



Payment Services

Authorization Interface



Spezifikation

Version 4.1.4

Inhaltsverzeichnis

1	Einleitung	4
1.1	Zusammenfassung	4
1.2	Voraussetzungen	4
1.3	Datensicherheit und PCI DSS	4
1.4	Unterstützte Zahlungsmittel	5
1.5	Formatangaben	5
2	Saferpay Client Library	6
2.1	Installation	6
2.1.1	.NET Client Library	6
2.1.2	Java Client Library	6
2.2	Schlüsselerzeugung	6
2.2.1	.NET Client Library	6
2.2.2	Java Client Library	7
2.3	IP-Zugriff konfigurieren	7
3	Klassen und Methoden der Client Library	8
3.1	Zusammenfassung	8
3.1.1	Zahlungsanfrage und Autorisationsantwort	8
3.1.2	Buchung, Stornierung und Tagesabschluss	8
3.2	MessageFactory Klasse	8
3.3	MessageObject Klasse	8
3.4	Open() Methode	8
3.5	Execute() Methode	9
3.6	CreateRequest() Methode	9
3.7	SetAttribute() Methode	9
3.8	GetAttribute() Methode	9
3.9	Capture() Methode	9
4	Saferpay https Interface	10
4.1	IP-Zugriff und Passwort für das Saferpay https Interface	10
4.2	https Interface Adressen	10
4.3	Übertragung von Nachrichten	11
5	Verarbeitungsschritte	12
5.1	Übersicht	12
5.2	Prozessbeschreibung	12
6	Parameter	14
6.1	Autorisationsanfrage	14
6.2	Autorisationsantwort	16
6.3	PayComplete-Anfrage	18
6.4	PayComplete-Antwort	19
7	Saferpay Testkonto	20
8	Beispiele	21
8.1	Wichtiger Hinweis	21
8.2	C# mit der .NET LIB	21
8.3	Java mit der Java LIB	23
8.4	Kommandozeilenaufufe mit der Java LIB	25
8.5	https Interface	27



9	RESULT-Werte	28
10	Kontakt	29
10.1	Saferpay Integration Team	29
10.2	Saferpay Support Team	29

1 Einleitung

1.1 Zusammenfassung

Mit dem Saferpay Authorization Interface, in der Folge auch nur AI genannt, können Online-Zahlungen im Hintergrund abgewickelt werden. Das AI ist geeignet für Shop-Systeme, Callcenter-Lösungen, Warenwirtschafts-, ERP- und CRM-Systeme. Dieses Dokument beschreibt die Integration des AI in bestehende Systeme mit der Saferpay Client Library und dem Saferpay https Interface. Zur Vereinfachung werden in diesem Dokument die Saferpay Client Library auch mit LIB und das Saferpay https Interface auch mit HI abgekürzt.

1.2 Voraussetzungen

Die Nutzung des AI setzt Folgendes voraus:

- Eine entsprechende Lizenz und somit das Vorhandensein einer gültigen Kennung mit Benutzername und Passwort für das Saferpay System.
- Mindestens ein aktives Saferpay Terminal, über das die Zahlungen durchgeführt werden können ist vorhanden und die dazugehörige Saferpay TERMINALID beziehungsweise die Saferpay ACCOUNTID liegt vor.
- Ein gültiger Akzeptanzvertrag für Kreditkarten oder ein anderes Zahlungsmittel liegt vor.

Um das Saferpay HI nutzen zu können, müssen die Händlerdaten auf dem HI eingerichtet werden. Nach der Einrichtung werden die für den Händler erzeugten Zertifikate zur SSL-verschlüsselten Kommunikation dort von Saferpay bereit gestellt. Die Einrichtung ist kostenfrei, muss jedoch für jeden Saferpay Business Account beantragt werden. Senden Sie dazu bitte eine formlose Email mit der Bitte um Einrichtung an DMP-MSCD-all@six-group.com, wenn Sie Ihren Saferpay-Vertrag in der Schweiz abgeschlossen haben und an service@saferpay.com, wenn Sie Ihren Vertrag in einem anderen Land (D, NL, A, und so weiter) abgeschlossen haben.

Wichtig! Bitte vergessen Sie nicht den Kollegen Ihre Saferpay Kundennummer und die IP-Adresse(n) für den Zugriff auf das https Interface mitzuteilen.

1.3 Datensicherheit und PCI DSS

Die Kreditkartenorganisationen haben das Sicherheitsprogramm PCI DSS (Payment Card Industry Data Security Standard) ins Leben gerufen, um Betrug mit Kreditkarten und deren Missbrauch vorzubeugen.

Bitte beachten Sie bei der Gestaltung des Zahlungsprozesses und dem Einsatz des Saferpay AI die PCI DSS Richtlinien. Zusammen mit dem optionalen Dienst Saferpay Secure Card Data können Sie die Zahlungsprozesse so sicher gestalten, dass keine Kreditkartennummern auf Ihren (Web)Servern verarbeitet, weitergeleitet oder gespeichert werden. Das Risiko eines Missbrauchs der Kreditkartendaten wird dadurch reduziert und eine aufwendige PCI DSS Zertifizierung für den Händler entfällt. Für weitere Informationen zu Saferpay Secure Card Data können Sie uns gerne kontaktieren.

Fragen zu PCI DSS kann Ihnen Ihr Verarbeiter oder ein darauf spezialisiertes Unternehmen beantworten (siehe https://www.pcisecuritystandards.org/pdfs/pci_pa-dss_list.pdf).

1.4 Unterstützte Zahlungsmittel

Mit dem Saferpay Authorization Interface können aktuell folgende Zahlungsmittel verarbeitet werden:

- Visa
- MasterCard
- Maestro international
- V PAY
- American Express
- Diners Club
- J.C.B.
- Union Card
- Elektronisches Lastschriftverfahren ELV (nur Deutschland)
- Laser Card

Alle anderen Zahlungsmittel, die eine Dateneingabe auf Seiten des Zahlungsmittelanbieters erfordern, können über die Saferpay Payment Page verarbeitet werden. Für weitere Informationen wenden Sie sich bitte an integration@saferpay.com.

1.5 Formatangaben

Folgende Abkürzungen für die Formatangaben werden in diesem Dokument verwendet:

- a Buchstaben (a - z, A - Z)
- n numerische Zeichen (0 - 9)
- an alphanumerische Zeichen (a - z, A - Z, 0 - 9)
- s Sonderzeichen (- : ; / \ < > . =)
- ans alphanumerische und Sonderzeichen

2 Saferpay Client Library

Die Saferpay LIB wird auf dem Server, der die Applikation des Händlers bereitstellt installiert. Nach der Installation stehen auf dem Server die Saferpay Klassen und Methoden zur Verfügung.

Für die Installation der LIB und die damit mögliche Schlüsselerzeugung werden auf dem Ziel-Server Root- beziehungsweise Administratorrechte benötigt!

