customers are forwarded by Cash-Ticket when they hit "cancel" on the payment panel.

  After successful 'create disposition' the merchant has to redirect the customer to the Cash-Ticket payment site. This URL has to include 'merchantID', 'merchantTransactionID', 'amount' and 'currency'.

3.2.2 getDispositionRawState

The Return Elements are result code, error code and current Disposition State code

One letter code	Meaning	Description
R	Created	The disposition has successfully been created. If nothing happens within 30 minutes the disposition will be transferred to state "X" by the cleanup job
S	Disposed	Cards have successfully been assigned to the disposition, the merchant can debit, no debits have taken place so far
E	Debited	Partially Debited, the disposition is still open, further debits are possible
O	Consumed	Final debit with closeflag=1 has been called. No further debits are possible
L	Cancelled	The disposition has actively been cancelled by the customer
I	Invalid	This disposition state is currently not in use
X	Expired	The time window for this disposition has ended (either before cards were assigned or before debit calls were made

3.2.3 getSerialNumbers

The 'getSerialNumbers' function is an extension of 'getDispositionRawState' containing result code, error code, disposition-value (current), currency, state and a list of serial numbers and their associated disposition-values.

getSerialNumbers is intended to replace getDispositionRAWstate if the merchant need to check the assigned Serial Numbers before executing the debit (e.g. to prevent usage of multiple "Promotional Campaign PIN`s" for one User Account)

3.2.4 ModifyDispositionValue (optional)

The merchant can reduce the originally disposed amount with the 'modifyDispositionValue' request, e.g. if he cannot ship the goods, he can partially cancel part of the order. The disposition has to be in state disposed or (partially) debited. The input parameters are 'mid', 'mtid', the 'amount' to which the merchant wants to reduce the disposition's value (the difference will be made available on the card(s) used for this reservation) and merchant 'currency'.

3.2.5 Execute Debit

The 'execute debit' function performs the process of withdrawing money from the customer's **Cash-Ticket**. For each disposition <n> debits can be executed. Each debit reduces the open disposition amount. It will differ from merchant to merchant whether the whole disposition amount will be withdrawn by a single debit or several debits will be executed.

The following data must be passed:

- mid: the 'merchantID'
- mtid: the 'merchantTransactionID': unique identifier for each disposition
- 'amount': the debit amount; max. 11 digits before the decimal point, exactly two digits after the decimal point
- 'currency': the merchant currency
- 'closeFlag': indicates if further debits will be executed or not. If the close flag is '1' the disposition will be set to totally consumed and no further debits are possible.

3.2.6 getMID (optional)

Every merchant currency have its own MID assigned.
With getMID the merchant can query the assigned MID for the requested currency.

4. Functional Specification

4.1. Element Details

Element Name	Element Description
username	(SOPG user) provided by Cash-Ticket
password	provided by Cash-Ticket
mid	merchant ID, uniquely identifies merchant, length = 10
mtid	merchant transaction ID, uniquely generated by merchant, max. length = 60 characters (typically it includes a customer ID and the timestamp e.g. user321-2006082413246789)
amount	amount of payment transaction, fixed point decimal with max. 11 digits before comma and exactly 2 digits after comma (e.g. '3200.00'); use a point as decimal separator
currency	ISO code for currency, length 3 - 5 character (e.g. 'EUR')
okurl	OK-URL to which the customer is forwarded by Cash-Ticket after successful payment. OK-URL has to be passed URL encoded.
nokurl	NOT-OK-URL to which customers are forwarded by Cash-Ticket when they hit "cancel" on the payment panel. NOT-OK-URL has to be passed URL encoded
close	close transaction flag, possible values '0' (don't close transaction) or '1' (close transaction, this is the last debit)
subID	Optional for every possible request, can be used for reporting relevant information.
dispositionState	Returns the current statuscode, possible values 'R', 'S', 'E, 'O', 'L', 'I' or 'X'
resultcode	0 : successful, 1 : logical problem, 2 : technical problem
errorcode	Contains an error number if resultcode is not equal to 0.

