



Payment Services

# Payment Page



## Spezifikation

Version 4.1.6

**Inhaltsverzeichnis**

<b>1</b>	<b>Einleitung</b> .....	<b>4</b>
<b>1.1</b>	<b>Voraussetzungen</b> .....	<b>4</b>
<b>1.2</b>	<b>Anzeige der Payment Page</b> .....	<b>4</b>
<b>1.3</b>	<b>Datensicherheit und PCI DSS</b> .....	<b>5</b>
<b>1.4</b>	<b>Unterstützte Zahlungsmittel</b> .....	<b>5</b>
<b>1.5</b>	<b>Formatangaben</b> .....	<b>5</b>
<b>2</b>	<b>Saferpay Client Library</b> .....	<b>6</b>
<b>2.1</b>	<b>Installation</b> .....	<b>6</b>
2.1.1	.NET Client Library.....	6
2.1.2	Java Client Library .....	6
<b>2.2</b>	<b>Schlüsselerzeugung</b> .....	<b>6</b>
2.2.1	.NET Client Library.....	6
2.2.2	Java Client Library .....	7
<b>3</b>	<b>Klassen und Methoden der Client Library</b> .....	<b>8</b>
<b>3.1</b>	<b>Zusammenfassung</b> .....	<b>8</b>
3.1.1	Erzeugung des Zahlungslinks.....	8
3.1.2	Prüfung der Autorisationsantwort.....	8
3.1.3	Verbuchung einer Reservation .....	8
<b>3.2</b>	<b>MessageFactory Klasse</b> .....	<b>8</b>
<b>3.3</b>	<b>MessageObject Klasse</b> .....	<b>9</b>
<b>3.4</b>	<b>Open() Methode</b> .....	<b>9</b>
<b>3.5</b>	<b>CreatePayInit() Methode</b> .....	<b>9</b>
<b>3.6</b>	<b>VerifyPayConfirm() Methode</b> .....	<b>9</b>
<b>3.7</b>	<b>CreateRequest() Methode</b> .....	<b>9</b>
<b>3.8</b>	<b>SetAttribute() Methode</b> .....	<b>9</b>
<b>3.9</b>	<b>GetAttribute() Methode</b> .....	<b>9</b>
<b>3.10</b>	<b>GetPostURL() Methode</b> .....	<b>9</b>
<b>3.11</b>	<b>GetPostData() Methode</b> .....	<b>9</b>
<b>3.12</b>	<b>GetPostSignature() Methode</b> .....	<b>10</b>
<b>3.13</b>	<b>Capture() Methode</b> .....	<b>10</b>
<b>4</b>	<b>Saferpay https Interface</b> .....	<b>11</b>
<b>4.1</b>	<b>https Interface Adressen</b> .....	<b>11</b>
<b>4.2</b>	<b>Transaktionsverlauf</b> .....	<b>11</b>
<b>4.3</b>	<b>Erzeugung des Zahlungs-URL über die CreatePayInit-Adresse</b> .....	<b>12</b>
<b>4.4</b>	<b>Prüfung der Autorisationsantwort über die VerifyPayConfirm-Adresse</b> .....	<b>12</b>
<b>4.5</b>	<b>Beispiel VerifyPayConfirm</b> .....	<b>13</b>
<b>4.6</b>	<b>Verbuchung einer Zahlung über die PayComplete-Adresse</b> .....	<b>13</b>
<b>4.7</b>	<b>Aufruf der Payment Page über die Redirect-Adresse</b> .....	<b>14</b>
4.7.1	Nachteile von Redirect.....	14
4.7.2	https Interface Adresse.....	14
4.7.3	Transaktionsverlauf mit Redirect .....	14
<b>5</b>	<b>Verarbeitungsschritte</b> .....	<b>15</b>
<b>5.1</b>	<b>Übersicht</b> .....	<b>15</b>
<b>5.2</b>	<b>Prozessbeschreibung</b> .....	<b>15</b>
<b>6</b>	<b>Parameter</b> .....	<b>17</b>
<b>6.1</b>	<b>PayInit Parameter</b> .....	<b>17</b>
<b>6.2</b>	<b>PayConfirm Parameter</b> .....	<b>20</b>
<b>6.3</b>	<b>PayComplete Parameter</b> .....	<b>22</b>



6.4	PayComplete-Antwort .....	23
7	Saferpay Testkonto.....	24
8	Beispiele .....	25
8.1	Wichtiger Hinweis .....	25
8.2	C# mit der .NET LIB .....	25
8.3	Java mit der Java LIB .....	27
8.4	Kommandozeilenaufrufe mit der Java LIB .....	29
8.5	https Interface .....	31
9	Fehler-Codes .....	33
10	Kontakt.....	34
10.1	Saferpay Integration Team .....	34
10.2	Saferpay Support Team .....	34

## 1 Einleitung

Die Saferpay Payment Page, in der Folge auch PP genannt, ist ein Online-Bezahlformular, das von Saferpay bereit gestellt wird. Dieses Dokument beschreibt die Integration der PP in bestehende Shop-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.

### 1.1 Voraussetzungen

Die Nutzung der PP setzt Folgendes voraus:

- Eine entsprechende Saferpay eCommerce-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](mailto:DMP-MSCD-all@six-group.com), wenn Sie Ihren Saferpay-Vertrag in der Schweiz abgeschlossen haben und an [service@saferpay.com](mailto:service@saferpay.com), wenn Sie Ihren Vertrag in einem anderen Land (D, NL, A, und so weiter) abgeschlossen haben. Bei Verwendung der Redirect-Adresse teilen Sie den Kollegen bitte unbedingt die vollständige Web-Adresse für den SUCCESSLINK mit, da diese dann fest bei Saferpay hinterlegt werden muss.

### 1.2 Anzeige der Payment Page

Saferpay unterstützt ausschließlich die ganzseitige Anzeige der Payment Page. Für andere Integrationsvarianten, wie die Einbettung der PP in einen Inline Frame oder das Öffnen der PP als Pop-Up, leistet Saferpay keinen Support.

Saferpay folgt damit der Forderung vieler Verarbeiter und der Kreditkartenorganisationen Visa und MasterCard, die aus Sicherheitsgründen eine Integration ihrer Zahlungsmittel im IFrame oder als Pop-Up untersagen. Hintergrund ist, dass es bei der Darstellung der PP im IFrame oder als Pop-Up für den Nutzer anhand der Browser-Adresszeile nicht ersichtlich ist, dass er sich nicht mehr auf der Seite des Webshops befindet. Da der Nutzer bei Aufruf der PP zur Eingabe seiner Zahlungsmitteldaten aufgefordert wird, muss für den Kunden zu jedem Zeitpunkt ersichtlich sein, bei welchem Anbieter beziehungsweise auf welcher Website er sich befindet.

Ein Vorteil der ganzseitigen Darstellung ist eine wesentliche Verbesserung der Benutzerfreundlichkeit für den Kunden, da URL und SSL-Zertifikat jederzeit sichtbar sind. Weiter werden Zahlungsabbrüche reduziert, die durch Browser-Einstellungen oder Sicherheitsabfragen der Verarbeiter bedingt sind.

### 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.

Bei Verwendung der PP erfasst der Karteninhaber seine Kreditkartennummer und das Verfalldatum nicht innerhalb der E-Commerce-Applikation des Händlers, sondern innerhalb des Saferpay Zahlungsfensters (PP). Da die E-Commerce-Applikation und Saferpay auf physisch getrennten Plattformen betrieben werden, besteht keine Gefahr, dass die Kreditkartendaten in der Datenbank des Händler-Systems gespeichert werden können. Das Risiko eines Missbrauchs der Kreditkartendaten wird dadurch reduziert und eine aufwendige PCI DSS Zertifizierung für den Händler entfällt.