Die LIB gibt es als .NET- oder Java-Version. Die entsprechenden Dateien können im Download-Bereich des Saferpay Backoffices unter folgenden Adressen heruntergeladen werden:

.NET LIB: https://www.saferpay.com/download/saferpay_dotnet.exe

Java LIB: https://www.saferpay.com/download/saferpay_java.zip

Kann weder die .NET LIB, noch die Java LIB genutzt werden oder keine lokale Installation möglich ist, steht alternativ das Saferpay https Interface zur Verfügung.

2.1 Installation

2.1.1 .NET Client Library

Starten Sie die geladene Installationsdatei „saferpay_dotnet.exe“ und folgen Sie den Anweisungen des Setup-Assistenten.

2.1.2 Java Client Library

Entpacken Sie die geladene Zip-Datei „saferpay_java.zip“ und kopieren Sie für die Integration in Java das darin enthaltene „Saferpay.jar“ in das Verzeichnis jre/lib/ext.

Für die Integration in einer anderen Programmier- oder Skriptsprache kann das „Saferpay.jar“ in ein beliebiges Verzeichnis kopiert werden.

2.2 Schlüsselerzeugung

Neben der SSL-verschlüsselten Kommunikation zwischen der LIB und den Saferpay Servern werden die Daten eines Saferpay Accounts mit einer digitalen Signatur nach dem PGP-Verfahren (Pretty Good Privacy) versehen und geschützt. Für diesen Zweck muss für jeden Saferpay Account ein Schlüsselpaar erzeugt und auf dem Händler-Server gespeichert werden. Für die Erzeugung der Schlüssel werden ein gültiger Login und das Passwort für das Saferpay Backoffice benötigt. Nach erfolgreicher Schlüsselerzeugung kann das Passwort im Saferpay Backoffice geändert werden, da die Erzeugung der Schlüssel, wenn einmal erfolgt, nicht wiederholt werden muss. Erzeugte Schlüsselpaare behalten Ihre Gültigkeit und sollten daher sicher verwahrt und der Zugriff kontrolliert werden.

2.2.1 .NET Client Library

Nach Installation der .NET LIB steht für die Schlüsselerzeugung eine Grafische Benutzeroberfläche zur Verfügung. Diese findet sich unter:



Der Saferpay Client Setup erscheint. Folgen Sie anschließend den Anweisungen.

2.2.2 Java Client Library

Die Schlüsselerzeugung mit der Java LIB erfolgt über die Kommandozeile. Wechseln Sie hierfür in das Verzeichnis mit dem Saferpay.jar und geben Sie folgenden Befehl ein:

```
java -jar Saferpay.jar -conf -p . -r  
https://www.saferpay.com/user/setup.asp -u e99867001 -w XAjc3Kna
```

Das Beispiel verwendet die Zugangsdaten des Saferpay Testkontos.

Die Kommandozeilenhilfe erscheint nach diesem Aufruf:

```
java -jar Saferpay.jar -h
```

2.3 IP-Zugriff konfigurieren

Obwohl die Kommunikation mit Saferpay bereits verschlüsselt erfolgt, empfiehlt sich als zusätzliche Sicherheitsmaßnahme den Zugriff über die Client LIB auf den Saferpay Account einzuschränken. Hierfür kann der IP-Zugriff über den Punkt "IP-Freigabe" im Saferpay Backoffice (<https://www.saferpay.com/user/login.asp>) konfiguriert werden.

Freigegebene IP-Adressen

Start-IP	Ende-IP	
192.168.5.26	192.168.5.31	löschen
<input type="text"/>	<input type="text"/>	<input type="button" value="hinzufügen"/>

Nach der Hinterlegung einer oder mehrerer IP-Adressen ist der Zugriff auf den Saferpay Account auf diese beschränkt. Anfragen über andere IP-Adressen werden von Saferpay geblockt.

3 Klassen und Methoden der Client Library

Im Folgenden werden Klassen und Methoden beschrieben, die für die Integration der Saferpay Client LIB zur Verfügung stehen.

3.1 Zusammenfassung

3.1.1 Zahlungsanfrage und Autorisationsantwort

Die Zahlungsanfrage wird mit der CreateRequest() Methode erzeugt. Das erzeugte MessageObject wird mit den Transaktionsparametern gefüllt und mit Execute() ausgelöst. Anschließend können die Parameter der Autorisationsantwort ermittelt werden.

- 1) Erzeugung eines MessageFactory Object.
- 2) Öffnen der entsprechenden Konfiguration mit Open().
- 3) Aufruf von CreateRequest(), um ein leeres MessageObject zu erhalten.
- 4) Aufruf von SetAttribute() mit dem MessageObject zum Setzen der Parameter.
- 5) Aufruf von Execute() zum Auslösen der Zahlungsanfrage.
- 6) Aufruf von GetAttribute() mit dem MessageObject zum Auslesen der Antwortparameter.

3.1.2 Buchung, Stornierung und Tagesabschluss

- 1) Erzeugung eines MessageFactory Object.
- 2) Öffnen der entsprechenden Konfiguration mit Open().
- 3) Aufruf von CreateRequest("PayComplete"), um ein MessageObject zu erhalten.
- 4) Aufruf von SetAttribute() mit dem MessageObject zum Setzen der Parameter.
- 5) Aufruf von Capture() mit dem MessageObject.

3.2 MessageFactory Klasse

```
Class MessageFactory
{
    void Open(String path);
    MessageObject CreatePayInit();
    MessageObject VerifyPayConfirm(String data, String signature);
    MessageObject CreateRequest(String msgtype);
};
```

3.3 MessageObject Klasse

```
Class MessageObject
{
    void SetAttribute(String name, String value);
    String GetAttribute(String name);
    void Capture();
};
```

3.4 Open() Methode

Mit Open() wird auf das Schlüsselpaar des Händler Accounts referenziert. Damit sich weitere Funktionsaufrufe mit dieser MessageFactory ebenfalls auf diese Schlüssel beziehen, muss Open() vor allen anderen Methoden des MessageFactory Objects aufgerufen werden.

3.5 Execute() Methode

Der Aufruf von Execute() übermittelt die Message vom angegebenen Nachrichtentyp (msgtype).

3.6 CreateRequest() Methode

Erzeugt ein neues Request MessageObject vom angegebenen Nachrichtentyp (msgtype). So kann zum Beispiel mit CreateRequest("PayComplete") eine Reservation verbucht oder verworfen, ein Teilbetrag verbucht, eine Buchung storniert und der Tagesabschluss ausgelöst werden. CreateRequest("PayComplete") benötigt für eine Verbuchung immer die Parameter ID und ACCOUNTID. Für eine betragsreduzierte Verbuchung wird zusätzlich der Parameter AMOUNT benötigt, für das Verwerfen einer Reservation, das Stornieren einer Buchung und das Auslösen des Tagesabschlusses ist der Parameter ACTION erforderlich.

3.7 SetAttribute() Methode

Mit SetAttribute() werden die, für die Message benötigten Parameter gesetzt. Bitte beachten Sie bei der Angabe der Parameternamen die Großschreibung.

3.8 GetAttribute() Methode

GetAttribute() liefert den Wert eines Parameters der Message zurück. Ist der Parameter in der Nachricht nicht enthalten schlägt der Aufruf fehl. Bitte beachten Sie bei der Angabe der Parameternamen die Großschreibung.

