1 Contents

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

- NET 2.0+

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.

3 Important information about the functions

3.1 Validation

The validation used in the SDK should not be used as a component of the programming. 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 and are deactivated by default.

3.3 Debug

If debugging is activated, every action executed in the SDK is logged. At the end there is a generic list, sorted chronologically, containing all actions.

3.4 Creating a transaction/initialisation

The class SOPGClassicMerchantClient is a central function used to process all individual transactions. Before creating a transaction, an instance of the TransactionVals structure must be passed one time via SOPGClassicMerchantClient.setTransactionValues(TransactionVals vals). This structure contains all the important data for payment processing and must be parameterised with the corresponding entries before being passed.

3.5 Logging event methods/exception handling

As well as the collected querying of all logged messages via SOPGClassicMerchantClient.getLogMessages, the developer can also choose to subscribe to logging events via SOPGClassicMerchantClient.logMessageEvent.

4 Functions, parameters and return values

4.1 **SOPGClassicMerchantClient**(bool DebugStatus, string SysLang, bool AutoCorrect = false, DevStatus Status = DevStatus.LIVE)

Parameter	Mandatory	Value	Default	Description
DebugStatus	Yes	TRUE / FALSE	FALSE	Debug status
SysLang	Yes	EN / DE	EN	System language
AutoCorrect	No	TRUE / FALSE	FALSE	Auto correct status
Status	No	live / test	live	paysafecard system environment

Return value: 0

4.2 public struct TransactionVals

Parameter	Mandatory	Value	Default
username	Yes	Username	
password	Yes	Password	
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)	
shopId	Yes	ID of webshop	
shopLabel	Yes	label (brand name) of webshop	
country	Yes	ISO 3166-1 country code (2 characters)	
age	Yes	Positive number	
level	Yes	SIMPLE / FULL	
ok_url	Yes	valid URL	
nok_url	Yes	valid URL	
pn_url	Yes	valid URL	
close	Yes	0 = further debits 1 = last debit	
status	Yes	test / live	

4.3 **confirmMerchantData** (string currency = "EUR")

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:

GetMidReturn Object.

4.4 **createDisposition** ( )

Creates a transaction. This function does not take any parameters as these are should have been assigned previously. If the transaction is created successfully, the URL is generated for the payment transaction.

Return value:

CreateDispositionReturn object.

4.5 **getSerialNumbers** ( )

Checks the current status of a transaction. The transaction is defined by the mtid. This function does not take any parameters as these are should have been assigned previously.

Return value:

GetSerialNumbersReturn object.

* 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.

4.6 **executeDebit** ( )

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

Return value:

ExecuteDebitReturn object.

4.7 **getLog** (Logging.Logger.LogType)

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: (Array)

Status	Return value
logType	Logtype
message	Logmessage
callerFunc	Function
callParams	Parameters in the function
requResult	Return value

Note that error & warning is for development or for a log only. Only the getLog(Logtype.INFO) is allowed to be issued to the client!!

4.8 **public string getcustomerPanelUrl** ( )

Uses the set data to generate the customerPanelUrl which the client can use to pay. No parameters are passed as these were already set by createDisposition.

Return value:

Status	Return value
Always	CustomerPanel URL

5 Changing the language output

German and English already exist for the language outputs. However, it is possible to extend the SDK with any number 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.

6 Contact & support

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