

## 1. Contents

2.	General Information	2
3.	Important information about the functions	2
3.1	Validation	2
3.2	Auto-correct	2
3.3	Debug	2
4.	Functions, parameters and return values	3
4.1	<b>construct</b> (debugStatus, sysLang, autoCorrect, status)	3
4.2	<b>merchant</b> (username, password)	3
4.3	<b>confirmMerchantData</b> (currency)	3
4.4	<b>setCustomer</b> (amount, currency, mtid, merchantClientId)	4
4.5	<b>setShopId</b> (shopId)	4
4.6	<b>setShopLabel</b> (shopLabel)	4
4.7	<b>setRestrictedCountry</b> (country)	4
4.8	<b>setMinAge</b> (age)	4
4.9	<b>setMinKycLevel</b> (level)	5
4.10	<b>setUrl</b> (ok_url, nok_url, pn_url)	5
4.11	<b>createDisposition</b> ()	5
4.12	<b>getSerialNumbers</b> (mtid, currency, subId)	6
4.13	<b>executeDebit</b> (amount, close)	6
4.14	<b>getLog</b> (type)	7
5.	Other functions	7
5.1	<b>setAmount</b> (amount)	7
5.2	<b>setMerchantClientId</b> (merchantClientId)	7
5.3	<b>setCurrency</b> (currency)	8
5.4	<b>setMtid</b> (mtid)	8
5.5	<b>setClose</b> (close)	8
5.6	<b>setStatus</b> (status)	8
5.7	<b>getcustomerPanel</b> ()	9
5.8	<b>validate</b> (type,value)	9
5.9	<b>addLog</b> (msg_code, call, call_params, result, type)	9
5.10	<b>addDebug</b> (key,value)	10
5.11	<b>speaking</b> (key)	10
5.12	<b>reset</b> ()	10
5.13	<b>autoSet</b> ()	10
5.14	<b>clientConnect</b> ()	10
6.	Changing the language output	11
7.	Contact & support	11

## **2. 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.

### **2.1 Prerequisites**

- PHP 5.0+
- php\_soap.dll
- php\_openssl.dll

### **2.2 Area of application**

The SOPG-SDK is a collection of functions for automations and validations with extensive debugging and logs. The development was conceived of in such a way as to make it possible to integrate the SOPG from paysafeguard easily, quickly and with a low susceptibility to errors. Automatic corrections appear in the log as warnings and are deactivated by default.

## **3. Important information about the functions**

### **3.1 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.

### **3.2 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.

### **3.3 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.

## 4. Functions, parameters and return values

### 4.1 **construct** (debugStatus, sysLang, autoCorrect, status)

Parameter	Mandatory	Value	Default	Description
<b>debugStatus</b>	Yes	TRUE / FALSE	FALSE	paysafecard system environment
<b>sysLang</b>	Yes	EN / DE	EN	Debug status
<b>autoCorrect</b>	Yes	TRUE / FALSE	FALSE	System language
<b>status</b>	Yes	live / test	test	Auto correct status

Return value: 0

### 4.2 **merchant** (username, password)

Sets the username and password.

Parameter	Mandatory	Value
<b>username</b>	Yes	Username
<b>password</b>	Yes	Password

Return value: 0

### 4.3 **confirmMerchantData** (currency)

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

Parameter	Mandatory	Value
<b>currency</b>	Yes	EN: ISO 4217 currency code (3 characters)

Return value:

Status	Return value
<b>Successful</b>	true
<b>Error</b>	false

#### 4.4 **setCustomer** (amount, currency, mtid, merchantClientId)

Sets the mandatory parameters for a createDisposition (open transaction)

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

Return value: 0

#### 4.5 **setShopId** (shopId)

Sets the shop ID.

Parameter	Mandatory	Value
<b>shopId</b>	Yes	ID of the webshop

Return value: 0

#### 4.6 **setShopLabel** (shopLabel)

Sets the shop label

Parameter	Mandatory	Value
<b>shopLabel</b>	Yes	label (brand name) of webshop

Return value: 0

#### 4.7 **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
<b>country</b>	Yes	ISO 3166-1 country code (2 characters)

Return value: 0

#### 4.8 **setMinAge** (age)

Sets the minimum age.