4.1.1. Encoding of (N)OK-URLs:

URLs have to be passed URL-encoded. The encoding is done in two steps:

- the parameter of the URL have to be converted to standard HTML code, e.g. the character (ä Ä ö Ö ü Ü ß) are replaced by (ä Ä ö Ö ü Ü ß)
- all non-alphanumeric characters in the parameter of the (N)OK-URL have to be replaced by '%HH' (HH is the hex code of the character).

EXAMPLE:
"Danke für die Bezahlung" looks like "Danke für die Bezahlung" in standard HTML and finally has to be passed on as "Danke%20f%26uuml%3br%20die%20Bezahlung".

4.1.2. Usage of locale and language parameter

Basically, the locale parameter uses a combination of language and country code and will replace the language parameter in the long run, because it enables different payment interfaces depending on language AND country.

Currently, the following locales are supported:

de_de, de_at, en_uk, en_us, el_gr, es_es, nl_nl, it_it, fr_fr, pl_pl, pt_pt, si_si, sk_sk, tr_tr

For backward compatibility all existing language parameters still yield the same result as in former versions of the API, but every language will be automatically transformed into a locale. We would therefore suggest to use only the locale parameter.

Supported language parameters:

de, en, gr, el, es, it, fr, nl, pl, pt, si, sk, tr

If more than one country exists for a language, a default country is defined:

en → en_uk

de → de_de

The following rules apply for the usage of one or both parameters:

- If a **valid** locale is sent, it overrides any language parameter
- If no language **AND** no locale is sent, the default is "de_de"
- A **completely invalid** locale like "xy_xy" defaults to "en_uk"
- In case of a **partly invalid** (or not yet supported) locale like "en_xy", the language part will still be used and the result page will be "en_uk"

Examples:

locale	language	payment page displayed in
		de_de (german Germany)
de_at		de_at (german Austria)
en_uk		en_uk (british english)
en_us	de	en_us (american english)
	it	it_it (italian)
xy_xy[1]		en_uk (british english)
xy_xy[1]	el	el_gr (greek)
it_xy[2]		it_it (italian)
it_us[2]		it_it (italian)
	xy[1]	en_uk (british english)

[1] invalid

[2] partly invalid

5. Payment Test-Scenario

The scenario described in the following chapter contains the following steps:

- Create disposition: The merchant creates a disposition, which means the customer initiated the payment process.
- Get Disposition Raw State: The merchant verifies that the reservation has been successfully created and that it has not expired before redirecting to the payment panel.
- Assign card to disposition: One or more cards are assigned to the disposition. This is carried out by the customer. The respective amount is reserved on the card.
- Get Disposition Raw State: The merchant checks whether the reservation on the card has been done successfully.
- A) Debit: A debit is performed by the merchant. The debit represents the actual withdrawal of credit from the card.

  B) Modify disposition: A merchant can reduce the amount of the disposition, e.g. if shipment is not possible (dispositions can only be reduced, not increased)
- Get Disposition Raw State: The merchant verifies the proper status of the disposition after debiting.

5.1.1. Create Disposition

Who:

Merchant

Description:

The merchant creates a disposition by sending the 'createDisposition' request passing all necessary parameters.