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](https://www.pcisecuritystandards.org/pdfs/pci_pa-dss_list.pdf)).

### 1.4 Unterstützte Zahlungsmittel

Die Saferpay Payment Page kann aktuell folgende Zahlungsmittel verarbeiten:

- Visa (inklusive der 3-D Secure Sicherheitstechnologie Verified by Visa)
- MasterCard (inklusive der 3-D Secure Sicherheitstechnologie MasterCard SecureCode)
- Maestro international
- V PAY
- American Express
- Diners Club
- J.C.B.
- Union Card
- Elektronisches Lastschriftverfahren ELV (nur Deutschland)
- giropay
- iDEAL
- PayPal
- mpass
- eps
- ClickandBuy
- PostFinance Card und PostFinance E-Finance

### 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 E-Commerce-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](https://www.saferpay.com/download/saferpay_dotnet.exe)

Java LIB: [https://www.saferpay.com/download/saferpay\\_java.zip](https://www.saferpay.com/download/saferpay_java.zip)

Kann weder die .NET LIB, noch die Java LIB genutzt werden oder ist keine lokale Installation möglich, 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 <zielverzeichnis> -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
```

### 3 Klassen und Methoden der Client Library

Im Folgenden werden Klassen und Methoden beschrieben, die für die Integration der Saferpay Payment Page verwendet werden können.

#### 3.1 Zusammenfassung

##### 3.1.1 Erzeugung des Zahlungslinks

Der Zahlungslink wird mit der CreatePayInit() Methode erstellt. Das erzeugte MessageObject muss vor dem Aufruf von GetPostUrl() mit den Transaktionsparametern gefüllt werden.

- 1) Erzeugung eines MessageFactory Object.
- 2) Öffnen der entsprechenden Konfiguration mit Open().
- 3) Aufruf von CreatePayInit(), um ein leeres MessageObject zu erhalten.
- 4) Aufruf von SetAttribute() mit dem MessageObject zum Setzen der Parameter.
- 5) Aufruf von GetPostUrl(), um einen Zahlungslink für die Payment Page zu erhalten.

##### 3.1.2 Prüfung der Autorisationsantwort

- 1) Erzeugung eines MessageFactory Object.
- 2) Öffnen der entsprechenden Konfiguration mit Open().
- 3) Aufruf von VerifyPayConfirm() zur Verifizierung von DATA und SIGNATURE.
- 4) Auslesen von ID und TOKEN mit GetAttribute().
- 5) Speichern der Werte von ID und TOKEN.

##### 3.1.3 Verbuchung einer Reservation

- 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 von ID und TOKEN.
- 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);
    String GetPostURL();
    String GetPostData();
    String GetPostSignature();
    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 CreatePayInit() Methode

Mit CreatePayInit() kann ein Zahlungslink erstellt werden. Dieser wird unter Einbeziehung der gesetzten Parameter mit GetURL() erzeugt.

### 3.6 VerifyPayConfirm() Methode

Überprüft die digitale Signatur der Bestätigungsmeldung (MSGTYPE=PayConfirm) an den Shop über SUCCESSLINK oder NOTIFYURL, um sicherzustellen, dass die Antwort nicht manipuliert wurde.

### 3.7 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.

### 3.8 SetAttribute() Methode

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

### 3.9 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.

### 3.10 GetPostURL() Methode

Der Aufruf von GetPostURL() liefert den Zahlungslink der Message.

### 3.11 GetPostData() Methode

Der Aufruf von GetPostData() liefert den liefert das DATA der Message.

### **3.12 GetPostSignature() Methode**

Der Aufruf von GetPostSignature() liefert die SIGNATURE der Message.

### **3.13 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 https Interface Adressen

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

Erzeugung eines Zahlungs-URL:

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

Prüfung einer Autorisationsantwort:

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

Verbuchung einer Zahlung:

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

Aus Sicherheitsgründen werden die Parameter ACTION und AMOUNT von der PayComplete Adresse des HI nicht unterstützt.

### 4.2 Transaktionsverlauf

Grundsätzlich läuft eine Transaktion nach folgendem Schema ab:

1. Erzeugung des Zahlungs-URL (CreatePayInit)
2. Die Zahlung wird eigenständig vom Kunden über Saferpay abgewickelt.
3. Bei Erfolg wird der SUCCESSLINK mit der Autorisationsantwort aufgerufen.
4. Die Antwort sollte auf Plausibilität/Manipulation geprüft werden (VerifyPayConfirm).
5. Abschließend muss die Zahlung verbucht werden, um den Geldfluss über den Tagesabschluss einzuleiten. Die Verbuchung kann entweder manuell über das Saferpay Backoffice oder automatisiert mit der PayComplete-Nachricht erfolgen.

### 4.3 Erzeugung des Zahlungs-URL über die CreatePayInit-Adresse

Das Shop-System übermittelt die Saferpay PayInit-Parameter per GET oder POST an das HI. Dort wird mit Hilfe der jeweiligen Saferpay Account Signatur ein verschlüsselter Zahlungs-URL erzeugt und wieder an das Shop-System zurück geliefert. Anschließend kann der so erzeugte Zahlungs-URL zum Bezahlen verwendet werden, in dem er etwa als Link oder Button in die Shop-Seite eingefügt wird.

Beispielaufruf (GET):

```
https://www.saferpay.com/hosting/CreatePayInit.asp?AMOUNT=...
```

Das Ergebnis wird im Klartext ohne HTML-Tags zurückgeliefert:

```
https://www.saferpay.com/vt/Pay.asp?DATA=%3cIDP%20ACCOUNTID%3d%2299867%2d94913159%22%20ALLOWCOLLECT%3d%22no%22%20AMOUNT%3d%22100%22%20BACKLINK%3d%222e%22%20CURRENCY%3d%22DEM%22%20DELIVERY%3d%22no%22%20DESCRIPTION%3d%22Testkauf%20Warenkorb%22%20EXPIRATION%3d%2220010408%2012%3a13%3a50%22%20FAILLINK%3d%222e%22%20KEYID%3d%220%2d37217%2dea645c3f3f0911d583d70050da413f31%22%20MSGTYPE%3d%22PayInit%22%20SUCCESSLINK%3d%222e%22%20TOKEN%3d%22ea645c5d3f0911d583d70050da413f31%22%2f%3e&SIGNATURE=2f1ec1fa51002817941c22e98b9047422ba9ff8fce8b61dab8208a5aa8c82be7cda02ff8a66930481fc19b16d05e7bcedd2b0e5be98fecad3d48bd43916a502f
```

Im Falle eines Fehlers wird die Kennung „ERROR“ zusammen mit einer Fehlerbeschreibung zurückgeliefert:

```
ERROR: Missing AMOUNT attribute
```

### 4.4 Prüfung der Autorisationsantwort über die VerifyPayConfirm-Adresse

Nach erfolgreicher Zahlung wird der SUCCESSLINK aufgerufen und mit ihm werden die Parameter DATA und SIGNATURE per GET an das Shop-System zurückgeliefert. Der Parameter DATA enthält die PayConfirm-Nachricht mit den Details der Autorisationantwort und der Parameter SIGNATURE den Schlüssel, mit dem DATA von Saferpay signiert wurde. Um eine Manipulation der Autorisationsantwort auszuschließen, sollten beide Parameter unmittelbar nach Erhalt an die VerifyPayConfirm-Adresse des Saferpay HI gesendet werden, das die Anfrage mit OK oder ERROR beantwortet.

#### 4.5 Beispiel VerifyPayConfirm

Die Parameter DATA und SIGNATURE werden per GET oder POST an das Saferpay Gateway gesendet:

Beispielaufruf (GET):