3.9 Capture() Methode

Der Aufruf von Capture() übermittelt die Message vom Nachrichtentyp CreatePayComplete.

4 Saferpay https Interface

Alternativ zur Saferpay Client Library kann das Saferpay https Interface eingesetzt werden. Das kann zum Beispiel der Fall sein, wenn auf dem Zielsystem die Saferpay LIB nicht installiert oder betrieben werden kann.

4.1 IP-Zugriff und Passwort für das Saferpay https Interface

Saferpay stellt sicher, dass die mit der Händler-Applikation ausgetauschten Daten nicht manipuliert werden können. Manipulationen, die versierte Internetnutzer vornehmen könnten, werden erkannt und der Händler-Applikation angezeigt.

Der Zugriff auf das Saferpay https Interface ist nur möglich durch

- Explizite Freischaltung der rufenden IP-Adresse(n) auf dem Saferpay Server.
- Angabe des https Interface Passworts bei Autorisations- und Verbuchungsanfragen.

Wird eine missbräuchliche Verwendung des Zugangs festgestellt, führt dies zur sofortigen Sperrung des Händlerkontos auf dem Saferpay Gateway. Eine Freigabe kann nur durch die Saferpay Administration erfolgen.

4.2 https Interface Adressen

Das Saferpay https Interface ist über diese Web-Adressen erreichbar:

Autorisierung und Gutschrift

<https://www.saferpay.com/hosting/Execute.asp>

Buchung

<https://www.saferpay.com/hosting/PayCompleteV2.asp>

4.3 Übertragung von Nachrichten

Jede Anfrage wird vom HI beantwortet, mit Ausnahme bei technischen Problemen, die eine Rückgabe der Antwortnachricht nicht erlauben.

Anfrage (Request)

Die Anfragedaten (Parameter) können mittels POST oder GET an das HI übertragen werden.

Beispiel einer Anfrage:

```
https://www.saferpay.com/hosting/Execute.asp?spPassword=hfJK43SA  
&AMOUNT=1295&CURRENCY=EUR...weitere Attribute
```

Antwort (Response)

Das HI antwortet auf Anfragen im Klartext. Die positive Ausführung einer Anfrage wird vom HI mit „OK“ beantwortet, gefolgt von einem Doppelpunkt und den Antwortdaten. Das Standardformat für die Antwortdaten ist XML.

Die positive Quittierung mit „OK“ bedeutet, dass die Anfrage korrekt verarbeitet werden konnte. Die Auswertung der Antwortdaten muss jedoch von der Händler-Applikation erfolgen, etwa die Prüfung des Parameters RESULT.

Beispiel einer Antwortnachricht:

```
OK:<IDP RESULT="65" ...weitere Attribute... />
```

Im Falle eines (Applikations-) Fehlers antwortet der Gateway mit „ERROR“, gefolgt von einem Doppelpunkt und einer optionalen Fehlerbeschreibung.

```
ERROR: Hosting: Merchant not configured or unknown
```

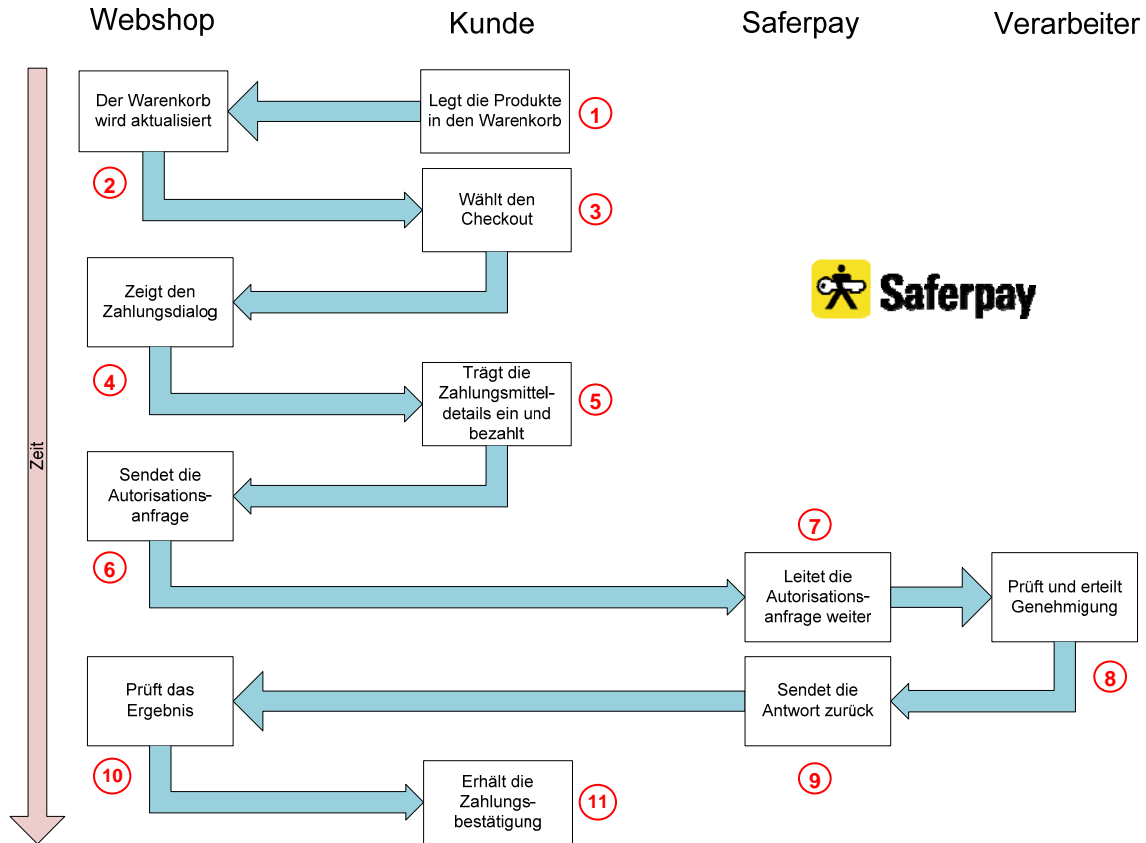
Hinweise

Die Verarbeitung von Anfragen erfolgt immer sequenziell. Gehen mehrere Anfragen parallel beim HI ein, findet spätestens im Saferpay Rechenzentrum eine Serialisierung anhand der Saferpay Kontonummer (ACCOUNTID) statt. Sollen mehrere Transaktionen parallel verarbeitet werden, ist ein Pool von Saferpay Kontonummern anzulegen, die abwechselnd verwendet werden (Round Robin).

