Payout

SOPG (ServiceOrientedPrepaidGateway -xmlbasedprotocol) Documentation

Version history

Version	Date	Description	Author
1.0	07-03-2014	1st version	Product
1.1	18-06-2014	Added error code 3198, 3197	Product
1.2	20-11-2014	Added subid parameter to payout function to enable payout for PSP's	Product
1.3	14-01-2015	Added process diagram (3.2)	Product
1.4	02-07-2015	Added 6 business related error-codes	Product
1.5	09-10-2017	Payout Review	Techsupport

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

Table of Content

1. Introduction

This document gives a detailed overview about the usage and parameters of paysafecards Service Oriented Prepaid Gateway (SOPG) for using payout.
This document is an extension to the SOPG Documentation for payment which can be downloaded via the following link API_SPEC_SOPG_CLASSIC.

An example of the implementation is given at the end of the document.

2. Payment, SOPG and paysafecard systems

Information type	Definition	Where to find
Classic paysafecard payment	This document explains the classic paysafecard payment workflow and is necessary to understand the core functionality of paysafecard API.	Document: API_SPEC_SOPG_CLASSIC
SOPG API (SOAP XML Web Service)	This chapter explains the definition of paysafecards API client functionalities.	Document: API_SPEC_SOPG_CLASSIC Chapter: 3
paysafecard systems	This chapter gives an overview of paysafecard environments.	Document: API_SPEC_SOPG_CLASSIC Chapter: 4
Error handling	All possible error codes are listed.	Document: API_SPEC_SOPG_CLASSIC Chapter: 8

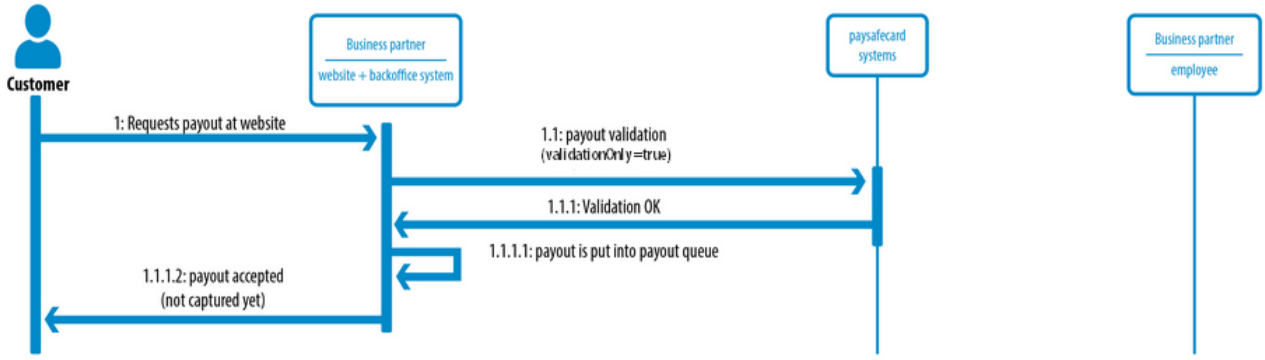
3. Payout process

3.1 Description of payout

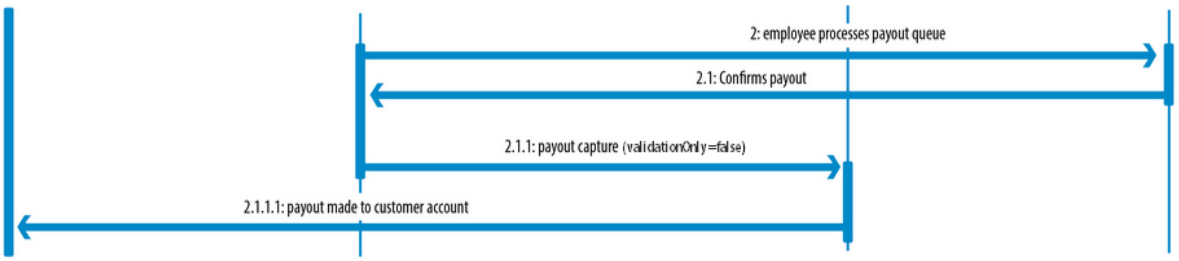
Payout allows the transfer of funds to my paysafecard customers; payout is executed by the business partner on demand of the customer.

3.2 Process diagram

Step 1: Customer requests payout and the request is validated



Step 2: Business partner executes payout



1. On the website of the business partner (merchant), the customer requests a payout from his balance to the customers' paysafecard account.
1.1 The payout request is validated with paysafecard system in real-time (no money is transferred).
1.1.1 paysafecard validates the request (can the payout take place, does the customer exist etc..)
1.1.1.1 the merchant puts the payout request in the back-office queue, to be approved manually later.
1.1.1.2 The customer is informed that the request is accepted and the payout is pending (or is refused because the validation failed).