Example Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:pscservice">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:createDisposition>
      <urn:username>USER</urn:username>
      <urn:password>PASSWORD</urn:password>
      <urn:mtid>18b02d230-a6822f-4cbb-aee9-0bc07d90cfa4</urn:mtid>
      <urn:amount>10.00</urn:amount>
      <urn:currency>EUR</urn:currency>
      <urn:okUrl>http%3a%2f%2fwww%2eCash-Ticket%2ecom</urn:okUrl>
      <urn:nokUrl>http%3a%2f%2fwww%2efest%2etuwien%2eac%2eat</urn:nokUrl>
    </urn:createDisposition>
  </soapenv:Body>
</soapenv:Envelope>
```

Example Response:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <ns1:createDispositionResponse xmlns:ns1="urn:pscservice">
      <ns1:createDispositionReturn>
        <ns1:mtid>18b02d230-a6822f-4cbb-aee9-0bc07d90cfa4</ns1:mtid>
        <ns1:mid>1000001396</ns1:mid>
        <ns1:resultCode>0</ns1:resultCode>
        <ns1:errorCode>0</ns1:errorCode>
      </ns1:createDispositionReturn>
    </ns1:createDispositionResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

5.1.2. getDispositionRawState

Who:

Merchant

Description:

After having called 'createDisposition' the merchant queries the Cash-Ticket server for the status of the payment to verify it has the expected state "R" (CREATED).

Example Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:pscservice">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:getDispositionRawState>
      <urn:username>USER</urn:username>
      <urn:password>PASSWORD</urn:password>
      <urn:mtid>18b02d230-a6822f-4cbb-aee9-0bc07d90cfa4</urn:mtid>
      <!--Zero or more repetitions:-->
      <urn:subId></urn:subId>
      <urn:currency>EUR</urn:currency>
    </urn:getDispositionRawState>
  </soapenv:Body>
</soapenv:Envelope>
```

Example Response:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <ns1:getDispositionRawStateResponse xmlns:ns1="urn:pscservice">
      <ns1:getDispositionRawStateReturn>
        <ns1:mtid>18b02d230-a6822f-4cbb-aee9-0bc07d90cfa4</ns1:mtid>
        <ns1:subId/>
        <ns1:resultCode>0</ns1:resultCode>
        <ns1:errorCode>0</ns1:errorCode>
        <ns1:currency>EUR</ns1:currency>
        <ns1:amount>10.0</ns1:amount>
        <ns1:dispositionState>R</ns1:dispositionState>
      </ns1:getDispositionRawStateReturn>
    </ns1:getDispositionRawStateResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

5.1.3. Assign Cards to Disposition

Who:

Customer (via Cash-Ticket's payment panel)

Prerequisite:

The command 'create disposition' was successfully executed and returned "0 0". Thus, the customer can be forwarded to the panel for assigning cards to the disposition. The Parameter MID can be used from the create Disposition Response

Description:

One or more cards are assigned to the created disposition. This task is executed by the customer.

If customers click cancel without assigning one or more cards to the disposition, they will be forwarded to the NOKURL.

After successful card assignment customers will be shown the second page of the payment panel where they will be forwarded to the OKURL when clicking on the OK-button on that page.

URL and parameters:

https:// customer.test.at.cash-ticket.com/ctcustomer/GetCustomerPanelServlet

OR

https:// customer.cc.at.cash-ticket.com/ctcustomer/GetCustomerPanelServlet

?mid=1000001396
&mtid=18b02d230-a6822f-4cbb-aee9-0bc07d90cfa4
&amount=10.00
&currency=EUR
&language=de (optional parameter)
&locale=de_at (optional parameter)

Input:

PIN-Code, e.g.: 4725 4983 6548 7393
Password: <default empty>
Terms of Use: <checkbox, default unchecked>

5.1.4. getDispositionRawState

Who:

Merchant

Description:

After the customer has assigned cards to the disposition the merchant queries the Cash-Ticket server for the status of the payment to verify it has the expected state "S" (DISPOSED) before calling the debit function.