5 Verarbeitungsschritte

5.1 Übersicht

Die folgende Grafik zeigt den Ablauf einer erfolgreichen Online Zahlung mit dem Saferpay Authorization Interface:



5.2 Prozessbeschreibung

- ① Der Kunde legt die zum Kauf gewünschten Artikel in den Warenkorb des Shops.
- ② Im Webshop wird der Warenkorb aktualisiert und der zu zahlende Betrag angezeigt.
- ③ Der Kunde begibt sich zur „Kasse“.
- ④ Der Shop-eigene Zahlungsdialog wird angezeigt.
- ⑤ Der Kunde gibt die benötigten Daten seines Zahlungsmittels ein und klickt auf „Bezahlen“.
- ⑥ Der Webshop übermittelt die Zahlungsanfrage an Saferpay.
- ⑦ Saferpay leitet die Zahlungsanfrage unmittelbar an den Verarbeiter weiter.
- ⑧ Der Verarbeiter prüft die Zahlungsanfrage, erteilt die Genehmigung und leitet die Autorisationsantwort an Saferpay zurück.
- ⑨ Saferpay leitet die Antwort auf die Zahlungsanfrage an den Shop weiter.

- ⑩ Der Shop erhält die Antwortdaten der Autorisation und prüft das Ergebnis.
- ⑪ Der Kunde erhält vom Shop seine Zahlungsbestätigung.

Die folgenden Punkte befinden sich nicht auf der Grafik, da sie nicht zwangsläufig von der Shop-Applikation ausgelöst werden müssen:

- ⑫ Der Betrag wird verbucht (PayComplete) und die Bestellung kann ausgeführt werden. Das Verbuchen kann sowohl direkt nach Erhalt der Autorisationsantwort aus dem Webshop oder zu einem späteren Zeitpunkt erfolgen, zum Beispiel aus der Warenwirtschaft heraus.

Das Verbuchen einer Reservation ist für den Tagesabschluss obligatorisch. Dieser berücksichtigt nur Transaktionen mit dem Status „Buchung“ und leitet sie zur Auszahlung an den Verarbeiter weiter. Das Geld wird anschließend in Form einer Sammelposition dem Geschäftskonto des Händlers gutgeschrieben. Vom Zahlungsmittelverarbeiter erhält der Händler eine Abrechnungsliste.

Das Verbuchen kann je nach Geschäftsfall auch später erfolgen, in der Regel innerhalb von sechs Tagen. Solange behält eine Reservation durchschnittlich ihre Gültigkeit. Da dies je nach Verarbeiter und Zahlungsmittel variieren kann, fragen Sie bitte direkt beim Verarbeiter nach, was für Sie gilt.


Der Tagesabschluss kann manuell oder automatisch ausgelöst werden.

- ⑬ Der Status der Transaktion wechselt durch das PayComplete von „Reservation“ auf „Buchung“. Der Transaktionsstatus wird im Saferpay Backoffice Journal angezeigt.

6 Parameter

6.1 Autorisationsanfrage

Die Tabelle listet die Parameter, die für den Nachrichtentyp „Authorization“ zur Verfügung stehen. Wenn nicht anders erwähnt, ist die Verwendung des jeweiligen Parameters Pflicht.

Parameter	Format	Beschreibung
spPassword	ans[..40]	<i>https Interface-Parameter</i> Das Passwort wird für die Kommunikation mit dem HI benötigt und separat zur Verfügung gestellt.
ACCOUNTID	ns[..15]	Die Saferpay Kontonummer des Händlers für diese Transaktion. Zum Beispiel "99867-94913159" für das Saferpay Testkonto.
PAN	n[..19]	Die Kreditkartennummer (Primary Account Number). PAN benötigt zusätzlich die Parameter EXP und CVC (bei Erstzahlungen).
EXP	n[4]	Verfalldatum, wie auf der Kreditkarte angegeben. Das Format ist MMJJ, zum Beispiel "1215" für 12/2015.
CVC	n[..4]	3- oder 4-stellige Kreditkartenprüfnummer, auch bekannt als: CID/4DBC (American Express) CVC2 (MasterCard) CVV2 (Visa) CAV (JCB)
		Die Verwendung des Parameters ist für Erstzahlungen mit einer Kreditkarte Pflicht. Für Folgezahlungen steht der CVC nicht zur Verfügung, da die elektronische Speicherung unter keinen Umständen gestattet ist!
TRACK2	ns[22]	Bankverbindung für das deutsche Elektronische Lastschriftverfahren. Ersetzt bei ELV-Zahlungen die Kreditkartenparameter PAN, EXP und CVC. Das Format ist ";59[BLZ 8-stellig]=[Kontonummer 10-stellig]".
CARDREFID	ans[..40]	<i>Optional</i> Ersatzwert für Kreditkartennummer und Verfalldatum oder Bankverbindung (ELV). Die Verwendung setzt den Service „Saferpay Secure Card Data“ voraus.
AMOUNT	n[..8]	Zahlungsbetrag in kleinster Währungseinheit. Zum Beispiel: "1230" entspricht Betrag 12,30 in Euro.
CURRENCY	a[3]	Dreistelliger ISO 4217 Währungs-Code. Zum Beispiel: "CHF" oder "EUR"
ORDERID	ans[..80]	<i>Optional, Pflichtparameter beim Zahlungsmittel giropay</i> ORDERID enthält die Referenznummer für eine Zahlung. Die ORDERID sollte für eine spätere Zuordnung eindeutig vergeben werden. Saferpay kann für ORDERID 80 Zeichen abbilden, verarbeiterseitig ist das jedoch in der Regel nicht möglich. Zu lange Zeichenketten werden dort abgeschnitten. In der Praxis bewährt haben sich 12 Zeichen. Im Zweifel erfragen Sie bitte bei Ihrem Verarbeiter wie viele Zeichen abgebildet werden können.

Parameter	Format	Beschreibung
NAME	ans[..50]	<i>Optional</i> Enthält den Karteninhabernamen. Sonderzeichen im Namen müssen HTML-kodiert (als Entity oder Unicode) übergeben werden.
MPI_SESSIONID	an[28]	<i>Optional</i> Die Session des VerifyEnrollment-Vorgangs wird für die Autorisationsanfrage benötigt, um eine Zahlung als 3-D Secure zu kennzeichnen (Nur für „Verified by Visa“ und „MasterCard SecureCode“).
IP	ns[..15]	<i>Optional</i> Enthält die IP-Adresse des Kunden zur Ermittlung des Herkunftslandes mittels Geotargeting.
AUTHCODE	n [6]	<i>Optional</i> Enthält den Autorisationscode des Verarbeiters, wenn die Anfrage zum Beispiel telefonisch vorautorisiert wurde.
AUTHFLAGS	n[..2]	<i>Optional</i> Folgende Werte sind möglich: 0 = Standardwert, die Zahlung erfolgt mit Autorisation. 4 = Autorisation der Zahlung bereits erfolgt (AUTHCODE). 16 = Eigenes Risiko, die Zahlung erfolgt ohne Autorisation.
ACTION	a[..6]	<i>Optional</i> Kennzeichnet die Zahlungsanfrage als Buchung oder Gutschrift. Werte: "Debit" (Standard, Karteninhaber bezahlt) und "Credit" (Karteninhaber erhält Geld).

6.2 Autorisationsantwort

Die folgende Tabelle listet die Parameter, die in der Autorisationsantwort enthalten sein können.