2. The back-office employee at the business partner manually approves the payout request from the queue.
2.1 The employee confirms the payout (or rejects it).
2.1.1 the business partner executes the payout.
2.1.1.1 paysafecard transfers the money to the customer.

3.3 Exchange of customer data

For each payout request the business partner needs to provide the customer's personal details (first name, last name, date of birth) to paysafecard during the payout call. paysafecard automatically validates the provided data against the registered my paysafecard account data.
The payout will automatically be refused if the data does not match.
If the data does not match 100%; the automatic validation cannot proceed and the payout will be refused automatically. Input will be normalized before the comparison starts by paysafecard.

3.4 Getting the MID limits

The business partner has one MID (for each currency, each MID has its own payout limit - the amount that still can be paid out by the merchant), information about the financial condition of a MID can be retrieved real-time via the SOAP function GetPayoutState.
In case the business partner has multiple MID's, automatically all associated MID's will be returned as well. As soon as a MID limit is reached, payout on this MID is not possible for this time-period.
This information is also available by logging in to the web-interface for business partner (the merchant reporting tool or "MRT"), but will only return the payout state for the current MID and not for associated MID's.

3.5 The payout report

For reconciliation purposes the business partner can download a CSV file containing all possible information for payments and payouts, this file is called the "payout report".
The payout report can be downloaded in the MRT.

3.6 Netting

All payout transaction need to be paid by the business partner. The total monthly payout amount is automatically deducted (netted) from the monthly payment amount.
The total amount of payout money may exceed the total amount of payment money up to a certain level; detailed information on this level can be retrieved with the GetPayoutState function described on page 7.

4. Payout implementation

4.1 Prerequisites

- Login credentials and authorization of payment server IP to paysafecard systems.
- The business partner received a my paysafecard account for testing purposes.

4.2 Implementation checklist & interface guidelines

How the paysafecard payout implementation ideally should look like from a customer perspective is documented in the Implementation checklist & interface guidelines on paysafecard.com.

5. Functions

Operation Name	Type	Description	Request Elements	Response Elements
payout	C	Top-up the customer's my paysafecard account with the given amount and currency.	username [required] password [required] ptid [required] amount [required] currency [required] customerIdType [required] customerId [required] merchantClientId [required] validationOnly [required] utcOffset [required] firstName [required] (subelement of CustomerDetailsBasic) lastName [required] (subelement of CustomerDetailsBasic) dateOfBirth [required] (subelement of CustomerDetailsBasic) subId [required for PSP's]	resultCode, errorCode, errorCodeDescription, mid, ptid
getpayoutState	Q	Retrieves the financial condition of one or multiple MID's.	username [required] password [required]	resultCode, errorCode, errorCodeDescription, mid, currency, totalpayoutAmount totalPaymentAmount, creditLine, totalpayoutBalance, dailypayoutLimit, dailypayoutAmount, dailypayoutBalance

C= Command, Q = Query

6. Parameters

username – custom account username
- provided by paysafecard to business partner for authentication

password – custom account password
- provided by paysafecard to business partner for authentication

ptid – (unique) payout transaction ID
- Must always be unique, also in case of failed
- payouts Max. length: 90 characters
- Must be different from the transaction id's used for
- payment Recommended value: up to 20 characters
- Provided by merchant
- Only the following is allowed A-Z, a-z, 0-9 as well as – (hyphen) and _ (underline)
- Example: 3516-6s4dfsad41

amount – payout amount
- Requested amount is not allowed to exceed 2500.00 EUR (or equivalent in a different transaction currency)
- in value. Max. 11 digits before – exactly 2 digits after
- decimal point Use a point as decimal separator
- Example: 100.00

currency – currency of the payout
- amount Max. Length: 3 characters,
- all uppercase ISO Currency Code
- Example: EUR

customerIdType – used authentication method for identification of the my paysafecard
- account Fixed value: EMAIL

customerId – related value to the
- customerIdType Max. length: 90 characters
- The e-mail address of the customer

merchantclientID – a unique end customer identifier (the unique ID of the end customer as registered at the merchant's database)
- NOTE: for security reasons do not use the customer's registered username, unless
- encrypted. max. length: 50 characters
- example: client123, hashed values, Random customer Identifier
- after a successful Payout, every additional Payout request requires the same merchantclientID and my paysafecard combination