Example Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:pscservice">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:getDispositionRawState>
      <urn:username>USER</urn:username>
      <urn:password>PASSWORD</urn:password>
      <urn:mtid>18b02d230-a6822f-4cbb-aee9-0bc07d90cfa4</urn:mtid>
      <!--Zero or more repetitions:-->
      <urn:subId></urn:subId>
      <urn:currency>EUR</urn:currency>
    </urn:getDispositionRawState>
  </soapenv:Body>
</soapenv:Envelope>
```

Example Response:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <ns1:getDispositionRawStateResponse xmlns:ns1="urn:pscservice">
      <ns1:getDispositionRawStateReturn>
        <ns1:mtid>18b02d230-a6822f-4cbb-aee9-0bc07d90cfa4</ns1:mtid>
        <ns1:subId/>
        <ns1:resultCode>0</ns1:resultCode>
        <ns1:errorCode>0</ns1:errorCode>
        <ns1:amount>10.0</ns1:amount>
        <ns1:currency>EUR</ns1:currency>
        <ns1:dispositionState>S</ns1:dispositionState>
      </ns1:getDispositionRawStateReturn>
    </ns1:getDispositionRawStateResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

5.1.5.A) Execute Debit

Who:

Merchant

Prerequisite:

The assignment of the cards to disposition was executed successfully.

Description:

The debiting is the process of withdrawing money from the customer's Cash-Ticket. For each disposition <n> debits can be executed and each debit reduces the open disposition amount. The merchant uses the closeFlag to inform Cash-Ticket, whether further debits are to be expected. After the merchant-specific time window expires, each open disposition is closed - no further debits will be accepted, remaining amounts reserved on the customers card will be made available again. In this example the merchant debits EUR 5,- (and the disposition's remaining value yields EUR 3,-)

Example Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:pscservice">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:executeDebit>
      <urn:username>USER</urn:username>
      <urn:password>PASSWORD</urn:password>
      <urn:mtid>18b02d230-a6822f-4cbb-aee9-0bc07d90cfa4</urn:mtid>
      <!--Zero or more repetitions:-->
      <urn:subId></urn:subId>
      <urn:amount>5.00</urn:amount>
      <urn:currency>EUR</urn:currency>
      <urn:close>1</urn:close>
    </urn:executeDebit>
  </soapenv:Body>
</soapenv:Envelope>
```

Example Response:

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

5.1.6.B) Modify Disposition for Payment Transaction

Who:

Merchant

Description:

At any time after 'create disposition' the merchant can reduce the originally disposed amount, e.g. if they cannot ship the goods. In this example the open amount is reduced to 8.00 EUR and 2.00 EUR are made available on the customer's Cash-Ticket again.