Parameter	Format	Beschreibung
MSGTYPE	a[..30]	Enthält immer den Wert „AuthorizationResponse“.
RESULT	n[..4]	Enthält das Ergebnis der Autorisationsanfrage. 0 = Anfrage erfolgreich ausgeführt. ≠0 = Anfrage nicht erfolgreich ausgeführt. Eine Liste möglicher RESULT-Werte findet sich im Kapitel RESULT-Werte.
ACCOUNTID	ns[..15]	Die Saferpay Kontonummer des Händlers für diese Transaktion. Zum Beispiel "99867-94913159" für das Saferpay Testkonto.
ID	an[28]	Eindeutige Saferpay Transaktionskennung.
TOKEN	ans[..40]	Kann zusätzliche Informationen zur Transaktionsverarbeitung enthalten. Standardwert: "(unused)"
PROVIDERID	n[..4]	Enthält die Provider ID des Zahlungsmittelverarbeiters.
PROVIDERNAME	ans[..30]	Enthält den Namen des Zahlungsmittelverarbeiters.
AUTHRESULT	n[..3]	Enthält den Antwort-Code des Verarbeiters. Ist keine Verbindung zum Verarbeiter zustande gekommen, weist der Wert in RESULT darauf hin. Die Werte variieren je nach verwendetem Zahlungsmittelprotokoll.
AUTHCODE	n[6]	Enthält bei erfolgreicher Autorisation den Autorisations-Code des Kreditkartenverarbeiters.
PAYMENT_PROTOCOL	ans[..30]	Der Name des Zahlungsmittelprotokolls, das der Verbindung zugrunde liegt.
CAVV	ans[28]	<i>3-D Secure Parameter*</i> Cardholder Authentication Verification Value Bei einer MasterCard ist hier der UCAF-Wert enthalten. Saferpay benutzt unabhängig vom Kreditkartentyp den Wert CAVV.
MPI_LIABILITYSHIFT	a[..3]	<i>3-D Secure Parameter*</i> Gibt an, ob technisch formal eine Haftungsumkehr besteht. Werte: "yes" oder "no"
<p>Achtung! Nicht alle Verarbeiter können die Haftungsumkehr während der Autorisation überprüfen und diese gegebenenfalls schon mit der Autorisationsantwort ausschließen. Daher ist es möglich, dass vom Verarbeiter für eine Transaktion aus vertraglichen Gründen keine Haftungsumkehr gewährt wird, auch wenn MPI_LIABILITYSHIFT und ECI das Gegenteil aussagen. Fragen Sie bei Bedarf direkt bei Ihrem Verarbeiter nach, ob dieser dazu in der Lage ist.</p>		
XID	ans[28]	<i>3-D Secure Parameter*</i> Extra Identifier Diese Base64-Zeichenfolge wird vom MPI vergeben und referenziert auf den Vorgang im 3-D Secure Protokoll.

Parameter	Format	Beschreibung
ECI	n[1]	3-D Secure Parameter* Electronic Commerce Indicator Wird zur Kennzeichnung von 3-D Secure Transaktionen benötigt („Verified by Visa“, „MasterCard SecureCode“): 0 = Internet Zahlung ohne Haftungsumkehr. 1 = 3-D Secure Zahlung mit Authentisierung. 2 = 3-D Secure Zahlung, Karte nimmt am Verfahren nicht teil.
AUTHDATE	ns[17]	Enthält den Zeitstempel der Autorisation. Format: YYYYMMDD hh:mm:ss
EXP	n[4]	Enthält das Verfalldatum der angefragten Kreditkarte. Format: MMY
PAN	ans[..23]	Enthält die maskierte Kreditkartennummer oder Bankverbindung der Anfrage. Zum Beispiel "xxxx xxxx xxxx 0111" oder "760260xx/xxxxxx4332".
CARDREFID	ans[..40]	Erhält den Ersatzwert für das Zahlungsmittel mit dem die Autorisationsanfrage gestellt wurde.
REFERRAL	ans[..30]	Enthält je nach Verarbeiter eine Telefonnummer oder einen Texthinweis für eine telefonische Vorautorisation.
ACQUIRER_TERMINALID	n[..10]	Enthält die Terminalkennung des ELV-Verarbeiters.
BANK_CODE_NUMBER	n[8]	Enthält die Bankleitzahl der angefragten Bankverbindung.
PROTOCOL_AID	n[8]	Enthält bei erfolgreicher Autorisation den Autorisations-Code des Lastschriftenverarbeiters.
PROTOCOL_STAN	n[..9]	Enthält die Sequenznummer des Lastschriftterminals.
AUTHMESSAGE	ans[..30]	Enthält eine Antwort zur Autorisation in Textform.

* Voraussetzung ist die Teilnahme am 3-D Secure Verfahren („Verified by Visa“, „MasterCard SecureCode“)

6.3 PayComplete-Anfrage

Folgende Parameter stehen für eine CreatePayComplete Nachricht zur Verfügung:

Parameter	Format	Beschreibung
spPassword	ans[..40]	<i>https Interface-Parameter</i> Das Passwort wird für die Kommunikation mit dem HI benötigt und separat zur Verfügung gestellt.
ID	an[28]	Saferpay Transaktionskennung der PayConfirm Nachricht. Pflichtparameter, es sei denn ACTION=CloseBatch.
AMOUNT	n[..8]	<i>Optional</i> Zahlungsbetrag in kleinster Währungseinheit, zum Beispiel "1230" entspricht dem Betrag 12,30 in Euro.
ACCOUNTID	ns[..15]	Die Saferpay Kontonummer des Händlers für diese Transaktion. Zum Beispiel "99867-94913159" für das Saferpay Testkonto.
ACTION		<i>Optional</i> Gibt eine erweiterte Verarbeitungsoption an. Mögliche Werte sind: "Settlement", "CloseBatch", "Cancel" Settlement Weist das Saferpay System an den Status der Transaktion von Reservation in Buchung zu ändern. Damit wird die Transaktion beim nächsten Tagesabschluss zur Auszahlung an den Verarbeiter weitergeleitet. Mit dem Parameter AMOUNT kann bei der Verbuchung ein geringerer Betrag angegeben werden, als reserviert wurde. Die Differenz wird hierbei automatisch verworfen. Maximal kann der autorisierte Betrag verbucht werden. CloseBatch Weist Saferpay an den Tagesabschluss für die angegebene ACCOUNTID durchzuführen. Wird der Parameter ID übergeben, schlägt der Aufruf fehl. Cancel Mit diesem Aufruf kann eine Reservation verworfen oder eine Buchung storniert werden, solange sie nicht mit dem Tagesabschluss verarbeitet wurde. Handelt es sich um eine Reservation wird die Transaktion nach dem Verwerfen im Saferpay Backoffice nur noch 6 Tage unter „verworfenen Reservationen“ angezeigt. Danach wird Sie aus der der Datenbank gelöscht. Stornierte Buchungen bleiben dagegen, als storniert gekennzeichnet, im Backoffice sichtbar. Ohne Angabe von ACTION wird standardmäßig ACTION="Settlement" gesetzt.

