

## Table of Content

<b>1. General Information</b>	<b>2</b>
1.1 Prerequisites	2
1.2 Area of application	2
<b>2. Important information about the functions</b>	<b>2</b>
2.1 Validation	2
2.2 Auto-correct	2
2.3 Debug	2
<b>3. Functions, parameters and return values</b>	<b>2</b>
3.1 <b>construct</b> (status, debugStatus, sysLang, autoCorrect)	2
3.2 <b>merchant</b> (username, password)	3
3.3 <b>confirmMerchantData</b> (currency)	3
3.4 <b>setCustomer</b> (amount, currency, mtid, merchantClientId)	3
3.5 <b>setShopId</b> (shopId)	3
3.6 <b>setShopLabel</b> (shopLabel)	3
3.7 <b>setRestrictedCountry</b> (country)	4
3.8 <b>setMinAge</b> (age)	4
3.9 <b>setMinKycLevel</b> (level)	4
3.10 <b>setUrl</b> (ok_url, nok_url, pn_url)	4
3.11 <b>createDisposition</b>	4
3.12 <b>getSerialNumbers</b> (mtid, currency, subId)	5
3.13 <b>executeDebit</b> (amount, close)	5
3.14 <b>Log.getLog</b> (type)	5
<b>4. Other functions</b>	<b>6</b>
4.1 <b>setAmount</b> (amount)	6
4.2 <b>setMerchantClientId</b> (merchantClientId)	6
4.3 <b>setCurrency</b> (currency)	6
4.4 <b>setMtid</b> (mtid)	6
4.5 <b>setClose</b> (close)	7
4.6 <b>setStatus</b> (status)	7
4.7 <b>getcustomerPanel</b>	7
4.8 <b>validate</b> (type,value)	7
4.9 <b>Log.addLog</b> (msg, call, call_params, result, type)	7
4.10 <b>Debug.addDebug</b> (key,value)	8
4.11 <b>Languages.getVariable</b> (Language, varName)	8
4.12 <b>reset</b>	8
4.13 <b>autoSet</b>	8
<b>5. Changing the language output</b>	<b>8</b>
<b>6. Contact &amp; support</b>	<b>8</b>

## 1. General Information

This documentation describes the scope of functionality and use of the Software Development Kit (SDK) for the Service Oriented Prepaid Gateway (SOPG) from paysafecard.

### 1.1 Prerequisites

- JAVA 7.0+
- Servlet container

### 1.2 Area of application

The SOPG-SDK is a collection of functions for automations and validations with extensive debugs and logs. The development was conceived of in such a way as to make it possible to integrate the SOPG from paysafecard quickly, easily and with low susceptibility to errors.

## 2. Important information about the functions

### 2.1 Validation

The validation used in the SDK should not be used as a component of the code. For this reason, the function is declared „private“. The primary purpose of validation in the SDK is error localization. It is no substitute for internal parameter validation.

### 2.2 Auto-correct

Auto-correct is no substitute for internal parameter formatting. Instead it is intended as a safety mechanism. Automatic corrections appear in the log as warnings and are deactivated by default.

### 2.3 Debug

If debugging is activated, every action executed in the SDK is logged. At the end the developer has an array, sorted chronologically, which contains all the actions.

## 3. Functions, parameters and return values

### 3.1 construct (status, debugStatus, sysLang, autoCorrect)

Parameter	Mandatory	Value	Data type	Description
status	Yes	live / test	String	paysafecard system environment
debugStatus	Yes	TRUE / FALSE	Boolean	Debug status
sysLang	Yes	EN/DE	String	System language
autoCorrect	Yes	TRUE / FALSE	Boolean	Auto correct status

Return value: 0

### 3.2 merchant (username, password)

Sets the username and password.

Parameter	Mandatory	Value
username	Yes	Username
password	Yes	Password

Return value: 0

### 3.3 confirmMerchantData (currency)

Checks whether an MID is defined for the indicated currency. Checks the access data at the same time.

Parameter	Mandatory	Value
currency	Yes	ISO 4217 currency code (3 characters)

Return value:

Status	Return value
Successful	true
Error	false

### 3.4 setCustomer (amount, currency, mtid, merchantClientId)

Sets the mandatory parameters for a createDisposition (create transaction).

Parameter	Mandatory	Value
amount	Yes	valid amount with two decimal places
currency	Yes	ISO 4217 currency code (3 characters)
mtid	Yes	unique transaction ID
merchantClientId	Yes	Unique endcustomer ID (e-mail, customer reference number, etc)

Return value: 0

### 3.5 setShopId (shopId)

Sets the shop ID.

Parameter	Mandatory	Value
shopId	Yes	ID of webshop

Return value: 0

### 3.6 setShopLabel (shopLabel)

Sets the shop label.

Parameter	Mandatory	Value
shopLabel	Yes	label (brand name) of webshop

Return value: 0

### 3.7 setRestrictedCountry (country)

Sets country restrictions. Parameter country is not allowed to be an array. To set multiple countries, the function will need to be called multiple times.

Parameter	Mandatory	Value
country	Yes	ISO 3166-1 country code (2 characters)

Return value: 0

### 3.8 setMinAge (age)

Sets the minimum age.

Parameter	Mandatory	Value	Data type
age	Yes	Positive number	String

Return value: 0

### 3.9 setMinKycLevel (level)

Sets the status the „my paysafecard account“ must have before it can be used for payment.

Parameter	Mandatory	Value
level	Yes	SIMPLE/FULL

Return value: 0

### 3.10 setUrl (ok\_url, nok\_url, pn\_url)

Sets the URLs for the forwarders and payment notification URL.