Example Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:pscservice">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:modifyDispositionValue>
      <urn:username>USER</urn:username>
      <urn:password>PASSWORD</urn:password>
      <urn:mtid>18b02d230-a6822f-4cbb-aee9-0bc07d90cfa4</urn:mtid>
      <!--Zero or more repetitions:-->
      <urn:subId></urn:subId>
      <urn:amount>8.00</urn:amount>
      <urn:currency>EUR</urn:currency>
    </urn:modifyDispositionValue>
  </soapenv:Body>
</soapenv:Envelope>
```

Example Response:

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

5.2. Description of results:

Result Name	Result Description
resultcode	0 : successful, 1 : logical problem, 2 : technical problem
errorcode	Contains an error number if resultcode is not equal to 0.

Appendix A: Errorcodes

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

general messages - errors: 0001 - 0600

1=No data selected.
2=%1 is not numeric.
3=Mandatory field %1 is empty.
4=Decimal field with name %1 and value %2 has no decimal point.
5=Decimal field with name %1 and value %2 has no digits before the decimal point.
6=Decimal field with name %1 and value %2 has too many digits before the decimal point (max. %3 allowed).
7=Decimal field with name %1 and value %2 has too few digits after the decimal point (must have %3).
8=Decimal field with name %1 and value %2 has too many digits after the decimal point (max. %3 allowed).
9=Decimal field with name %1 and value %2 is not a number with format N.M (where N are 1 to %3 digits, M are exactly %4 digits and M and N are numeric).
10=Decimal field with name %1 and value %2 has no digits after the decimal point (must have at least 1 and at most %3).
11=Decimal field with name %1 and value %2 must not be negative.
12=Decimal field with name %1 and value %2 is not a number with format N.M (where N are 1 to %3 digits, M are 1 to %4 digits and M and N are numeric).
13=Decimal field with name %1 is empty.
14=Cannot process more than %1 objects per transaction.
15=Answer to Challenge Question is empty.
16=Answer to Challenge Question is wrong.
17=Answer to Challenge Question contains invalid characters.

20=Challenge Question is empty.
21=Challenge Question with value %1 is too long (max. %2 characters are allowed).

25=Distributor ID is empty.
26=Distributor ID with value %1 is too long (max. %2 characters are allowed).

30=Distributor Name is empty.
31=Distributor Name with value %1 is too long (max. %2 characters are allowed).

35=Logistics Company ID is empty.
36=Logistics Company ID with value %1 is too long (max. %2 characters are allowed).

40=Logistics Company Name is empty.
41=Logistics Company Name with value %1 is too long (max. %2 characters are allowed).

45=Merchant Name is empty.
46=Merchant Name with value %1 is too long (max. %2 characters are allowed).

50=Merchant ID is empty.
51=Merchant ID with value %1 is too long (max. %2 characters are allowed).

55=Merchant-transaction ID is empty.
56=Merchant-transaction ID with value %1 is too long (max. %2 characters are allowed).

60='Not-OK' URL is empty.

65='OK' URL is empty.

70=Password is empty.
71=Password too long, max. number of characters 30.
72=Syntax error in password, allowed characters 0-9, a-z, A-Z.
75=Serial number is empty.
76=Serial number with value %1 is too long (max. %2 characters are allowed).
77=Serial number %1 is not numeric.

80=Card State %1 is invalid.
81=Submitted Card State %1 of field %2 is not equal to the expected Card State %3.

85=Card Type %1 is invalid.

90=Debit State %1 is invalid.
95=Disposition State %1 is invalid.
96=Submitted Disposition State %1 of field %2 is not equal to the expected Disposition State %3.

100=Distributor State %1 is invalid.
101=User Type %1 is invalid.
102=Assign action %1 is invalid.
103=Business type %1 is invalid.
104=Feature type %1 is invalid.

105=Import State %1 is invalid.
106=Submitted Import State %1 of field %2 is not equal to the expected Import State %3.
107=Actor Type %1 is invalid.

110=Logistics Company State %1 is invalid.

115=Merchant State %1 is invalid.
116=Submitted Merchant State %1 of field %2 is not equal to the expected Merchant State %3.

120=Close Debit-flag %1 is invalid (must be 0 or 1).

125=Currency is empty.
126=Currency with value %1 has invalid length (must have 3 characters).

130=Actor State %1 is invalid.
131=Submitted Actor State %1 of field %2 is not equal to the expected Actor State %3.

135=Actor Name is empty.
136=Actor Name with value %1 is too long (max. %2 characters are allowed).

140=Currency Name is empty.
