Inhaltsverzeichnis

1. Allgemeine Information

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

1.1 Voraussetzungen

- JAVA 7.0+
- Servlet container

1.2 Einsatzbereich

Das SOPG-SDK ist eine Sammlung von Funktionen zur Automatisierung und Validierung mit umfangreichem Debug und Log. Die Entwicklung wurde so gestalltet, 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 ein Array zur Verfügung, das chronologisch sortiert, alle Aktionen enthält.

3. Funktionen, Parameter und Rückgaben

3.1 construct (status, debugStatus, sysLang, autoCorrect)

Parameter	Pflicht	Optionen	Datentyp	Beschreibung
status	Ja	live / test	String	paysafecard Systemumgebung
debugStatus	Ja	TRUE / FALSE	Boolean	Status des Debug Modus
sysLang	Ja	EN/DE	String	Sprache des Systems
autoCorrect	Ja	TRUE / FALSE	Boolean	Automatische Korrektur

Rückgabe: 0

3.2 merchant (username, password)

Setzt den Benutzernamen und das Passwort.

Parameter	Pflicht	Optionen
username	Ja	Benutzername
password	Ja	Passwort

Rückgabe: 0

3.3 confirmMerchantData (currency)

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:

Status	Rückgabe
Erfolgreich	true
Fehler	false

3.4 setCustomer (amount, currency, mtid, merchantClientId)

Setzt die Pflichtparameter für ein createDisposition (Transaktionserstellung/Initialisierung).

Parameter	Pflicht	Optionen
amount	Ja	Zahl mit 2 Dezimalstellen
currency	Ja	ISO 4217 Währungscode (3-stellig)
mtid	Ja	Einmalige Transaktions-ID
merchantClientId	Ja	Einmalige ID des Endkunden (E-Mail, Kdnr, etc)

Rückgabe: 0

3.5 setShopId (shopId)

Setzt die ID des Shops.

Parameter	Pflicht	Optionen
shopId	Ja	ID des Webshops

Rückgabe: 0

3.6 setShopLabel (shopLabel)

Setzt die Bezeichnunhg für den Shop.

Parameter	Pflicht	Optionen
shopLabel	Ja	Marken(Name) des Webshops

Rückgabe: 0

3.7 setRestricedCountry (country)

Setzt Länderbeschränkungen. Parameter country darf kein array sein.
Um mehrere Länder zu setzen, muss die Funktion mehrmals aufgerufen werden.

Parameter	Pflicht	Optionen
country	Ja	ISO 3166-1 Ländercode (2-stellig)

Rückgabe: 0

3.8 setMinAge (age)

Setzt das Mindestalter.

Parameter	Pflicht	Optionen	Datentyp
age	Ja	Positive Zahl	String

Rückgabe: 0

3.9 setMinKycLevel (level)

Setzt den Status, den der „my paysafecard account“ haben muss, um bezahlen zu können.

Parameter	Pflicht	Optionen
level	Ja	SIMPLE/FULL

Rückgabe: 0

3.10 setUrl (ok_url, nok_url, pn_url)

Setzt die URL´s für die Weiterleitungen und die Payment-Notice-URL.

Parameter	Pflicht	Optionen
ok_url	Ja	gültige URL
nok_url	Ja	gültige URL
pn_url	Ja	gültige URL

Rückgabe: 0

3.11 createDisposition

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

Rückgabe:

Status	Rückgabe
Erfolgreich	CustomerPanel-URL durch getCustomerPanel
Fehler	(String) false

3.12 getSerialNumbers (mtid, currency, subId)

Prüft den aktuellen Status einer Transaktion. Die Transaktion wird durch die mtid definiert.

Parameter	Pflicht	Optionen
mtid	Ja	mtid einer gültigen Transaktion
currency	Ja	ISO 4217 Währungscode (3-stellig)
subId	Ja	Leer / SubId

Rückgabe:

Status	Rückgabe	Datentyp
Abbuchung	execute*	String
OK	true	String
Fehler	false	String

* 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.

3.13 executeDebit (amount, close)

Bucht den Betrag ‚amount' ab. Close 1 zeigt an, dass diese die letzte Abbuchung war.