```
https://www.saferpay.com/hosting/VerifyPayConfirm.asp?DATA=...
```

Stimmt die digitale Signatur mit den Werten aus DATA überein, wird die positive Überprüfung mit OK unter Angabe der Saferpay ID und TOKEN angezeigt:

```
OK: ID=56a77rg243asfhmkq3r&TOKEN=%3e235462FA23C4FE4AF65...
```

Im Falle eines Fehlers wird der Text "ERROR" zusammen mit einer Fehlerbeschreibung zurückgeliefert:

```
ERROR: Possible manipulation
```

#### 4.6 Verbuchung einer Zahlung über die PayComplete-Adresse

Das Verbuchen einer Autorisierung erfolgt über den Aufruf der CreatePayComplete-Adresse des HI. Hierfür müssen die Parameter ACCOUNTID und ID per GET oder POST an das Saferpay HI gesendet werden. Nach Erhalt der PayComplete-Nachricht antwortet das HI mit OK oder ERROR.

Eine Betrag reduzierende Verbuchung oder das Stornieren einer Zahlung über die CreatePayComplete-Adresse des HI sind aus Sicherheitsgründen nicht möglich. Die Parameter ACTION und AMOUNT werden deshalb ignoriert.

Beispielaufruf (GET):

```
https://www.saferpay.com/hosting/PayCompleteV2.asp?ACCOUNTID=99867-94913159&ID=5sfhmkq3rg54345abcd234&spPassword=XAjc3Kna*
```

Konnte die Operation erfolgreich durchgeführt werden, wird OK zurückgeliefert:

```
OK
```

Im Falle eines Fehlers wird der Text ERROR zusammen mit einer Fehlerbeschreibung zurückgeliefert:

```
ERROR: Error description
```

*\* Die Übergabe des Parameters spPassword ist nur beim Testkonto erforderlich. Für produktive Konten wird dieser Parameter nicht benötigt!*

## 4.7 Aufruf der Payment Page über die Redirect-Adresse

Manchmal ist es nicht möglich die Antwortdaten vom Saferpay https Interface entgegenzunehmen, weil die Webserver-Einstellungen es nicht zulassen oder vielleicht ein Shop-System eingesetzt wird, das ausschließlich mit JavaScript arbeitet. In solch einem Fall kann die Redirect-Adresse des HI zum Aufruf der PP verwendet werden.

### 4.7.1 Nachteile von Redirect

Da bei diesem Verfahren die Zahlungsdaten auf der Web-Seite hinterlegt werden müssen, besteht theoretisch die Möglichkeit, dass diese von versierten Internet-Nutzern manipuliert werden. Ebenso ließe sich die PayConfirm-Nachricht verändern oder simulieren.

*Verwenden Sie die Redirect-Adresse nur, wenn die Gegebenheiten es nicht anders zulassen.  
Wenn möglich, sollte immer die CreatePayInit-Adresse angesprochen werden!*

### 4.7.2 https Interface Adresse

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

Aufruf der Payment Page per Redirect:

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

### 4.7.3 Transaktionsverlauf mit Redirect

Die zur Zahlung notwendigen PayInit-Parameter werden per GET oder POST an das Saferpay HI übermittelt. Daraufhin wird auf dem HI ein Zahlungs-URL erzeugt und die Payment Page unmittelbar per „redirect“ aufgerufen. Nachdem der Kunde die Zahlung über die PP erfolgreich abgewickelt hat, wird die Autorisationsantwort (PayConfirm-Nachricht) zunächst an das Saferpay HI gesendet. Dort wird die diese auf Plausibilität und mögliche Manipulation überprüft (VerifyPayConfirm) und anschließend das Ergebnis per GET mit dem SUCCESSLINK an das Shop-System übermittelt. Dem Shop-System wird das Ergebnis der Prüfung mit dem Parameter RESULT angezeigt. RESULT kann die Werte „0“ für Erfolg oder „1“ bei Manipulation oder einem Fehler enthalten.

Beispielaufruf (GET):

<https://www.saferpay.com/hosting/Redirect.asp?AMOUNT=1095&...>

Durch den Aufruf wird der Kunde über seinen Browser direkt zur PP geleitet, um darüber die Zahlung abzuwickeln.

Nach erfolgter Zahlung findet auf dem HI die Überprüfung der digitalen Signatur statt. Anschließend wird das Ergebnis per GET an den SUCCESSLINK weitergeleitet. Bei erfolgreicher Überprüfung wird das Ergebnis mit RESULT=0 angezeigt und weitere PayConfirm-Parameter mit den Autorisationsdetails sind verfügbar:

[http://www.shop.de/kasse\\_ok.php?RESULT=0&ID=J89HBV...](http://www.shop.de/kasse_ok.php?RESULT=0&ID=J89HBV...)

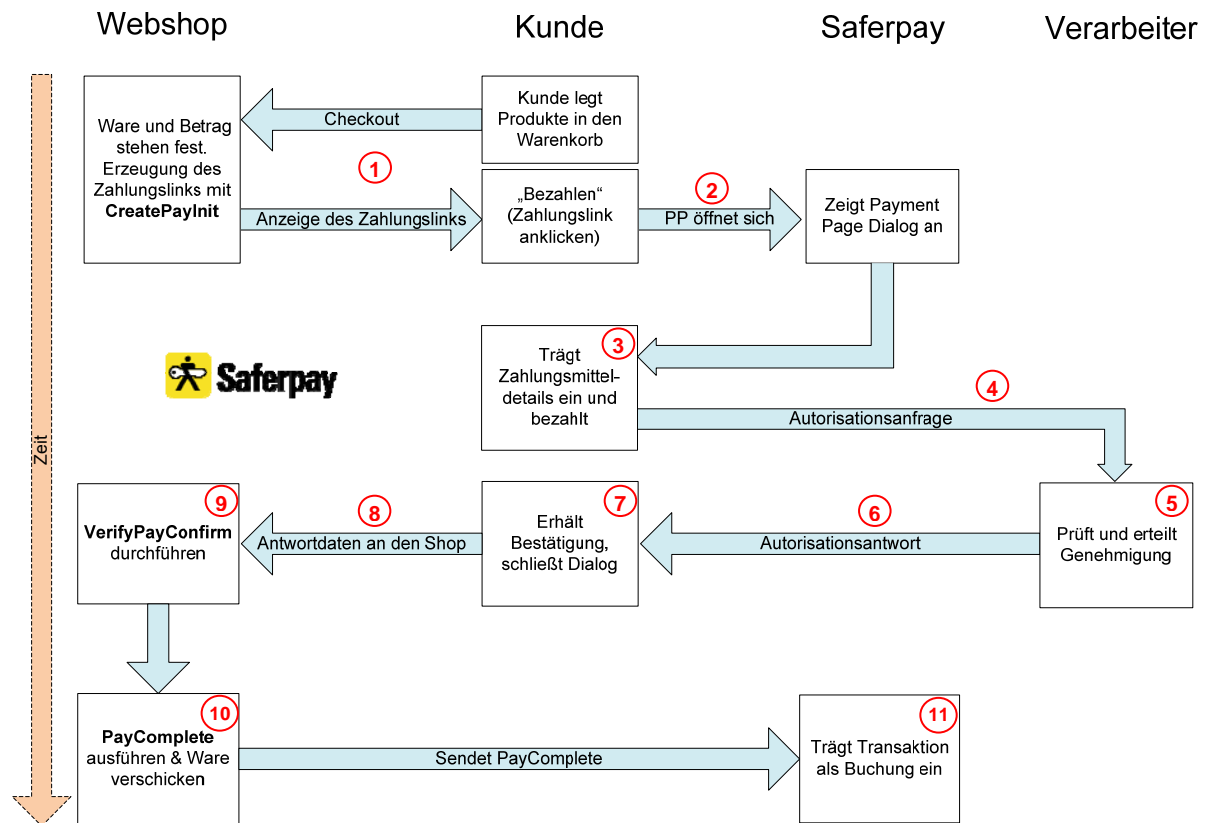
Schlägt die Überprüfung der digitalen Signatur fehl wird nur das Ergebnis RESULT=1 zurückgeliefert:

[http://www.shop.de/kasse\\_ok.pl?Session=123&RESULT=1](http://www.shop.de/kasse_ok.pl?Session=123&RESULT=1)

## 5 Verarbeitungsschritte

### 5.1 Übersicht

Die folgende Grafik zeigt den Ablauf einer erfolgreichen Online Zahlung über die Saferpay Payment Page:



### 5.2 Prozessbeschreibung

- ① Sobald der Kunde seinen Warenkorb gefüllt hat und der Zahlungsbetrag feststeht, erzeugt der Webshop den Zahlungslink (CreatePayInit). Der Webshop stellt auf der Bestellbestätigung den Zahlungslink in Form eines „Bezahlen“ Buttons oder ähnlich dar.
- ② Der Kunde klickt auf „Bezahlen“ und die PP öffnet sich.
- ③ Der Dialog der PP gibt dem Kunden ein Zahlungsmittel vor oder dieser wählt dort das Zahlungsmittel für die Zahlungsabwicklung selbst aus und gibt die verlangten Zahlungsmitteldetails ein.
- ④ Anschließend wird die Online Autorisation des Zahlungsmittels durchgeführt.
- ⑤ Der Verarbeiter prüft die Autorisationsanfrage und erteilt die Genehmigung.
- ⑥ Der Verarbeiter übermittelt die Autorisationsantwort.
- ⑦ In der PP wird eine Bestätigung der Zahlung angezeigt und der Kunde wird aufgefordert den Vorgang durch Klicken auf „Beenden“ abzuschließen.
- ⑧ Die PP wird geschlossen und der Kunde über den SUCCESSLINK in den Shop zurück geleitet. Der Shop erhält die Antwortdaten der Autorisation.

- 9 Das Händlersystem prüft die Zahlungsbestätigung (VerifyPayConfirm) und speichert sie zusammen mit den Auftragsinformationen ab.
- 10 Der Betrag wird verbucht (PayComplete) und die Bestellung kann ausgeführt werden.

*Das Verbuchen einer Reservation ist für den Tagesabschluss zwingend erforderlich. 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.*

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



## 6 Parameter

### 6.1 PayInit Parameter

Die Tabelle listet die Parameter, die für die CreatePayInit Message zur Verfügung stehen. Wenn nicht anders erwähnt ist die Verwendung des jeweiligen Parameters Pflicht.

Parameter	Format	Beschreibung
ACCOUNTID	ns[..15]	Die Saferpay Kontonummer des Händlers für diese Transaktion. Zum Beispiel "99867-94913159" für das Saferpay Testkonto.
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"
DESCRIPTION	ans[..50]	Enthält eine Beschreibung des Angebots, die in der PP und nach erfolgreicher Zahlung auch in den Details der Transaktion im Saferpay Backoffice angezeigt wird. Eine Formatierung der Anzeige in der PP über Steuerzeichen ist nicht möglich. Die Verwendung von mehr als 50 Zeichen wird deshalb, auch wenn es technisch möglich ist, nicht empfohlen. Der übergebene Wert muss zur korrekten Verarbeitung von Sonderzeichen zwingend HTML-kodiert werden.
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. Zulange 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.
VTCONFIG	an[..20]	<i>Optional</i> VTCONFIG bestimmt die zu verwendende Konfiguration für die PP. Es können verschiedene Konfigurationen für die PP im Saferpay Backoffice erstellt werden. Einige Optionen der PP, wie das Einfügen eines Logos können nur dort aktiviert werden. Wert: Name der Konfiguration im Saferpay Backoffice.
SUCCESSLINK	ans[..1024]	URL an den der Kunde nach erfolgreichem Abschluss der Reservation per Browser Redirect weitergeleitet wird. Die Bestätigungsnachricht (PayConfirm) wird von Saferpay an diese Adresse per GET angehängt.
FAILLINK	ans[..1024]	URL an den der Kunde per Browser Redirect weitergeleitet wird, wenn die Autorisation nicht durchgeführt werden konnte.
BACKLINK	ans[..1024]	URL an den der Kunde per Browser Redirect weitergeleitet wird, wenn er die Transaktion abbricht.

Parameter	Format	Beschreibung
NOTIFYURL	ans[..1024]	<p><i>Optional</i></p> <p>Voll qualifizierter URL, der bei erfolgreicher Autorisation von den Saferpay Servern direkt aufgerufen wird und die Bestätigungsmeldung (PayConfirm) per POST übermittelt. Der Aufruf ist nur über die Standardports für http (80) oder https (443) möglich. Andere Ports funktionieren nicht. Grundsätzlich wird empfohlen den NOTIFYURL zu verwenden, um die Rückmeldung beim Händlersystem, unabhängig von möglichen Fehlern oder Problemen beim Kunden, sicherzustellen. Um die Zuordnung einer Bestellung zu erleichtern hat sich die Übergabe einer Shop Session ID als GET Parameter an den NOTIFYURL bewährt.</p>
AUTOCLOSE	n[..2]	<p><i>Optional</i></p> <p>Schließt die PP nach erfolgreicher Autorisation automatisch und ruft den SUCCESSLINK auf. Der übergebene Wert gibt die Zeit in Sekunden an, die die Belegseite vor dem Schließen angezeigt wird. Empfohlener Wert: "0"</p>
CCNAME	a[..3]	<p><i>Optional</i></p> <p>Aktiviert das Feld zur Eingabe des Karteninhabernamens in der PP. Werte: "yes" oder "no"</p>
NOTIFYADDRESS	ans[..50]	<p><i>Optional</i></p> <p>Email-Adresse des Händlers. Saferpay sendet nach erfolgreicher Reservation eine Bestätigungs-Email an diese Adresse.</p>
USERNOTIFY	ans[..50]	<p><i>Optional</i></p> <p>Email-Adresse des Kunden. Saferpay sendet nach erfolgreicher Reservation eine Bestätigungs-Email an diese Adresse.</p>