141=Currency Name with value %1 is too long (max. %2 characters are allowed).

145=Batch ID Name is empty.
146=Batch ID Name with value %1 is too long (max. %2 characters are allowed).
147=Batch ID %1 is not numeric.

150=Wrong position of delimiter %1 between month and year in date %2 (current position = %3, expected position = %4 or %5).
151=Wrong position of delimiter %1 between day and month in date %2 (current position = %3, expected position = %4 or %5).
152=The year must have 4 digits (you specified %1 as year).
153=Day %1 is not numeric.
154=Month %1 is not numeric.
155=Year %1 is not numeric.
156=Year must be between %1 and %2 (you specified %3).
157=Month must be between %1 and %2 (you specified %3).
158=Day must be between %1 and %2 (you specified %3).
159=A blank character is expected between date and time (you specified %1 as date/time).
160=Wrong position of delimiter %1 between hours and minutes (current position = %2, expected position = %3 or %4).
161=Wrong position of delimiter %1 between minutes and seconds (current position = %2, expected position = %3 or %4).
162=Wrong position of delimiter %1 between seconds and nanoseconds (current position = %2, expected position = %3 or %4).
163=Hours %1 are not numeric.
164=Minutes %1 are not numeric.
165=Seconds %1 are not numeric.
166=Nanoseconds %1 are not numeric.
167=Hours must be between %1 and %2 (you specified %3).
168=Minutes must be between %1 and %2 (you specified %3).
169=Seconds must be between %1 and %2 (you specified %3).
170=Nanoseconds must be between %1 and %2 (you specified %3).
171=Wrong position of delimiter %1 between month and day of date %2 (current position = %3, expected position = %4 or %5).
172=Date %1 is not a valid date in format YYYY-MM-DD.

175=Stock Location is empty.
176=Stock Location with value %1 is too long (max. %2 characters are allowed).

180=Factory is empty.
181=Factory with value %1 is too long (max. %2 characters are allowed).

185=SAP Reference Number is empty.
186=SAP Reference Number with value %1 is too long (max. %2 characters are allowed).

190=Material Number is empty.
191=Material Number with value %1 is too long (max. %2 characters are allowed).

192=Sequence Number is empty.
193=Sequence Number with value %1 is too long (max. %2 characters are allowed).
194=Sequence Number %1 is not numeric.

195=Row Number is empty.
196=Row Number with value %1 is too long (max. %2 characters are allowed).
197=Row Number %1 is not numeric.

200='date from' not specified.
201=Date filter and Transaction ID filter can only be used exclusively.
202=Entered 'date from' is after 'date to'.
203=You must not specify both Currency Code and Exchange Rate.
204=You must not specify 'valid to' without 'valid from'.
205=Unexpected application error - please contact system administrator; error details: %1.
206=The time window (%1 days, %2 hours, %3 minutes) is not valid.
207=The submitted 'valid from' date-time %1 of the submitted Exchange Rate for Currency ID %2 is less than or equal to the 'valid from' date-time of an existing Exchange Rate for this Currency ID.
208=The days must be between 0 and 365 (you specified %1).
209=The hours must be between 0 and 24 (you specified %1).
210=The minutes must be between 0 and 59 (you specified %1).
211=Buying Rate must be larger than Selling Rate.
212=Bad input parameter.
213=State filter and Transaction ID Filter can only be used exclusively.
214=Entered 'date from' %1 must not be a future date (current time %2).
215=The Terms Of Use checkbox is not activated.
216=Reporting Criteria and TransactionID Filter can only be used exclusively.
217=Either both Reporting Criteria or none must be specified.
218=Business Types and TransactionID Filter can only be used exclusively.
219='date to' must not be more than %1 days after 'date from'.
220='date from' and 'date to' must not be empty if no invoice number is specified.
221=Hit list would contain more than %1 rows. Please be more specific!
222=Buying rate must be larger than 0.
223=Selling rate must be larger than 0.
224=Selling rate must be larger or equal than half buying rate.

300=Time interval is empty.
301=Time interval with value %1 is too long (max. %2 characters are allowed).
302=Time interval of field %1 is not numeric (you specified %2).
303=Time interval of field %1 must be greater than 0 (you specified %2).

305=User name is empty.
306=User name is wrong.
307=User name contains invalid characters.
308=User password is empty.
309=User password violates password rules.
310=User password contains invalid characters.
311=Invalid base64 encoded certificate

315=Object id with name %1 is empty.