Parameter	Pflicht	Optionen	Datentyp
amount	Ja	gültiger Betrag mit zwei Dezimalstellen	String
close	Ja	0 = weitere Abbuchungen 1 = letzte Abbuchung	String

Rückgabe:

Status	Rückgabe
Erfolgreich	true
Fehler	false

3.14 Log.getLog (type)

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

Die ArrayList enthält für jeden Logeintrag eine HashMap mit den folgenden Keys:

Status	Rückgabe
msg	Logmeldung
action	Funktion
action_params	Parameter in der Funktion
result	Rückgabe

Bitte beachten Sie, dass error & warning nur für die Entwicklung oder für ein Protokoll verwendet werden dürfen. Dem Kunden darf nur type „info“ ausgegeben werden!!

4. Weitere Funktionen

Die unter Punkt 3 genannten Funktionen sind die Grundfunktionen zur einfachen Integration der SOPG. Die SDK verfügt über weitere Funktionen, die von der SDK automatisch an der richtigen Stelle verwendet werden.

4.1 setAmount (amount)

Prüft (überarbeitet) und setzt einen Betrag.

Parameter	Pflicht	Optionen	Datentyp
amount	Ja	gültiger Betrag mit zwei Dezimalstellen	String

Rückgabe: 0

4.2 setMerchantClientId (merchantClientId)

Prüft und setzt die merchantClientId.

Parameter	Pflicht	Optionen	Datentyp
merchantClientId	Ja	Eindeutige ID des Endkunden (E-Mail, Kdnr, etc)	String

Rückgabe: 0

4.3 setCurrency (currency)

Prüft und setzt die Währung.

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

Rückgabe: 0

4.4 setMtid (mtid)

Prüft und setzt die mtid.

Parameter	Plicht	Optionen
mtid	Ja	Einmalige Transaktions-ID

Rückgabe: 0

4.5 setClose (close)

Prüft und setzt den Status ‚close'.

Parameter	Pflicht	Optionen	Datentyp
close	Ja	0 = weitere Abbuchungen 1 = letzte Abbuchung	String

Rückgabe: 0

4.6 setStatus (status)

Definiert den Status für den Aufruf.

Parameter	Pflicht	Optionen
status	Ja	test / live

Rckgabe: 0

4.7 getcustomerPanel

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

Status	Rückgabe
Immer	CustomerPanelUrl

4.8 validate (type,value)

Prüft ‚value' nach dem Muster ‚type'.

Parameter	Pflicht	Optionen
type	Ja	username, password, amount, merchantClientId, currency, shopId, ShopLabel, mtid, subId, close, nok_url, ok_url, pn_url, minAge, MinKycLevel, restricedCountry
value	Ja	Der Wert, der nach dem Muster Type geprüft werden soll.

Rückgabe:

Status	Rückgabe
Erfolgreich	true
Fehler	false

4.9 Log.addLog (msg, call, call_params, result, type)

Setzt die übergebenen Werte auf dem angegebenem Loglevel ‚type'.

Parameter	Pflicht	Optionen
msg	Ja	Eine gültige Log-Nachricht
call	Ja	Name der Funktion oder Aktion
call_params	Nein	Parameter für Funktion oder Aktion
result	Nein	Ergebnis der Aktion
type	Ja	error / warning / info

Rückgabe: 0

4.10 Debug.addDebug (key,value)

Setzt den Debugwert ‚value' mit Key ‚key'.

Parameter	Pflicht	Optionen
key	Ja	Bezeichnung für Debugwert
value	Ja	Wert für Debug

Rückgabe: 0

4.11 Languages.getVariable (Language, varName)

Mittels dem über varName übergebenem Variablenname, wird in der mit Language angegebenen Sprache, nach einer entsprechenden Übersetzung gesucht. Sollte die Variable nicht vorhanden sein, wird die Variable in der Englischen Sprache gesucht, ist sie dort auch nicht vorhanden, wird „ERROR: Unknown error." zurückgegeben, sollte aber die Sprache auch nicht existieren, wird „ERROR: Language does not exist." zurückgegeben.

Parameter	Pflicht	Optionen
Language	Ja	Sprache
VarName	Ja	Name der Variable

Rückgabe:

Status	Rückgabe
Erfolgreich	Fehlermeldung
Fehler	Fehlermeldung wie i. d. Beschreibung

4.12 reset

Setzt jede mögliche Variable um NullpointerException zu vermeiden.
Rückgabe: 0

4.13 autoSet

Setzt Standardparameter.
Rückgabe: 0

5. 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.

6. Kontakt & Support

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