6.4 PayComplete-Antwort

Die Antwort auf die Verbuchungsanfrage enthält die folgenden Parameter:

Parameter	Format	Beschreibung
MSGTYPE	a[..30]	Enthält immer den Wert „PayConfirm“.
ID	an[28]	Saferpay Transaktionskennung
RESULT	n[..4]	Enthält das Ergebnis der Autorisationsanfrage. 0 = Anfrage erfolgreich ausgeführt. ≠0 = Anfrage nicht erfolgreich ausgeführt.
MESSAGE	ans[..30]	Enthält eine Antwort zur Buchungsanfrage in Textform.
AUTHMESSAGE	ans[..30]	Kann eine Antwort zur Autorisation in Textform enthalten.

7 Saferpay Testkonto

Während der Integrationsphase in den Webshop empfiehlt sich die Verwendung des Saferpay Testkontos.

ACCOUNTID	99867-94913159
Login	e99867001
Password	XAjc3Kna
spPassword	XAjc3Kna (nur für das https Interface)

Kartenummer	Beschreibung
9451123100000111	Saferpay Testkarte „enrolled“, liefert ECI=1 mit der Autorisationsantwort. Die Karte nimmt am 3-D Secure Verfahren teil.
9451123100000004	Saferpay Testkarte „not enrolled“, liefert ECI=2 mit der Autorisationsantwort. Die Karte nimmt am 3-D Secure Verfahren nicht teil.
9451123100000202	Saferpay Testkarte „unable to enroll“, liefert ECI=0 mit der Autorisationsantwort. Mit dieser Karte sind nur SSL-Transaktionen möglich.
9451123100000103	Saferpay Testkarte Result Testkarte um besondere Antwort-Codes zu erzwingen. Die letzten beiden Stellen des Parameters AMOUNT bestimmen den Ausgang der Autorisationsanfrage. Bei einem Wert gleich Null "00" wird eine erfolgreiche Zahlungsanfrage durchgeführt. Bei einem Wert ungleich Null sendet Saferpay in der Antwort RESULT="65" und der übergebene Wert entspricht dem Ablehnungs-Code in AUTHRESULT.

Das Testkonto wird von mehreren Entwicklern genutzt. Daher kommt es vor, dass auf dem Testkonto Vorgänge und Buchungen erscheinen, die von anderen stammen.

Das Testkonto unterstützt nur Saferpay Testkarten. Andere Kartentypen sind nicht verfügbar. Die Testkarten besitzen keine fest zugehörige Kartenprüfnummer (CVC / CVV2) und Verfalldatum. Beides kann frei gewählt werden, mit der Einschränkung, dass die Kartenprüfnummer dreistellig oder vierstellig numerisch sein und das Verfalldatum in der Zukunft liegen muss. Ansonsten gleicht der Bezahlvorgang mit den Saferpay Testkarten dem mit Kreditkarten in einer produktiven Umgebung.

8 Beispiele

8.1 Wichtiger Hinweis



Bitte beachten Sie, dass eigene Werte HTML-kodiert werden sollten, entweder als HTML-Entity oder Unicode. So stellen Sie sicher, dass alle Sonderzeichen korrekt an Saferpay übergeben werden.

8.2 C# mit der .NET LIB

Autorisationsanfrage Buchung

```
MessageFactory mf = new MessageFactory();
mf.Open("");
MessageObject request = mf.CreateRequest("Authorization");

request.SetAttribute("ACCOUNTID", "99867-94913159");
request.SetAttribute("AMOUNT", "12500");
request.SetAttribute("CURRENCY", "EUR");
request.SetAttribute("PAN", "9451123100000111");
request.SetAttribute("EXP", "1214");
request.SetAttribute("CVC", "123");
request.SetAttribute("NAME, Server.HtmlEncode("Stefanie Müller"));
request.SetAttribute("ORDERID", "123456789");

MessageObject response = request.Execute();
```

Autorisationsanfrage Gutschrift

```
MessageFactory mf = new MessageFactory();
mf.Open("");
MessageObject request = mf.CreateRequest("Authorization");

request.SetAttribute("ACCOUNTID", "99867-94913159");
request.SetAttribute("AMOUNT", "12500");
request.SetAttribute("CURRENCY", "EUR");
request.SetAttribute("PAN", "9451123100000111");
request.SetAttribute("EXP", "1214");
request.SetAttribute("CVC", "123");
request.SetAttribute("ACTION", "Credit");
request.SetAttribute("ORDERID", "123456789");

MessageObject response = request.Execute();
```



Autorisationsantwort

```
int result = Convert.ToInt32(response.GetAttribute("RESULT"));
if (result == 0)
{
    String id = response.GetAttribute("ID");
    Console.WriteLine("Autorisation erfolgreich!");
}
else
{
    Console.WriteLine("Autorisation fehlgeschlagen! RESULT=" + result);
    return;
}
```

Verbuchung einer Zahlung mit CreatePayComplete

```
MessageFactory mf = new MessageFactory();
mf.Open("");
MessageObject mo_paycomplete = mf.CreateRequest("PayComplete");

mo_paycomplete.SetAttribute("ID", id);
mo_paycomplete.SetAttribute("ACCOUNTID", "99867-94913159");

MessageObject captureresponse = mo_paycomplete.Capture();
```

Betragsreduzierte Verbuchung einer Zahlung mit CreatePayComplete

```
MessageFactory mf = new MessageFactory();
mf.Open("");
MessageObject mo_paycomplete = mf.CreateRequest("PayComplete");

mo_paycomplete.SetAttribute("ID", id);
mo_paycomplete.SetAttribute("ACCOUNTID", "99867-94913159");
mo_paycomplete.SetAttribute("AMOUNT", "10000");

MessageObject captureresponse = mo_paycomplete.Capture();
```

Stornierung einer Zahlung mit CreatePayComplete

```
MessageFactory mf = new MessageFactory();
mf.Open("");
MessageObject mo_paycomplete = mf.CreateRequest("PayComplete");

mo_paycomplete.SetAttribute("ID", id);
mo_paycomplete.SetAttribute("ACCOUNTID", "99867-94913159");
mo_paycomplete.SetAttribute("ACTION", "Cancel");

MessageObject captureresponse = mo_paycomplete.Capture();
```

Tagesabschluss auslösen mit CreatePayComplete

```
MessageFactory mf = new MessageFactory();
mf.Open("");
MessageObject mo_paycomplete = mf.CreateRequest("PayComplete");

mo_paycomplete.SetAttribute("ACCOUNTID", "99867-94913159");
mo_paycomplete.SetAttribute("ACTION", "CloseBatch");

MessageObject captureresponse = mo_paycomplete.Capture();
```

PayComplete-Antwort

```
int result = Convert.ToInt32(response.GetAttribute("RESULT"));
if (result == 0)
{
    String id = captureresponse.GetAttribute("ID");
    String msg = captureresponse.GetAttribute("MESSAGE");
    Console.WriteLine("Verbuchung erfolgreich!");
}
else
{
    Console.WriteLine("Verbuchung fehlgeschlagen!");
    return;
}
```

