

Inhaltsverzeichnis

1. Allgemeine Information	2
1.1 Voraussetzungen	2
1.2 Einsatzbereich	2
2. Wichtige Funktionsinformationen	2
2.1 Validierung	2
2.2 Automatische Korrektur	2
2.3 Debug	2
2.4 Transaktionserstellung / Initialisierung	2
2.5 Logging-Eventmethoden / Exception-Handling	2
3. Funktionen, Parameter und Rückgaben	3
3.1 SOPGClassicMerchantClient	3
3.2 public struct TransactionVals	3
3.3 confirmMerchantData (string currency = „EUR“)	3
3.4 createDisposition	4
3.5 getSerialNumbers	4
3.6 executeDebit	4
3.7 getLog (Logging.Logger.LogType)	4
3.8 public string getCustomerPanelUrl	4
4. Änderung der Sprachausgabe	5
5. Kontakt & Support	5

1. Allgemeine Information

Diese Dokumentation beschreibt den Funktionsumfang und den Einsatz der Software Development Kit (SDK) für das Service Oriented Prepaid Gateway (SOPG) von paysafecard.

1.1 Voraussetzungen

- NET 2.0+

1.2 Einsatzbereich

Das SOPG-SDK ist eine Sammlung von Funktionen zur Automatisierung und Validierung mit umfangreichem Debug und Log. Die Entwicklung wurde so gestaltet, dass es einfach, schnell und mit einer geringen Fehleranfälligkeit möglich ist, das SOPG von paysafecard zu integrieren.

2. Wichtige Funktionsinformationen

2.1 Validierung

Die in der SDK eingesetzte Validierung sollte nicht als Bestandteil der Programmierung verwendet werden. Aus diesem Grund wurde die Funktion auch als „private“ deklariert. Die Validierung in der SDK dient in erster Linie zur Fehlerlokalisierung und ist kein Ersatz für eine eigene Parameter-Validierung.

2.2 Automatische Korrektur

Der Einsatz der automatischen Korrektur ist kein Ersatz für eine eigene Parameterformatierung, sondern soll als Absicherung dienen. Automatische Korrekturen werden im Log als Warning angezeigt. Die automatische Korrektur ist standardmäßig deaktiviert.

2.3 Debug

Bei aktivem Debug wird jede Aktion, die vom SDK ausgeführt wird, protokolliert. Am Ende steht dann eine generische Auflistung zur Verfügung, die chronologisch sortiert, alle Aktionen enthält.

2.4 Transaktionserstellung / Initialisierung

Als zentrale Funktion zur Abwicklung aller einzelnen Transaktionen steht die Klasse `SOPGClassicMerchantClient` zur Verfügung. Vor Ausführung einer Transaktion muss per `SOPGClassicMerchantClient.setTransactionValues(TransactionVals vals)` einmalig eine Instanz der `TransactionVals`-Struktur übergeben werden. Diese Struktur enthält alle wichtigen Daten zur Zahlungsabwicklung und muss vor der Übergabe mit den entsprechenden Angaben parametrisiert werden.

2.5 Logging-Eventmethoden / Exception-Handling

Neben der gesammelten Abfrage aller geloggtten Nachrichten per `SOPGClassicMerchantClient.getLog` steht dem Entwickler auch die Möglichkeit offen, Logging-Events per `SOPGClassicMerchantClient.logMessageEvent` zu abonnieren.