316=Object id with name %1 and value %2 is not a number.

350=Technical error authenticating user
351=LDAP username must be empty for this actor type

LDAP error messages 400 to 449 reserved

400=Base distinguished name not found for certificate to be added.
401=Certificate distinguished name not found.
402=User not found.
403=Certificate distinguished name cannot be added. It already exists.

404=User cannot be added. It already exists.
405=The password and the confirmation password don't match.
406=Cannot find user %1 to delete from user management.

general messages - success messages 0601 - 0900

601=The command completed successfully.
602=The command completed successfully, no data found.
603=The command completed successfully, more data match filter criteria (change filter criteria to reduce amount of data returned).

card messages - error messages: 1001 - 1600

1001=%1 is not allowed to activate Cards.
1002=%2 is not allowed to have Cards assigned.
1003=Serial Numbers are not continuous.
1004=Card with Serial Number %1 has an unexpected state %2, expected is %3.
1005=Card with Serial Number %1 has not a location "%3", but is at "%2".
1006=Card with Serial Number %1 is not assigned to %2.
1007=Card with Serial Number %1 does not exist.
1008=Access denied.
1009=%1 is not allowed to have Cards assigned.
1010=The state of one or more Cards could not be set to State 'INVALID'.
1011=Currency %1 does not match Card Currency %2.
1012=Card State %1 is not valid for this request, expected Card State is %2.
1013='from-serial number' is greater than 'to-serial number'.
1014=No Password set for this Card.
1015=Access denied because of a repeated recent access violation.
1016=Wrong Password entered.
1017=New Password 1 and new Password 2 are different.
1018=The Card with Serial Number %1 could not be set to State 'INVALID'.
1019=Access violation.
1020=Challenge Question Answer 1 and Challenge Question Answer 2 are different.
1021=No Card with Batch ID %1 could be found.
1022=Invalid Batch Number for card with Serial Number %1.
1023=Duplicate Card found in print file.
1024=Card is missing in print file.
1025=Card is in an invalid State, expected State is 'GENERATED'.
1026=Number of copies printed is invalid, Card is set to State 'INVALID'.
1027=%1 is not allowed to deactivate Cards.
1028=Password could not be set since Challenge Question Answer 1 is different from Challenge Question Answer 2.
1029=You need to specify question, answer and answer verification to set the Challenge Question.
1030=Card with Serial Number %1 has an unexpected Material Number %2, expected is %3.
1031=Load Cards completed with error: %1.
1032=Card Batch %1 not found.
1033=At least one Card Type has to be selected.
1034=No Cards have been selected for this range.
1035=Card State %1 is not valid for this request, expected Card State %2 or %3.
1036=Card State %1 is not valid for this request, expected Card State %2, %3 or %4.
1037=Card in print file does not exist.
1038=Duplicate entry in card list found.
1039=Card with Serial Number %1 has not a location "%3" or "%4", but is at "%2".
1040=Card with Serial Number %1 has already been assigned to **paysafe**card stock %2.
1041=Card with Serial Number %1 has already been assigned to Logistics Company %2.

1042=Card with Serial Number %1 cannot be assigned to a distributor before it has been assigned to a logistics company or PSC stock.
1043=Invalid material number %1 for card with serial number %2, expected: %3
1044=Card with Serial Number %1 already exists.
1045=Card with Random Number %1 already exists.
1046=Card with Random Number %1 has currently no disposition amount available.

card messages - success messages: 1601 - 1900

1601=Password has been set successfully.
1602=Password has been reset successfully.
1603=Password has been changed successfully.
1604=Challenge Question Answer has been set successfully.
1605=Challenge Question answered correctly, Password has been changed.
1606=The Cards have been invalidated successfully.
1607=All cards have been transferred to the Print Shop.
1608=Card(s) %1 to %2 activated successfully.
1609=Card(s) %1 to %2 deactivated successfully.
1610=Load Cards completed successfully: %1.
1611=Load Cards completed with warning: %1.
1612=Receive print file completed successfully: %1.
1613=Password counter has been reset successfully.

card messages - error messages: 1901 - 2000

1901=Card %1 could not be set to state 'INVALID'.
1902=Receive print file failed for card %1 with %2.
1903=Receive print file failed with %1.
1904=Cards with serial number from %1 to %2 have been transferred to printshop.