LANGID	a[2]	<p><i>Optional</i></p> <p>Sprachen-Code gemäß ISO 639-1. Gibt die Sprache in der PP vor. Eine Liste der verfügbaren Sprachen-Codes findet sich unter <a href="http://support.saferpay.de/download/LanguageList.pdf">http://support.saferpay.de/download/LanguageList.pdf</a>. Ohne LANGID wird die Sprache des Kundenbrowsers in der PP angezeigt. Wird diese nicht erkannt, ist Deutsch die Standardsprache.</p>
SHOWLANGUAGES		<p>Aktiviert das Menü zur Auswahl der Sprache in der PP. Werte: "yes" oder "no"</p>
PROVIDERSET	ns[..40]	<p><i>Optional</i></p> <p>Gibt die, in der PP angezeigten Zahlungsmittel vor. Standardmäßig werden in der PP alle aktivierten Zahlungsmittel eines Terminals angezeigt. PROVIDERSET werden eine oder mehrere durch Komma getrennte Provider IDs übergeben. Die Provider IDs für das Saferpay Testkonto lauten 90 (Saferpay Testkarte) und 634 (Giropay). Die Provider IDs Ihres Terminals erhalten Sie von <a href="mailto:integration@saferpay.com">integration@saferpay.com</a>.</p>
DURATION	n[14]	<p><i>Optional</i></p> <p>Begrenzt die Gültigkeitsdauer des Zahlungslinks. Nach Ablauf wird der Link ungültig. Format: YYYYMMDDhhmmss</p>

Parameter	Format	Beschreibung
CARDREFID	ans[..40]	<i>Optional</i> Ersatzwert für Kreditkartennummer und Verfalldatum oder Bankverbindung (nur deutsches ELV). Der Ersatzwert kann von der E-Commerce-Applikation oder Saferpay (new) vergeben werden. Verwendung nur mit dem Service „Saferpay Secure Card Data“ möglich. Werte: Eindeutiger Ersatzwert oder "new"
DELIVERY	a[..3]	<i>Optional</i> Gibt an, ob ein Adressformular angezeigt wird. Werte: "yes" (default), "no"
APPEARANCE	an[..7]	<i>Optional</i> Passt das Erscheinungsbild der PP dem anzeigenden Display an. Werte: "auto" (default), "mobile", "desktop"
<b>Adressoptionen:</b>		
ADDRESS	a[..8]	<i>Optional</i> Gibt vor, ob ein Adressformular für Liefer-, Kunden- oder Rechnungsadresse wird. Werte: "DELIVERY", "CUSTOMER", "BILLING"
Liegt die Adresse bereits vor, kann sie an das Adressformular übergeben werden. Sonderzeichen in den Werten der folgenden Parameter müssen HTML-kodiert (als Entity oder Unicode) übergeben werden:		
COMPANY	ans[..50]	<i>Optional</i> Firmenname
GENDER	a[1]	<i>Optional</i> Geschlecht Werte: "f", "m", "c" (company)
FIRSTNAME	ans[..50]	<i>Optional</i> Vorname
LASTNAME	ans[..50]	<i>Optional</i> Nachname
STREET	ans[..50]	<i>Optional</i> Straße
ZIP	an[..10]	<i>Optional</i> Postleitzahl
CITY	ans[..50]	<i>Optional</i> Ort
COUNTRY	a[2]	<i>Optional</i> Länderkennung nach ISO 3166. Eine Liste der verfügbaren Länderkennungen findet sich unter <a href="http://support.saferpay.de/download/CountryList.pdf">http://support.saferpay.de/download/CountryList.pdf</a> .
EMAIL	ans[..50]	<i>Optional</i> Email-Adresse
PHONE	ns[..20]	<i>Optional</i> Telefonnummer

\* Für die Verwendung von CARDREFID="new" muss bei Saferpay zunächst ein numerischer Startwert für den Account hinterlegt werden. Kontaktieren Sie hierfür [integration@saferpay.com](mailto:integration@saferpay.com).

## 6.2 PayConfirm Parameter

Folgende Parameter können von der PayConfirm Nachricht übermittelt werden:

Parameter	Format	Beschreibung
MSGTYPE	a[..30]	Enthält immer den Wert "PayConfirm".
VTVERIFY	ans[..40]	Kann zusätzliche Informationen zur Transaktionsverarbeitung enthalten. Standardwert: "(obsolete)"
KEYID	ans[..40]	Kennung des Schlüssels mit dem die Signatur erzeugt wurde.
ID	an[28]	Eindeutige Saferpay Transaktionskennung.
TOKEN	ans[..40]	Kann zusätzliche Informationen zur Transaktionsverarbeitung enthalten. Standardwert: "(unused)"
ACCOUNTID	ns[..15]	Die Saferpay Kontonummer des Händlers für diese Transaktion. Zum Beispiel "99867-94913159" für das Saferpay Testkonto.
AMOUNT	n[..8]	Zahlungsbetrag in kleinster Währungseinheit, zum Beispiel "1230" entspricht dem Betrag 12,30 in Euro.
CURRENCY	a[3]	Dreistelliger ISO 4217 Währungs-Code, zum Beispiel "CHF" oder "EUR".
CARDREFID	ans[..40]	<i>Optional</i> (nur wenn der Parameter beim CreatePayInit Aufruf mit übergeben wurde.) Enthält den Ersatzwert für Kreditkartennummer und Verfalldatum oder die Bankverbindung (nur deutsches ELV),
SCDRESULT	n[..4]	<i>Optional</i> (nur wenn der Parameter CARDREFID beim CreatePayInit Aufruf angegeben wurde) Enthält den Antwort-Code der Registrierung im SCD: 0 Registrierung erfolgreich. 7000 Allgemeiner Fehler (siehe DESCRIPTION). 7001 Anfrage konnte nicht vollständig verarbeitet werden. 7002 Kartentyp auf dem Terminal nicht verfügbar. 7003 Parameter mit ungültigem Inhalt oder Format. 7004 CARDREFID nicht gefunden (nur bei Autorisierung). 7005 Fehlender Parameter in der Anfrage. 7006 CARDREFID existiert bereits. 7007 Keine Berechtigung für SCD vorhanden.
PROVIDERID	n[..4]	Enthält die Provider ID des Zahlungsmittelverarbeiters.
PROVIDERNAME	ans[..30]	Enthält den Namen des Zahlungsmittelverarbeiters.
ORDERID	an[..39]	<i>Optional</i> Die Referenznummer des Händlers, die beim CreatePayInit übergeben wurde.
IP	ns[..15]	<i>Optional*</i> Enthält die IP Adresse des Kunden. Nur bei vorhandenem Saferpay Risk Management verfügbar
IPCOUNTRY	a[2]	<i>Optional*</i> Herkunftsland der IP-Adresse des Bezahlers nach ISO 3166. Länder-Code (Bsp.: CH, DE, AT). Ist eine Zuordnung nicht möglich lautet der Wert "IX". Beispiel: "DE"
CCCOUNTRY	a[2]	<i>Optional*</i> Herkunftsland der Karte nach ISO 3166. Ist eine Zuordnung nicht möglich lautet der Wert "IX". Beispiel: "DE"

Parameter	Format	Beschreibung
MPI_LIABILITYSHIFT	a[..3]	<p><i>Optional**</i> Gibt an, ob technisch formal eine Haftungsumkehr besteht. Werte: "yes" oder "no"</p> <div style="border: 1px solid red; padding: 5px;"> <p><b>Achtung!</b> 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> </div>
ECI	n[1]	<p><i>Optional**</i> Electronic Commerce Indicator Kennzeichnet 3-D Secure Transaktionen („Verified by Visa“, „MasterCard SecureCode“): 0 = Internet Zahlung ohne Haftungsumkehr. 1 = 3-D Secure Zahlung mit Authentifizierung. 2 = 3-D Secure Zahlung, Karte nimmt am Verfahren nicht teil.</p>
XID	ans[28]	<p><i>Optional**</i> Diese Base64-Zeichenfolge wird vom MPI vergeben und referenziert auf den Vorgang im 3-D Secure Protokoll.</p>
CAVV	ans[28]	<p><i>Optional**</i> Cardholder Authentication Verification Value Bei einer MasterCard ist hier der UCAF-Wert enthalten. Saferpay benutzt unabhängig vom Kreditkartentyp den Wert CAVV.</p>

\* Nur verfügbar bei vorhandenem Saferpay Risk Management

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

### 6.3 PayComplete Parameter

Folgende Parameter stehen für eine CreatePayComplete Message zur Verfügung. Wenn nicht anders erwähnt ist die Verwendung des jeweiligen Parameters Pflicht:

Parameter	Format	Beschreibung
ID	an[28]	Saferpay Transaktionskennung der PayConfirm Nachricht. Pflichtparameter, es sei denn ACTION=CloseBatch.
AMOUNT	n[..8]	Zu verbuchender 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 PayComplete Anfrage. 0 = Anfrage erfolgreich ausgeführt. ≠0 = Anfrage nicht erfolgreich ausgeführt.
MESSAGE	ans[..30]	Enthält eine Antwort zur Verbuchungsanfrage in Textform.
AUTHMESSAGE	ans[..30]	Kann eine Antwort zur Verbuchungsanfrage in Textform enthalten.

## 7 Saferpay Testkonto

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

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

Kartennummer	Beschreibung
Kontonummer: 12345674 BLZ: 76026000	Giropay Testkonto Testkonto für giropay Zahlungen.
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 eine Ablehnung der Zahlungsanfrage zu erzwingen. Alle Anfragen, bei denen die letzten zwei Stellen des Betrags (AMOUNT) ungleich Null "00" sind werden abgelehnt.

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 numerisch sein muss und das Verfalldatum in der Zukunft liegt. 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

#### Zahlungslink mit CreatePayInit erzeugen:

```

MessageFactory mf = new MessageFactory();
mf.Open(""); // Saferpay configuration path, e.g. "c:\\Programme\\Saferpay\\Client"
mo_payinit = mf.CreatePayInit();

string m_accountid = "99867-94913159";
string m_amount = "2095";
string m_currency = "EUR";
string m_description = "Test Einkauf";
string m_address = "no";
string m_orderid = "0815-4711";
string m_backlink = "http://www.myshop.com/back.aspx";
string m_faillink = "http://www.myshop.com/Fail.aspx";
string m_successlink = "http://www.myshop.com/Success.aspx";
string m_notifyurl = "http://www.myshop.com/notify.aspx";

mo_payinit.SetAttribute("ACCOUNTID", m_accountid);
mo_payinit.SetAttribute("AMOUNT", m_amount);
mo_payinit.SetAttribute("CURRENCY", m_currency);
mo_payinit.SetAttribute("DELIVERY", m_address);
mo_payinit.SetAttribute("ORDERID", Server.HtmlEncode(m_orderid));
mo_payinit.SetAttribute("DESCRIPTION", Server.HtmlEncode(m_description));
mo_payinit.SetAttribute("SUCCESSLINK", m_successlink);
mo_payinit.SetAttribute("BACKLINK", m_backlink);
mo_payinit.SetAttribute("FAILLINK", m_faillink);
mo_payinit.SetAttribute("NOTIFYURL", m_notifyurl);

string paymenturl = mo_payinit.GetPostUrl();
string data = mo_payinit.GetPostData();
string signature = mo_payinit.GetPostSignature();
  
```

#### Zahlungslink aus einem Formular aufrufen:

```

<html>
<head><title>Zahlung mit der Saferpay Payment Page</title></head>
<body>
<h2>Beispiel - Saferpay Payment Page per POST aufrufen</h2>
  <form action="<%=paymenturl %>"method="POST">
    <input type="hidden" name="DATA" value="<%=data %>">
    <input type="hidden" name="SIGNATURE" value="<%=signature %>" >
    <input type="submit" value="Bezahlen">
  </form>
</body>
</html>
  
```



### Prüfung der PayConfirm Nachricht:

```
string data = Request.QueryString.Get("DATA");
string signature = Request.QueryString.Get("SIGNATURE");

MessageFactory mf = new MessageFactory();
mf.Open("");
mo_payconfirm = mf.VerifyPayConfirm(data, signature);

string id = mo_payconfirm.GetAttribute(ID);
```

### Verbuchung der Zahlung mit CreatePayComplete:

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

mo_paycomplete.SetAttribute("ID", id);
mo_paycomplete.SetAttribute("ACCOUNTID", m_accountid);

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

#### Zahlungslink mit CreatePayInit erzeugen:

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

MessageFactory mf = new MessageFactory();
mf.Open(""); // Saferpay configuration path, e.g. "c:\\Programme\\Saferpay\\Client"
MessageObject mo_payinit = mf.CreatePayInit();

String m_accountid = "99867-94913159";
String m_amount = "2095";
String m_currency = "EUR";
String m_description = "Test Einkauf";
string m_address = "no";
String m_orderid = "0815-4711";
String m_backlink = "http://www.myshop.com/back.jsp";
String m_faillink = "http://www.myshop.com/Fail.jsp";
String m_successlink = "http://www.myshop.com/Success.jsp";
String m_notifyurl = "http://www.myshop.com/notify.jsp";

mo_payinit.SetAttribute("ACCOUNTID", m_accountid);
mo_payinit.SetAttribute("AMOUNT", m_amount);
mo_payinit.SetAttribute("CURRENCY", m_currency);
mo_payinit.SetAttribute("DELIVERY", m_address);
mo_payinit.SetAttribute("ORDERID", StringEscapeUtils.escapeHtml(m_orderid));
mo_payinit.SetAttribute("DESCRIPTION", StringEscapeUtils.escapeHtml(m_description));
mo_payinit.SetAttribute("SUCCESSLINK", m_successlink);
mo_payinit.SetAttribute("BACKLINK", m_backlink);
mo_payinit.SetAttribute("FAILLINK", m_faillink);
mo_payinit.SetAttribute("NOTIFYURL", m_notifyurl);

String paymenturl = mo_payinit.GetPostUrl();
String data = mo_payinit.GetPostData();
String signature = mo_payinit.GetPostSignature();
```

#### Zahlungslink aus einem Formular aufrufen:

```
<html>
<head><title>Zahlung mit der Saferpay Payment Page</title></head>
<body>
<h2> Beispiel - Saferpay Payment Page per POST aufrufen</h2>
  <form action="<%=paymenturl %>"method="POST">
    <input type="hidden" name="DATA" value="<%=data %>">
    <input type="hidden" name="SIGNATURE" value="<%=signature %>" >
    <input type="submit" value="Bezahlen">
  </form>
</body>
</html>
```

#### Prüfung der PayConfirm Nachricht:

```
import Saferpay.*

String data = request.getParameter("DATA");
String signature = request.getParameter("SIGNATURE");

MessageFactory mf = new MessageFactory();
mf.Open("");
MessageObject mo_payconfirm = mf.VerifyPayConfirm(data, signature);

String id = mo_payconfirm.GetAttribute(ID);
```



### Verbuchung der Zahlung mit CreatePayComplete:

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

mo_paycomplete.SetAttribute("ID", id);
mo_paycomplete.SetAttribute("ACCOUNTID", m_accountid);

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

### Zahlungslink mit CreatePaylnit erzeugen:

```
java -jar Saferpay.jar -payinit -p C:\Programme\Saferpay\Client -a AMOUNT 1930 -a CURRENCY EUR
-a DELIVERY no -a ACCOUNTID 99867-94913159 -a DESCRIPTION "Test Einkauf" -a ORDERID 0815-4711
-a FAILLINK "http://www.testshop.de/fail.php" -a SUCCESSLINK
"http://www.testshop.de/success.php" -a BACKLINK "http://www.testshop.de/back.php" -a
NOTIFYURL "http://www.testshop.de/log.php"
```