ValidationOnly – Validate the payout without actually transferring the
- funds value true: Test the payout
- value false: to execute payout

utcOffset – the difference in hours and minutes from Coordinated Universal
- Time (UTC) example: -03:00

firstName – (subelement of CustomerDetailsBasic): the first name of the payout
- customer example: John
- max. length. 40 characters

lastName –(subelement of CustomerDetailsBasic): the last name of the payout
- customer example: Do
- max. length. 40 characters

dateOfBirth – (sub-element of CustomerDetailsBasic): the date of birth of the payout customer in YYYY-MM-DD format
- example: 1979-12-20

subId – Mandatory parameter for PSP's (payment service providers), to distinguish multiple websited, paramter must be left empty if nothing else is agreed
- So-called "reporting criteria" offers the possibility to classify
- transactions Max. length: 8 characters (case sensitive)
- Agreement with paysafecard
- needed Example : webshop1

resultCode – indicates the type of
- error 0 indicates that no error occurred
- 1 indicates that there is a problem with the submitted data (e.g., wrong credentials, transaction has expired, etc.)
- 2 (technical problem) means that the service is temporarily not available

errorCode – indicates the error that occurred, see "complete list of error codes" at the bottom of this
- document 0 indicates that no error occurred
- Any other value indicates an error occurred

errorCodeDescription – provides a more detailed description of the
- errorcode NULL in case of no error
- In case of an error a string value will be returned

MID – merchant ID, unique ID of the merchant/currency
- pair 10 digits long

totalpayoutAmount – The total amount of payouts that were executed on this MID within the current billing cycle
- Double value
- Example: 50000,00

totalPaymentAmount – The total amount of payments that were executed on this MID within the current billing cycle in the currency of the MID
- Double value
- Example : 300000,00

creditLine – An extra payout credit line provided by paysafecard to support payout in case an MID is out of balance in the currency of the MID
- Double value
- Example 5000,00

totalpayoutBalance – The total amount that still can be paid out this billing cycle in the currency of
- the MID Double value
- 10000,00

dailypayoutLimit – the maximum amount of money that can be paid out on this MID for the current day (24 hour), in the currency of the MID (only if configured by paysafecard)
- Double value if configured
- Example : 0,00

dailypayoutAmount – the total amount that has been paid out on this MID today in the currency of
- the MID Double value
- Example : 5000,00

dailypayoutBalance – the total amount that still can be paid out on this MID today in the currency of
- the MID Double value if the dailypayoutlimit is configured
- Example : 0,00

7. Example payout

In this chapter, a test scenario is presented using example data. In practice, it will differ from business partner to business partner whether the function payout and GetPayoutState is executed once or multiple times.

> Do not use the enclosed example data! Each business partner will receive a consistent set of test data for testing purposes.

7.1 GetPayoutState

The business partner requests the financial state of a pay-out MID before executing a payout, to determine how much money is still available for payout.

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:pscservice">
<soapenv:Header/>
<soapenv:Body>
<urn:getpayoutState>
 <urn:username>GetLimits</urn:username>
 <urn:password>1d8b5gw</urn:password>
</urn:getpayoutState>
</soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns1="urn:pscservice">
<soap:Body>
 <ns1:getpayoutStateResponse>
 <ns1:getpayoutStateReturn>
 <ns1:resultCode>0</ns1:resultCode>
 <ns1:errorCode>0</ns1:errorCode>
 <ns1:errorCodeDescription/>
 <ns1:payoutState>
  <ns1:mid>3332222221</ns1:mid>
  <ns1:currency>GBP</ns1:currency>
  <ns1:totalpayoutAmount>50.0</ns1:totalpayoutAmount>
  <ns1:totalPaymentAmount>100.0</ns1:totalPaymentAmount>
  <ns1:creditLine>148.0</ns1:creditLine>
  <ns1:totalpayoutBalance>198.0</ns1:totalpayoutBalance>
  <ns1:dailypayoutLimit>1300.0</ns1:dailypayoutLimit>
  <ns1:dailypayoutAmount>50.0</ns1:dailypayoutAmount>
  <ns1:dailypayoutBalance>1250.0</ns1:dailypayoutBalance>
 </ns1:payoutState>
</ns1:getpayoutStateReturn>
</ns1:getpayoutStateResponse>
</soap:Body>
</soap:Envelope>
```

7.2 Payout validation (precheck)

With this function the complete payout process is simulated without actually transferring the funds into the customer's account.
The validationOnly parameter should be set to true.

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:pscservice">
<soapenv:Header/>
<soapenv:Body>
<urn:payout>
  <urn:username>SOAPUI_DV</urn:username>
  <urn:password>1d8b5gw</urn:password>
  <urn:ptid>SOPG_2014-02-20_13-30-59</urn:ptid>
  <urn:amount>0.03</urn:amount>
  <urn:currency>EUR</urn:currency>
  <urn:customerIdType>EMAIL</urn:customerIdType>
  <urn:customerId>ftmoISsATO@ESntgeHuho.xGx</urn:customerId>
  <urn:merchantClientId>soapUI</urn: merchantClientId>
  <urn:validationOnly>true</urn:validationOnly>
  <urn:utcOffset>+01:00</urn:utcOffset>
  <urn:customerDetailsBasic>
            <urn:firstName>Test</urn:firstName>
            <urn:lastName>Tester</urn:lastName>
            <urn:dateOfBirth>1961-01-06</urn:dateOfBirth>
  </urn:customerDetailsBasic>
</urn:payout>
</soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns1="urn:pscservice">
<soap:Body>
<ns1:payoutResponse>
<ns1:payoutReturn>
 <ns1:ptid>SOPG_2014-02-20_12-19-43</ns1:ptid>
 <ns1:requestedCurrency>EUR</ns1:requestedCurrency>
 <ns1:requestedAmount>0.03</ns1:requestedAmount>
 <ns1:validationOnly>false</ns1:validationOnly>
<ns1:resultCode>0</ns1:resultCode>
<ns1:errorCode>0</ns1:errorCode>
<ns1:errorCodeDescription></ns1:errorCodeDescription>
</ns1:payoutReturn>
</ns1:payoutResponse>
</soap:Body>
</soap:Envelope>
```