1905=Cards with serial number from %1 to %2 have been assigned.

payment messages - error messages: 2001 - 2600

2001=Transaction (%1/%2) already exists. Please contact your webshop.
2002=Transaction (%1/%2) does not exist. Please contact your webshop.
2003=Transaction (%1/%2) is in invalid state %3, expected is %4.
2004=Insufficient funds for payment, open amount is %1.
2005=The entered password is incorrect.
2006=Transaction currency %1 is invalid for transaction (%2/%3). Please contact your webshop.
2007=The amount %1 is invalid for the used card.
2008=The amount %1 exceeds the card balance.
2009=The amount %1 is invalid for the transaction (%2/%3). Please contact your webshop.
2010=The amount %1 is insufficiently disposed for the transaction (%2/%3).
2011=The Currency %1 is invalid for this transaction, expected is %2.
2012=Payment transaction failed.
2013=Error finding transaction: %1.
2014=No disposition has been made for this payment transaction.
2015=Payment transaction failed.
2016=Error finding Merchant: %1.
2017=Transaction (%1/%2) is in invalid State %3, expected is %4 or %5.
2018=MicroDebits for transaction (%1/%2) are not sequential.
2019=MicroDebit for transaction (%1/%2) does not exist.
2020=Business type of transaction (%1/%2) is %3. Amount cannot be modified.
2021=The amount %1 is invalid for the transaction (%2/%3). The minimum transaction amount is %4.
2022=Transaction (%1/%2) has no cards assigned.

2023=Business type of transaction (%1/%2) is %3, expected: %4. Please contact your webshop.
2024=Debit cannot be performed, business type of transaction (%1/%2) is %3.
2025=Transaction amount is empty. Please contact your webshop.
2026=Transaction amount %1 is not numeric. Please contact your webshop.
2027=Transaction amount %1 is invalid. Please contact your webshop.
2028=Business type %1 is invalid for transaction.

payment messages - success messages: 2601 - 2900

2601=Payment completed successfully.
2602=Command completed successfully. No transactions found.

payment messages - error messages: 2901 - 3000

2901=Cleanup for open transactions failed: %1.
2902=Cleanup for expired transactions failed for card dispositions: %1.
2903=Cleanup for expired transactions failed for dispositions: %1.
2904=Cleanup for open transactions completed successfully: %1 dispositions expired.
2905=Cleanup for expired transactions completed successfully: expired %1 dispositions and %2 card dispositions.
2906=Debit for expired transaction (%1/%2) completed successfully.

master reference - error message: 3001 - 3600

3001=Merchant %1 is not active. Please contact your webshop.
3002=Currency %1 is not valid for merchant %2. Please contact your webshop.
3003=Merchant %1 does not exist. Please contact your webshop.
3004=Distributor %1 does not exist.
3005=Logistics Company %1 does not exist.
3006=Card Type %1 is not accepted by the merchant.
3007=Merchant %1 exceeded time window to debit the transaction.
3008=Cards cannot be assigned to logistics company %1.
3009=Cards cannot be assigned to distributor %1.
3010=Distributor %1 is not active.
3011=Logistics Company %1 is not active.
3012=Merchant %1 exceeded time window for Micropayment.
3013=Merchant %1 already exists.
3014=Reporting Criterion %1 for Merchant %2 doesn't exist.
3015=Reporting Criterion %1 for Merchant %2 is in state %3, expected %4 or %5.

master reference - success messages: 3601 - 3900

3601=Command completed successfully.
3602=Command completed successfully.
3603=Command completed successfully.

feature messages: 3901 - 4000

3901=Feature with primary key (%1 %2 %3) cannot be found.
3902=The user %1 is not allowed to access the feature %2.

merchant API technical messages - error messages: 4001 - 4600

4001=SSL error.
4002=Invalid function request.
4003= above maximum disposition amount (€ 1000 or equivalent in selected currency).
4004=Invalid proxy request.
4005=Connection error.
4006=Unexpected response from server.
4007=Undefined error - this should not happen.

4008=Error reported from backend.
4010=Error opening configuration file.
4011=Configuration file is no regular readable file.
4012=Incorrect syntax in configuration file.
4013=Incorrect value in configuration file.
4014=Error HTTP response from API proxy: %1.

technical error messages: 5001-5500

5001=General technical error.
5002=MAC check.

technical success messages: 5601-6000

5601=%1 objects of type %2 have been marked for archiving successfully at %3.
5602=%1 rows in table %2 have been deleted successfully at %3.

SOPG specific error codes

10004= General Technical Error
10008= Authentication failed
10015= Currency not valid for SOPG User