Parameter	Mandatory	Value
<b>age</b>	Yes	Positive number

Return value: 0

#### 4.9 **setMinKycLevel** (level)

Sets the status the "my paysafecard account" must have before it can be used for a payment

Parameter	Mandatory	Value
level	Yes	SIMPLE/FULL

Return value: 0

#### 4.10 **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

#### 4.11 **createDisposition** ()

Initiate/create a transaction. This function does not take any parameters as these are should have been assigned previously. If the transaction is initiated successfully, getCustomerPanel is called and the URL is generated for the payment transaction.

Return value:

Status	Return value
Successful	CustomerPanel-URL via getCustomerPanel
Error	false

#### 4.12 **getSerialNumbers** (mtid, currency, subId)

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

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

Return value:

Status	Return value
Debit	execute*
OK	true
Error	false

**\* 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.**

#### 4.13 **executeDebit** (amount, close)

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

Parameter	Mandatory	Value
amount	Yes	valid amount with two decimal places
close	Yes	0 = further debits 1 = last debit

Return value:

Status	Return value
Successful	true
Error	false

#### 4.14 **getLog** (type)

Outputs the log in the selected system language. When the system language is not available, the log is output in English.

Parameter	Mandatory	Value
<b>type</b>	Yes	info / warning / error

Return value: (Array)

Status	Return value
<b>msg_code</b>	Code for log message
<b>msg</b>	Log message
<b>action</b>	Function
<b>action_params</b>	Parameters in the function
<b>result</b>	Return value

**Note that error & warning is for development or for a log only Only the \$log[info][msg] is allowed to be issued to the client!!**

### 5. **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.

#### 5.1 **setAmount** (amount)

Checks (revises) and sets the amount.

Parameter	Mandatory	Value
<b>amount</b>	Yes	valid amount with two decimal places

Return value: 0

#### 5.2 **setMerchantClientId** (merchantClientId)

Checks and sets the merchantClientId.

Parameter	Mandatory	Value
<b>merchantClientId</b>	Yes	Unique client ID

Return value: 0

### 5.3 **setCurrency** (currency)

Checks and sets the currency.

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

Return value: 0

### 5.4 **setMtid** (mtid)

Checks and sets the mtid.

Parameter	Mandatory	Value
mtid	Yes	unique transaction ID

Return value: 0

### 5.5 **setClose** (close)

Checks and sets the status 'close'.

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

Return value: 0

### 5.6 **setStatus** (status)

Defines the paysafecard system environment for the call.

Parameter	Mandatory	Value
status	Yes	test / live

Return value:

Status	Return value
Successful	true
Error	false



### 5.7 **getcustomerPanel** ()

Takes the set data and generates the customerPanelUrl where the client can use to make payment.  
No parameters are passed as these were already set by createDisposition.

Return value:

Status	Return value
Always	CustomerPanelUrl

### 5.8 **validate** (type,value)

Checks 'value' against the pattern 'type'

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

Return value:

Status	Return value
Successful	true
Error	false

### 5.9 **addLog** (msg\_code, call, call\_params, result, type)

Sets the passed values to the provided log level 'type'

Parameter	Mandatory	Value
<b>msg_code</b>	Yes	A valid msg_code
<b>call</b>	Yes	Name of the function or action
<b>call_params</b>	No	Parameter for function or action
<b>result</b>	No	Result of the action
<b>type</b>	Yes	error / warning / info

Return value: 0

### 5.10 **addDebug** (key,value)

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

Parameter	Mandatory	Value
<b>key</b>	Yes	Label for debug value
<b>value</b>	Yes	Value for debug

Return value: 0

### 5.11 **speaking** (key)

The msg\_code passed in key is searched and output in the language defined in the sysLang. If the msg\_code does not exist in sysLang, the system will search in the English language. If msg\_code is entered in none of the two languages, an unknown error message is issued with information about the key.

Parameter	Mandatory	Value
<b>key</b>	Yes	Code of the message

Return value:

Status	Return value
<b>Successful</b>	Error message in sysLang or EN
<b>Error</b>	General error message in EN

### 5.12 **reset** ()

Sets each possible variable to prevent a PHP error message.

Return value: 0

### 5.13 **autoSet** ()

Sets the default parameters

Return value: 0

### 5.14 **clientConnect** ()

Class call from SoapClient. Establishes a SOAP connection.

Return value: 0

## **6. 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 an entire 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.

## **7. Contact & support**

For technical integration questions and support please contact us directly at  
techsupport@paysafecard.com