3. Funktionen, Parameter und Rückgaben

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

Parameter	Pflicht	Optionen	Standard	Beschreibung
DebugStatus	Ja	TRUE / FALSE	FALSE	Status des Debug Modus
SysLang	Ja	EN / DE	EN	Sprache des Systems
AutoCorrect	Nein	TRUE / FALSE	FALSE	Automatische Korrektur
DevStatus	Nein	live / test	live	paysafecard Systemumgebung

Rückgabe: 0

3.2 public struct TransactionVals

Parameter	Pflicht	Optionen
username	Ja	Benutzername
password	Ja	Passwort
amount	Ja	positiver Betrag mit 2 Dezimalstellen
currency	Ja	ISO 4217 Währungscode (3-stellig)
mtid	Ja	Einmalige Transaktions-ID
merchantClientId	Ja	Eindeutige ID des Endkunden (E-Mail, Kdnr, etc)
shopId	Ja	ID des Webshops
shopLabel	Ja	Marken(Name) des Webshops
country	Ja	ISO 3166-1 Ländercode (2-stellig)
age	Ja	Positive Zahl
level	Ja	SIMPLE / FULL
ok_url	Ja	gültige URL
nok_url	Ja	gültige URL
pn_url	Ja	gültige URL
close	Ja	0 = weitere Abbuchungen 1 = letzte Abbuchung
status	Ja	test / live

3.3 confirmMerchantData (string currency = „EUR“)

Prüft ob für die angegebene Währung eine MID hinterlegt ist. Somit werden auch die Zugangsdaten geprüft.

Parameter	Pflicht	Optionen
currency	Ja	ISO 4217 Währungscode (3-stellig)

Rückgabe:
GetMidReturn-Objekt.

3.4 createDisposition

Erstellt eine Transaktion. Diese Funktion benötigt keine übergebenen Parameter, da diese zuvor schon zugewiesen werden müssen. Bei erfolgreicher Transaktionserstellung/Initialisierung wird die URL für den Bezahlvorgang generiert.

Rückgabe:

CreateDispositionReturn-Objekt.

3.5 getSerialNumbers

Prüft den aktuellen Status einer Transaktion. Die Transaktion wird durch die mtid definiert. Diese Funktion benötigt keine übergebenen Parameter, da diese zuvor schon zugewiesen werden müssen.

Rückgabe:

GetSerialNumbersReturn-Objekt.

* Bitte beachten Sie, dass nur bei der Rückgabe von ‚execute‘ eine Abbuchung mit executeDebit ausgeführt werden darf. Bei OK oder Fehler muss eine Mitteilung ausgegeben werden.

3.6 executeDebit

Bucht den in TransactionVals angegebenen Betrag ab. Close 1 zeigt an, dass diese die letzte Abbuchung war.

Rückgabe:

ExecuteDebitReturn-Objekt.

3.7 getLog (Logging.Logger.LogType)

Gibt den Log in der gewählten Systemsprache aus. Ist die Systemsprache nicht verfügbar, wird der Log in Englisch ausgegeben.

Parameter	Pflicht	Optionen
type	Ja	info / warning / error

Rückgabe: (Array an LogMessage-Objekten)

Status	Rückgabe
logType	Logtyp
message	Logmeldung
callerFunc	Funktion
callParams	Parameter in der Funktion
requResult	Rückgabe

Bitte beachten Sie, dass error & warning nur für die Entwicklung oder für ein Protokoll verwendet werden dürfen. Dem Kunden darf nur getLog(LogType.INFO) ausgegeben werden!!

3.8 public string getCustomerPanelUrl

Generiert aus den gesetzten Daten die customerPanelUrl, mit der ein Kunde dann bezahlen kann. Es werden keine Parameter übergeben, da diese zuvor schon durch createDisposition gesetzt wurden.

Rückgabe:

Status	Rückgabe
Immer	CustomerPanel-URL

4. Änderung der Sprachausgabe

Für Sprachausgaben ist Deutsch und Englisch bereits vorhanden. Die SDK kann aber ohne Probleme um beliebig viele Sprachen erweitert werden. Um Fehler zu vermeiden, ist es am besten, wenn ein ganzes Spracharray kopiert und die Sprache (de/en) umbenannt wird. Die Sprache EN sollte nicht verändert werden, da diese vom SDK verwendet wird, wenn es in der sysLang zu einem Fehler kommt.

5. Kontakt & Support

Für technische Anfragen bitte direkt an techsupport@paysafecard.com wenden.