8.3 Java mit der Java LIB

Autorisationsanfrage Buchung

```
import Saferpay.*;
import org.apache.commons.lang.*

MessageFactory mf = new MessageFactory();
mf.Open("");
MessageObject request = mf.CreateRequest("Authorization");

request.SetAttribute("ACCOUNTID", "99867-94913159");
request.SetAttribute("AMOUNT", "12500");
request.SetAttribute("CURRENCY", "EUR");
request.SetAttribute("PAN", "9451123100000111");
request.SetAttribute("EXP", "1214");
request.SetAttribute("CVC", "123");
request.SetAttribute("NAME", StringEscapeUtils.escapeHtml("Stefanie Müller"));
request.SetAttribute("ORDERID", "123456789");

MessageObject response = request.Execute();
```



Autorisationsanfrage Gutschrift

```
import Saferpay.*;

MessageFactory mf = new MessageFactory();
mf.Open("");
MessageObject request = mf.CreateRequest("Authorization");
request.SetAttribute("ACCOUNTID", "99867-94913159");
request.SetAttribute("AMOUNT", "12500");
request.SetAttribute("CURRENCY", "EUR");
request.SetAttribute("PAN", "9451123100000111");
request.SetAttribute("EXP", "1214");
request.SetAttribute("CVC", "123");
request.SetAttribute("ACTION", "Credit");
request.SetAttribute("ORDERID", "123456789");

MessageObject response = request.Execute();
```

Autorisationsantwort

```
int result = response.GetAttribute("RESULT");
if (result == 0)
{
    String id = response.GetAttribute("ID");
    System.out.println("Autorisation erfolgreich!");
}
else
{
    System.out.println("Autorisation fehlgeschlagen! RESULT=" + result);
}
```

Verbuchung einer Zahlung mit CreatePayComplete

```
import Saferpay.*;

MessageFactory mf = new MessageFactory();
mf.Open("");
MessageObject mo_paycomplete = mf.CreateRequest("PayComplete");

mo_paycomplete.SetAttribute("ID", id);
mo_paycomplete.SetAttribute("ACCOUNTID", "99867-94913159");

MessageObject captureresponse = mo_paycomplete.Capture();
```

Betragsreduzierte Verbuchung einer Zahlung mit CreatePayComplete

```
import Saferpay.*;

MessageFactory mf = new MessageFactory();
mf.Open("");
MessageObject mo_paycomplete = mf.CreateRequest("PayComplete");

mo_paycomplete.SetAttribute("ID", id);
mo_paycomplete.SetAttribute("ACCOUNTID", "99867-94913159");
mo_paycomplete.SetAttribute("AMOUNT", "10000");

MessageObject captureresponse = mo_paycomplete.Capture();
```


Stornierung einer Zahlung mit CreatePayComplete

```
import Saferpay.*;

MessageFactory mf = new MessageFactory();
mf.Open("");
MessageObject mo_paycomplete = mf.CreateRequest("PayComplete");

mo_paycomplete.SetAttribute("ID", id);
mo_paycomplete.SetAttribute("TOKEN", token);
mo_paycomplete.SetAttribute("ACCOUNTID", "99867-94913159");
mo_paycomplete.SetAttribute("ACTION", Cancel);

MessageObject captureresponse = mo_paycomplete.Capture();
```

Tagesabschluss auslösen mit CreatePayComplete

```
import Saferpay.*;

MessageFactory mf = new MessageFactory();
mf.Open("");
MessageObject mo_paycomplete = mf.CreateRequest("PayComplete");

mo_paycomplete.SetAttribute("ACCOUNTID", "99867-94913159");
mo_paycomplete.SetAttribute("ACTION", "CloseBatch");

MessageObject captureresponse = mo_paycomplete.Capture();
```

PayComplete-Antwort

```
int result = response.GetAttribute("RESULT");
if (result == 0)
{
    String id = captureresponse.GetAttribute("ID");
    String msg = captureresponseresponse.GetAttribute("MESSAGE");
    System.out.println("Verbuchung erfolgreich!");
}
else
{
    System.out.println("Verbuchung fehlgeschlagen!");
    return;
}
```

8.4 Kommandozeilenaufrufe mit der Java LIB

Autorisationsanfrage Buchung

```
java -jar Saferpay.jar -exec -p c:/programme/soplex/saferpay/keys/99867 -m Authorization -a
ACCOUNTID 99867-94913159 -a PAN 9451123100000004 -a EXP 1214 -a CVC 123 -a AMOUNT 12500 -a
CURRENCY EUR -a ORDERID 123456789 -a NAME "Stefanie Mülller" -of cai.txt
```

Autorisationsanfrage Gutschrift

```
java -jar Saferpay.jar -exec -p c:/programme/soplex/saferpay/keys/99867 -m Authorization -a
ACCOUNTID 99867-94913159 -a PAN 9451123100000004 -a EXP 1214 -a CVC 123 -a AMOUNT 12500 -a
CURRENCY EUR -a ORDERID 123456789 -a NAME "Stefanie Mülller" -a ACTION Credit -of
cai_refund.txt
```



Autorisationsantwort

Buchung (Inhalt von cai.txt):

```
<IDP PAN="xxxx xxxx xxxx 0004" CCCOUNTRY="XX" EXP="1214"  
REFERRAL="017772357" AUTHRESULT="1" AUTHCODE="745000"  
ACCOUNTID="99867-94913159" RESULT="0" AUTHDATE="20110418 14:12:31"  
PAYMENT_PROTOCOL="CARCDS" PROVIDERNAME="Saferpay Test Card"  
PROVIDERID="90" ID="Ctp7OpbnQ8phSA13Ev9Wb512S0bA"  
MSGTYPE="AuthorizationResponse" AUTHMESSAGE="request was processed successfully" ECI="0"  
CONTRACTNUMBER="123456789" TOKEN="(unused)"/>
```

Gutschrift (Inhalt von cai_refund.txt):

```
<IDP PAN="xxxx xxxx xxxx 0004" CCCOUNTRY="XX" EXP="1214"  
REFERRAL="017772357" AUTHRESULT="1" AUTHCODE=" " "  
ACCOUNTID="99867-94913159" RESULT="0" AUTHDATE="20110418 14:24:14"  
PAYMENT_PROTOCOL="CARCDS" PROVIDERNAME="Saferpay Test Card"  
PROVIDERID="90" ID="xI41vCAC1Sz2vAKY9YGOA7dLI9Ub"  
MSGTYPE="AuthorizationResponse" AUTHMESSAGE="request was processed successfully" ECI="0"  
CONTRACTNUMBER="123456789" TOKEN="(unused)"/>
```

Verbuchung einer Zahlung mit CreatePayComplete

```
java -jar Saferpay.jar -capt -p c:/programme/soplex/saferpay/keys/99867 -a ACCOUNTID 99867-94913159 -if cai.txt -of capt.txt
```

Alternativ können die Parameter direkt übergeben werden:

```
java -jar Saferpay.jar -capt -p c:/programme/soplex/saferpay/keys/99867 -i Ctp7OpbnQ8phSA13Ev9Wb512S0bA -a ACCOUNTID 99867-94913159 -of capt.txt
```