7.3 Payout execution

With this function the payout is executed and money is transferred to the customer's account. Technically the function is similar to the payout validation, with only one parameter change; the validationOnly parameter should be set to false.

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:pscservice">
<soapenv:Header/>
<soapenv:Body>
<urn:payout>
  <urn:username>SOAPUI_DV</urn:username>
  <urn:password>1d8b5gw</urn:password>
  <urn:ptid>SOPG_2014-02-20_13-30-59</urn:ptid>
  <urn:amount>0.03</urn:amount>
  <urn:currency>EUR</urn:currency>
  <urn:customerIdType>EMAIL</urn:customerIdType>
  <urn:customerId>ftmoISsATO@ESntgeHuho.xGx</urn:customerId>
  <urn:merchantClientId>soapUI</urn: merchantClientId>
  <urn:validationOnly>false</urn:validationOnly>
  <urn:utcOffset>+01:00</urn:utcOffset>
  <urn:customerDetailsBasic>
          <urn:firstName>Test</urn:firstName>
          <urn:lastName>Tester</urn:lastName>
          <urn:dateOfBirth>1961-01-06</urn:dateOfBirth>
  </urn:customerDetailsBasic>
</urn:payout>
</soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns1="urn:pscservice">
<soap:Body>
<ns1:payoutResponse>
<ns1:payoutReturn>
 <ns1:ptid>SOPG_2014-02-20_12-19-43</ns1:ptid>
 <ns1:requestedCurrency>EUR</ns1:requestedCurrency>
 <ns1:requestedAmount>0.03</ns1:requestedAmount>
 <ns1:validationOnly>false</ns1:validationOnly>
<ns1:resultCode>0</ns1:resultCode>
<ns1:errorCode>0</ns1:errorCode>
<ns1:errorCodeDescription></ns1:errorCodeDescription>
</ns1:payoutReturn>
</ns1:payoutResponse>
</soap:Body>
</soap:Envelope>
```

8. Complete list of error codes

In case of the error codes below, the business partner should display an informational message to the customer according to the interface guidelines available here:
https://www.paysafecard.com/business/downloads/payout-guidelines/

10007 = General technical error
3100 = Not available
3105 = Product not allowed
3150 = Missing mandatory parameter
3152 = Distributor missing
3153 = Invalid card type
3155 = Terminal is blocked
3161 = Merchant not allowed to perform this action
3162 = my paysafecard account not found by provided credentials
3164 = Duplicate payout request
3165 = Invalid amount
3166 = Merchant limit reached
3167 = Customer balance exceeded
3168 = payout rejected - account registration not completed
3169 = Payout id collides with existing disposition id
3170 = Top-up limit exceeded
3171 = Payout amount is below minimum payout amount of the merchant
3183 = No unload merchant configuration for this country
3193 = Customer not active
3194 = Customer yearly payout limit exceeded.
3195 = Customer account details do not match.
3197 = There is already the maximum number of pay-out accounts assigned to this merchantClient
3198 = There is already the maximum number of pay-out merchant clients assigned to this account.
3199 = Payout blocked due to security reasons
33000 = Reporting Criteria Limit Reached
33001 = Reporting Criteria not allowed to perform this Action

Other errors can be communicated to the customer as “general technical error”. In general, when one of these errors occur the business partner should contact paysafecard immediately via integration@paysafecard.com to solve the issue.

9. Other documentation

SOPG documentation	https://www.paysafecard.com/fileadmin/Website/Dokumente/B2B/paysafecard_classic_payment_api_en.pdf
Implementation checklist	https://www.paysafecard.com/fileadmin/Website/Dokumente/B2B/payout-implementation-checklist.PDF
Marketing related material (logos)	https://www.paysafecard.com/en/business/support/downloads/paysafecard-logos/
Implementation checklist	Provided by email