### Erzeugter Zahlungslink:

```
https://www.saferpay.com/vt/Pay.asp?DATA=%3CIDP+ALLOWCOLLECT%3D%22no%22+EXPIRATION%3D%22201103
25+16%3A06%3A53%22+DESCRIPTION%3D%22Test+Einkauf%22+BACKLINK%3D%22http%3A%2F%2Fwww.testshop.de
%2Fback.php%22+AMOUNT%3D%221930%22+DELIVERY%3D%22no%22+ACCOUNTID%3D%2299867-
94913159%22+SUCCESSLINK%3D%22http%3A%2F%2Fwww.testshop.de%2Fsuccess.php%22+CURRENCY%3D%22EUR%2
2+ORDERID%3D%220815-
4711%22+FAILLINK%3D%22http%3A%2F%2Fwww.testshop.de%2Ffail.php%22+MSGTYPE%3D%22PayInit%22+KEYID
%3D%220-99867-
959af13b7be94dd99cbd20ac7caa5888%22+NOTIFYURL%3D%22http%3A%2F%2Fwww.testshop.de%2Flog.php%22+T
OKEN%3D%228ae94d50562c01e0d24085728e033992%22+2F%3E&SIGNATURE=38FE7F5483578D24EDC3F9A8D00CC1B8
7733B5CE1F984F82E2ED084C52B01B0E783C8007F022610D93E26E2AA254C02245A09A0F25A0C5A4961B9FDFB9FF8F
50
```

### Prüfung der PayConfirm Nachricht:

### Rücksprung nach erfolgreicher Zahlung in den Shop zum SUCCESSLINK:

```
http://www.testshop.de/success.php?DATA=%3CIDP+MSGTYPE%3d%22PayConfirm%22+TOKEN%3d%22%28unused
%29%22+VTVERIFY%3d%22%28obsolete%29%22+KEYID%3d%221-
0%22+ID%3d%22br1b01AS3AphUA4fnAr0bQUS7thA%22+ACCOUNTID%3d%2299867-
94913159%22+PROVIDERID%3d%2290%22+PROVIDERNAME%3d%22Saferpay+Test+Card%22+ORDERID%3d%220815-
4711%22+AMOUNT%3d%221930%22+CURRENCY%3d%22EUR%22+IP%3d%22193.247.180.193%22+IPCOUNTRY%3d%22CH%
22+CCCOUNTRY%3d%22XX%22+MPI_LIABILITYSHIFT%3d%22yes%22+MPI_XID%3d%22NEpiXQIKWz8xBBsZF38FVUUwTA
o%3d%22+ECI%3d%222%22+XID%3d%22NEpiXQIKWz8xBBsZF38FVUUwTAo%3d%22+%2f%3E&SIGNATURE=b43f0ac94ce2
60fa520010558dc552ef1c67a839538bd2346d6e593fa74b3e2db22e6de0f494a312bf3436af662219144dda2bf472
a5447da205009668a791fa
```

### Empfangenes DATA:

```
<IDP MSGTYPE="PayConfirm" TOKEN="(unused)" VTVERIFY="(obsolete)" KEYID="1-0"
ID="br1b01AS3AphUA4fnAr0bQUS7thA" ACCOUNTID="99867-94913159" PROVIDERID="90"
PROVIDERNAME="Saferpay Test Card" ORDERID="0815-4711" AMOUNT="1930" CURRENCY="EUR"
IP="193.247.180.193" IPCOUNTRY="CH" CCCOUNTRY="XX" MPI_LIABILITYSHIFT="yes"
MPI_XID="NEpiXQIKWz8xBBsZF38FVUUwTAo=" ECI="2" XID="NEpiXQIKWz8xBBsZF38FVUUwTAo=" />
```

### Empfangene SIGNATURE:

```
b43f0ac94ce260fa520010558dc552ef1c67a839538bd2346d6e593fa74b3e2db22e6de0f494a312bf3436af662219
144dda2bf472a5447da205009668a791fa
```

### Durchführung des VerifyPayConfirm:

```
java -jar Saferpay.jar -payconfirm -p C:\Programme\Saferpay\Client -d
%3CIDP+MSGTYPE%3d%22PayConfirm%22+TOKEN%3d%22%28unused%29%22+VTVERIFY%3d%22%28obsolete%29%22+K
EYID%3d%221-0%22+ID%3d%22br1b01AS3AphUA4fnAr0bQUS7thA%22+ACCOUNTID%3d%2299867-
94913159%22+PROVIDERID%3d%2290%22+PROVIDERNAME%3d%22Saferpay+Test+Card%22+ORDERID%3d%220815-
4711%22+AMOUNT%3d%221930%22+CURRENCY%3d%22EUR%22+IP%3d%22193.247.180.193%22+IPCOUNTRY%3d%22CH%
22+CCCOUNTRY%3d%22XX%22+MPI_LIABILITYSHIFT%3d%22yes%22+MPI_XID%3d%22NEpiXQIKWz8xBBsZF38FVUUwTA
o%3d%22+ECI%3d%222%22+XID%3d%22NEpiXQIKWz8xBBsZF38FVUUwTAo%3d%22+%2f%3E -s
b43f0ac94ce260fa520010558dc552ef1c67a839538bd2346d6e593fa74b3e2db22e6de0f494a312bf3436af662219
144dda2bf472a5447da205009668a791fa
```

Verbuchung der Zahlung mit CreatePayComplete:

```
java -jar Saferpay.jar -capture -p C:\Programme\Saferpay\Client -i  
brlb01AS3AphUA4fnAr0bQUS7thA -a ACCOUNTID 99867-94913159 -of capt.txt
```



## 8.5 https Interface

### Zahlungslink mit CreatePayInit erzeugen:

```
https://www.saferpay.com/hosting/CreatePayInit.asp?spPassword=XAjc3Kna&ACCOUNTID=99867-94913159&ORDERID=123456789-001&AMOUNT=1000&CURRENCY=EUR&DESCRIPTION=Testkauf&SUCCESSLINK="http://www.myshop.com/Success.aspx"&FAILLINK="http://www.myshop.com/Fail.aspx"&BACKLINK="http://www.myshop.com/back.aspx"&NOTIFYURL="http://www.myshop.com/notify.aspx"
```

### Antwort liefert den Zahlungs-URL:

```
https://www.saferpay.com/vt/Pay.asp?DATA=%3CIDP+MSGTYPE%3d%22PayInit%22+MSG_GUID%3d%22939a4c930b5c482588d91f54f74ac110%22+CLIENTVERSION%3d%222.0%22+KEYID%3d%220-99867-7d5a273c0f5043e28811e764d6433086%22+TOKEN%3d%22bbf6577cd8e74d65a27f084c9cfe2592%22+ALLOWCOLLECT%3d%22no%22+DELIVERY%3d%22no%22+EXPIRATION%3d%220110625+12%3a01%3a56%22+ACCOUNTID%3d%2299867-94913159%22+AMOUNT%3d%221000%22+CURRENCY%3d%22EUR%22+DESCRIPTION%3d%22Testkauf%22+SUCCESSLINK%3d%22http%3a%2f%2fwww.myshop.com%2fSuccess.aspx%22+BACKLINK%3d%22http%3a%2f%2fwww.myshop.com%2fback.aspx%22+FAILLINK%3d%22http%3a%2f%2fwww.myshop.com%2fFail.aspx%22+ORDERID%3d%22123456789-001%22+CCNAME%3d%22yes%22+NOTIFYURL%3d%22http%3a%2f%2fwww.myshop.com%2fnotify.aspx%22+%2f%3e&SIGNATURE=428b356c87f7fdcf44417f670197c4a6395385e623e224653610b94db8acclead509b7b5e6dfc465dcf987b3cf4b284fc799ee93ff9cb151c3bc9981e8320232
```