Betragsreduzierte Verbuchung einer Zahlung mit CreatePayComplete

```
java -jar Saferpay.jar -capt -p c:/programme/soplex/saferpay/keys/99867 -a ACCOUNTID 99867-94913159 -if cai.txt -a AMOUNT 10000 -of capt.txt
```

Alternativ können die Parameter direkt übergeben werden:

```
java -jar Saferpay.jar -capt -p c:/programme/soplex/saferpay/keys/99867 -i 8xMY2hbb33dGtA8x96Y1b27jjOfb -a ACCOUNTID 99867-94913159 -a AMOUNT 10000 -of capt.txt
```

Stornierung einer Zahlung mit CreatePayComplete

```
java -jar Saferpay.jar -capt -p c:/programme/soplex/saferpay/keys/99867 -a ACCOUNTID 99867-94913159 -if cai.txt -a ACTION Cancel -of cancel.txt
```

Alternativ können die Parameter direkt übergeben werden:

```
java -jar Saferpay.jar -capt -p c:/programme/soplex/saferpay/keys/99867 -i Ctp7OpbnQ8phSA13Ev9Wb512S0bA -a ACCOUNTID 99867-94913159 -a ACTION Cancel -of cancel.txt
```

Tagesabschluss auslösen mit CreatePayComplete

```
java -jar Saferpay.jar -capt -p c:/programme/soplex/saferpay/keys/99867 -a ACTION CloseBatch -a ACCOUNTID 99867-94913159 -of daily.txt
```

8.5 https Interface

Autorisationsanfrage Buchung

<https://www.saferpay.com/hosting/execute.asp?spPassword=XAjc3Kna&ACCOUNTID=99867-94913159&ORDERID=123456789-001&AMOUNT=1000&CURRENCY=EUR&PAN=9451123100000004&EXP=1214&CVC=123>

Autorisationsanfrage Gutschrift

<https://www.saferpay.com/hosting/execute.asp?spPassword=XAjc3Kna&ACTION=Credit&ACCOUNTID=99867-94913159&ORDERID=123456789-001&AMOUNT=1000&CURRENCY=EUR&PAN=9451123100000004&EXP=1214>

Autorisationsantwort

```
OK:<IDP RESULT="0" MSGTYPE="AuthorizationResponse" ID="EvrKOEApM3YtSapnE0M1AU28nCYb"
TOKEN="(unused)" AUTHRESULT="1" AUTHMESSAGE="request was processed successfully"
AUTHCODE="500000" PROVIDERID="90" PROVIDERNAME="Saferpay Test Card" ECI="0" CCCOUNTRY="XX"
CONTRACTNUMBER="123456789" ORDERID="123456789-001" AUTHDATE="20110621 10:07:26" EXP="1214"
PAN="xxxx xxxx xxxx 0004"/>
```

Verbuchung einer Zahlung mit CreatePayComplete

<https://www.saferpay.com/hosting/paycompletev2.asp?spPassword=XAjc3Kna&ACCOUNTID=99867-94913159&ID=EvrKOEApM3YtSapnE0M1AU28nCYb>

Betragsreduzierte Verbuchung einer Zahlung mit CreatePayComplete

<https://www.saferpay.com/hosting/paycompletev2.asp?spPassword=XAjc3Kna&ACCOUNTID=99867-94913159&ID=8IK98fhtS4d7UAzMGGCEAEQ12vbA&AMOUNT=500>

Stornierung einer Zahlung mit CreatePayComplete

<https://www.saferpay.com/hosting/paycompletev2.asp?spPassword=XAjc3Kna&ACCOUNTID=99867-94913159&ID=EvrKOEApM3YtSapnE0M1AU28nCYb&ACTION=Cancel>

Tagesabschluss auslösen mit CreatePayComplete

<https://www.saferpay.com/hosting/paycompletev2.asp?spPassword=XAjc3Kna&ACCOUNTID=99867-94913159&ACTION=CloseBatch>

PayComplete-Antwort

```
OK:<IDP RESULT="0" />
```

9 RESULT-Werte

Eine Autorisation wurde nur erfolgreich durchgeführt, wenn RESULT=0 ist. Bei allen anderen RESULT-Werten handelt es sich um fehlgeschlagene Anfragen.

Wert	Beschreibung	Erklärung
Autorisationsantwort:		
5	Access denied	Der Zugriff mit dieser IP-Adresse ist für den Account nicht freigegeben.
61	Invalid card	Die Plausibilitätsprüfung der Karte ist fehlgeschlagen.
62	Invalid date	Das Verfalldatum ist nicht plausibel.
63	Card expired	Die Karte ist verfallen.
64	Unknown card	Die Karte ist unbekannt, sie konnte keinem Kartentyp zugeordnet werden.
65	Authorization declined	Der Kartenverarbeiter hat die Transaktion abgelehnt. Im Feld AUTHRESULT ist der Ablehnungsgrund vom Verarbeiter enthalten.
67	No contract available	Auf dem Terminal ist kein Akzeptanzvertrag für den Kartentyp oder die Währung vorhanden.
70	Geo IP not white listed	Das IP-Herkunftsland der Anfrage ist nicht im Saferpay Risk Management freigegeben.
83	Invalid currency	Der Währungs-Code ist unbekannt.
84	Invalid amount	Der Betrag ist ungültig.
85	No credits available	Das Transaktionsabo ist aufgebraucht.
102	Function not supported	Der Verarbeiter unterstützt diese Funktion nicht.
104	PAN black listed	Die Karte ist durch das Saferpay Risk Management blockiert worden.
105	Card country not white listed	Das Herkunftsland der Karte ist nicht im Saferpay Risk Management freigegeben.
113	CVC wrong value	Die Kartenprüfnummer enthält einen ungültigen Wert.
114	CVC mandatory	Die Angabe der Kartenprüfnummer ist zwingend erforderlich.
8100	Referenced transaction not found	Die MPI_SESSIONID ist unbekannt.

10 Kontakt

10.1 Saferpay Integration Team

Haben Sie Fragen zu diesem Dokument oder Probleme bei der Saferpay Integration oder benötigen Unterstützung? Dann wenden Sie sich gern an das Integration Team:

Saferpay Schweiz
SIX Payment Services AG
Hardturmstrasse 201
8021 Zürich
+41 848 66 44 44
www.saferpay.com
integration@saferpay.com

Saferpay Europa
SIX Payment Services (Germany) GmbH
Langenhorner Chaussee 92-94
22415 Hamburg
+49 40 325 967- 280
www.saferpay.com
integration@saferpay.com

10.2 Saferpay Support Team

Haben Sie Fragen zu Fehlermeldungen oder gibt es Probleme im laufenden Betrieb? Dann steht Ihnen unser Support Team zur Verfügung:

Saferpay Schweiz
SIX Payment Services AG
Hardturmstrasse 201
8021 Zürich
+41 848 66 44 44
www.saferpay.com
support@saferpay.com

Saferpay Europa
SIX Payment Services (Germany) GmbH
Langenhorner Chaussee 92-94
22415 Hamburg
+49 40 325 967- 250
www.saferpay.com
support@saferpay.com

Das Saferpay Team wünscht Ihnen viel Erfolg mit Ihrer Saferpay E-Payment Lösung!