Parameter	Mandatory	Value
ok_url	Yes	valid URL
nok_url	Yes	valid URL
pn_url	Yes	valid URL

Return value: 0

### 3.11 createDisposition ()

Creates a transaction. This function does not take any parameters as these are ought to have been assigned already. If the transaction is created successfully, getCustomerPanel is called and the URL is generated for the payment transaction.

Return value:

Status	Return value
Successful	CustomerPanel-URL via getCustomerPanel
Error	(String) false

### 3.12 getSerialNumbers (mtid, currency, subId)

Checks the current status of a transaction. The transaction is defined by the mtid.

Parameter	Mandatory	Value
mtid	Yes	mtid for a valid transaction
currency	Yes	ISO 4217 currency code (3 characters)
subId	Yes	Empty / SubId

Return value:

Status	Return value	Data type
Debit	execute*	String
OK	true	String
Error	false	String

\* Note: a debit is only allowed to be executed with executeDebit when the return value is ,execute'.  
A message must be output in the event of an OK or an error.

### 3.13 executeDebit (amount, close)

Debits the amount ,amount'. Close 1 indicates that this was the last debit.

Parameter	Mandatory	Value	Data type
amount	Yes	valid amount to two decimal places	String
close	Yes	0 = further debits 1 = last debit	String

Return value:

Status	Return value
Successful	true
Error	false

### 3.14 Log.getLog (type)

Issues the log in the selected system language. If the system language is not available, the log is issued in English.

Parameter	Mandatory	Value
type	Yes	info / warning / error

Return value: (ArrayList<HashMap<String, String>>)

The arraylist contains a hashmap for each log entry with the following keys:

Status	Return value
msg	Log message
action	Function
action_params	Parameter in the function
result	Return value

**Note that error & warning is for development or for a log only. Only the „info“ type is allowed to be issued to the client!!**

## 4. Other functions

The functions named under 3 are the basic functions for the simple integration of the SOPG. The SDK has extra functions that are automatically used by the SDK in the right place.

### 4.1 setAmount (amount)

Checks (revises) and sets the amount.

Parameter	Mandatory	Value	Data type
amount	Yes	valid amount to two decimal places	String

Return value: 0

### 4.2 setMerchantClientId (merchantClientId)

Checks and sets the merchantClientId.

Parameter	Mandatory	Value	Data type
merchantClientId	Yes	Unique client endcustomer ID (e-mail, customer reference number, etc)	String

Return value: 0

### 4.3 setCurrency (currency)

Checks and sets the currency.

Parameter	Mandatory	Value
currency	Yes	ISO 4217 currency code (3 characters)

Return value: 0

### 4.4 setMtid (mtid)

Checks and sets the mtid.

Parameter	Mandatory	Value
mtid	Yes	unique transaction ID

Return value: 0

#### 4.5 setClose (close)

Checks and sets the status ,close'.

Parameter	Mandatory	Value	Data type
close	Yes	0 = further debits 1 = last debit	String

Return value: 0

#### 4.6 setStatus (status)

Defines the paysafecard system environment for the call.

Parameter	Mandatory	Value
status	Yes	test / live

Return value: 0

#### 4.7 getcustomerPanel ()

Takes the set data and generates the customerPanelUrl where a client is then able to make payment. No parameters are passed as these were already set by createDisposition.

Return value:

Status	Return value
Always	CustomerPanelUrl

#### 4.8 validate (type,value)

Checks ,value' against the pattern ,type'.

Parameter	Mandatory	Value
type	Yes	username, password, amount, merchantClientId, currency, shopId, ShopLabel, mtid, subId, close, nok_url, ok_url, pn_url, minAge, MinKycLevel, restrictedCountry
value	Yes	The value to be checked against the Type pattern.

Return value:

Status	Return value
Successful	true
Error	false

#### 4.9 Log.addLog (msg, call, call\_params, result, type)

Sets the passed values to the specified log level ,type'.

Parameter	Mandatory	Value
msg	Yes	A valid log message
call	Yes	Name of the function or action
call_params	No	Parameter for function or action
result	No	Result of the action
type	Yes	error / warning / info

Return value: 0

#### 4.10 Debug.addDebug (key,value)

Sets the debug value ,value' with key ,key'.

Parameter	Mandatory	Value
key	Yes	Name for debug value
value	Yes	Value for debug

Return value: 0

#### 4.11 Languages.getVariable (Language, varName)

The system uses the variable name, which is passed via varName, to search for a corresponding translation in the language denoted by Language. If this variable does not exist, the system then searches for the variable in English. If it still cannot find the variable, „ERROR: Unknown error.“ is returned. If, however, this language does not exist, the system returns „ERROR: Language does not exist.“.

Parameter	Mandatory	Value
Language	Yes	Language
VarName	Yes	Name of the variable

Return value:

Status	Return value
Successful	Error message
Error	Error message as in the description

#### 4.12 reset ()

Sets each possible variable to prevent a NullPointerException.

Return value: 0

#### 4.13 autoSet ()

Sets the default parameters.

Return value: 0

## 5. Changing the language output

German and English already exist for the language outputs. It is possible to extend the SDK with various numbers of other languages. To prevent errors, we recommend copying a complete language array and renaming the language (de/en). Do not change the language EN as this language is used by the SDK when there is an error in the sysLang.

## 6. Contact & support

For technical integration questions and support please contact us directly at [techsupport@paysafecard.com](mailto:techsupport@paysafecard.com)