### Überprüfung der PayConfirm-Nachricht:

### Rücksprung nach erfolgreicher Autorisation in den Shop zum SUCCESSLINK:

```
http://www.myshop.com/Success.aspx?DATA=%3CIDP+MSGTYPE%3d%22PayConfirm%22+TOKEN%3d%22%28unused%29%22+VTVERIFY%3d%22%28obsolete%29%22+KEYID%3d%221-0%22+ID%3d%22A668MSAprOj4tAzv7G9lAQufUr3A%22+ACCOUNTID%3d%2299867-94913159%22+PROVIDERID%3d%2290%22+PROVIDERNAME%3d%22Saferpay+Test+Card%22+ORDERID%3d%22123456789-001%22+AMOUNT%3d%221000%22+CURRENCY%3d%22EUR%22+IP%3d%22193.247.180.193%22+IPCOUNTRY%3d%22CH%22+CCCOUNTRY%3d%22XX%22+MPI_LIABILITYSHIFT%3d%22yes%22+MPI_TX_CAVV%3d%22AAABBIIFmAAAAAAAAAAAAAAAAAAAA%3d%22+MPI_XID%3d%22CxMTYwhoUXtCBAEndBULcRIQaAY%3d%22+ECI%3d%221%22+CAVV%3d%22AAABBIIFmAAAAAAAAAAAAAAAAAAAA%3d%22+XID%3d%22CxMTYwhoUXtCBAEndBULcRIQaAY%3d%22+%2f%3E&SIGNATURE=7b2bb163f4ef86d969d992b4e2d61ad48d3b9022e0ec68177e35fe53184e6b3399730d1a3641d2a984ce38699daad72ab006d5d6a9565c5ae1cff8bdc8a1eb63
```

### Empfangenes DATA:

```
<IDP MSGTYPE="PayConfirm" TOKEN="(unused)" VTVERIFY="(obsolete)" KEYID="1-0" ID="A668MSAprOj4tAzv7G9lAQufUr3A" ACCOUNTID="99867-94913159" PROVIDERID="90" PROVIDERNAME="Saferpay Test Card" ORDERID="123456789-001" AMOUNT="1000" CURRENCY="EUR" IP="193.247.180.193" IPCOUNTRY="CH" CCCOUNTRY="XX" MPI_LIABILITYSHIFT="yes" MPI_TX_CAVV="AAABBIIFmAAAAAAAAAAAAAAAAAAAA" MPI_XID="CxMTYwhoUXtCBAEndBULcRIQaAY" ECI="1" CAVV="AAABBIIFmAAAAAAAAAAAAAAAAAAAA" XID="CxMTYwhoUXtCBAEndBULcRIQaAY" />
```

### Empfangene SIGNATURE:

```
7b2bb163f4ef86d969d992b4e2d61ad48d3b9022e0ec68177e35fe53184e6b3399730d1a3641d2a984ce38699daad72ab006d5d6a9565c5ae1cff8bdc8a1eb63
```



**https Aufruf VerifyPayConfirm:**

```
https://www.saferpay.com/hosting/verifypayconfirm.asp?spPassword=XAjc3Kna&ACCOUNTID=99867-94913159&DATA=%3CIDP+MSGTYPE%3d%22PayConfirm%22+TOKEN%3d%22%28unused%29%22+VTVERIFY%3d%22%28obsolete%29%22+KEYID%3d%221-0%22+ID%3d%22A668MSAprOj4tAzv7G91AQUFUr3A%22+ACCOUNTID%3d%2299867-94913159%22+PROVIDERID%3d%2290%22+PROVIDERNAME%3d%22Saferpay+Test+Card%22+ORDERID%3d%22123456789-001%22+AMOUNT%3d%221000%22+CURRENCY%3d%22EUR%22+IP%3d%22193.247.180.193%22+IPCOUNTRY%3d%22CH%22+CCOUNTRY%3d%22XX%22+MPI_LIABILITYSHIFT%3d%22yes%22+MPI_TX_CAVV%3d%22AAABBIIFmAAAAAAAAAAAAAAAAAAAA%3d%22+MPI_XID%3d%22CxMTYwhoUXtCBAEndBULcRIQaAY%3d%22+ECI%3d%221%22+CAVV%3d%22AAABBIIFmAAAAAAAAAAAAAAAAAAAA%3d%22+XID%3d%22CxMTYwhoUXtCBAEndBULcRIQaAY%3d%22+%2f%3E&SIGNATURE=7b2bb163f4ef86d969d992b4e2d61ad48d3b9022e0ec68177e35fe53184e6b3399730d1a3641d2a984ce38699daad72ab006d5d6a9565c5ae1cff8bdc8a1eb63
```

**Antwort liefert Transaktionskennung:**

```
OK: ID=A668MSAprOj4tAzv7G91AQUFUr3A&TOKEN=(unused)
```

**Verbuchung der Zahlung mit CreatePayComplete:**

```
https://www.saferpay.com/hosting/paycompletev2.asp?spPassword=XAjc3Kna&ACCOUNTID=99867-94913159&ID=A668MSAprOj4tAzv7G91AQUFUr3A
```

**Antwort liefert das Ergebnis:**

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



## 9 Fehler-Codes

Folgende Antwort-Codes können durch die Saferpay Library zurückgeliefert werden:

Code	Name	Beschreibung
<b>Allgemeine Fehler</b>		
0x80040201	Context Missing	Kein Konfigurationskontext definiert. Rufen Sie Open() vor dem Aufruf anderer Methoden auf.
0x80040202	File Not Found	Datei konnte nicht gefunden werden. Überprüfen Sie, ob die Komponente richtig installiert ist.
0x80040203	File Access Error	Fehler beim Dateizugriff. Prüfen Sie die Zugriffsrechte auf das Konfigurationsverzeichnis.
0x80040204	Invalid File Format	Datei enthält keinen gültigen XML Inhalt.
0x80040206	Invalid Path	Ungültiger Pfad oder URL.
0x80040207	Invalid Option	Ungültige Parameterangabe.
0x80040208	Request Failed	Anfrage fehlgeschlagen.
0x80040209	Cryptographic Error	Verschlüsselungsfehler.
0x8004020f	No Configuration	Keine Konfigurationsdaten im Pfad von Open() vorhanden.
0x80040214	Verify Failed	Signatur ungültig. Überprüfung der Signatur fehlgeschlagen.
<b>HTTP Fehler</b>		
0x80042019	Invalid URL	Ungültige URL-Angabe.
0x80042021	DNS Error	Fehler bei der Auflösung einer Adresse.
0x80042xxx	HTTP Server Error	HTTP Server-Fehler. Die letzten 3 Zeichen sind der vom Server zurückgelieferte Status-Code.
<b>Socket Fehler</b>		
0x80043xxx	Socket Base	Die letzten 3 Zeichen sind der Socket Fehler-Code.



## 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](http://www.saferpay.com)

[integration@saferpay.com](mailto:integration@saferpay.com)

Saferpay Europa

**SIX Payment Services (Germany) GmbH**

Langenhorner Chaussee 92-94

22415 Hamburg

+49 40 325 967- 280

[www.saferpay.com](http://www.saferpay.com)

[integration@saferpay.com](mailto: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](http://www.saferpay.com)

[support@saferpay.com](mailto:support@saferpay.com)

Saferpay Europa

**SIX Payment Services (Germany) GmbH**

Langenhorner Chaussee 92-94

22415 Hamburg

+49 40 325 967- 250

[www.saferpay.com](http://www.saferpay.com)

[support@saferpay.com](mailto:support@saferpay.com)

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

