



Payment Services

# Integration Billpay Business



## Spezifikation

Version 2.2

## Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b> .....	<b>4</b>
1.1	Zusammenfassung .....	4
1.2	Voraussetzungen .....	4
1.3	Akzeptanz der Billpay Allgemeine Geschäfts- und Datenschutzbestimmungen .....	4
1.4	Formatangaben .....	5
<b>2</b>	<b>Saferpay Client Library</b> .....	<b>6</b>
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
<b>3</b>	<b>Klassen und Methoden der Client Library</b> .....	<b>8</b>
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 .....	9
3.5	Execute() Methode .....	9
3.6	CreatePayComplete() Methode .....	9
3.7	CreateRequest() Methode .....	9
3.8	SetAttribute() Methode .....	9
3.9	GetAttribute() Methode .....	9
3.10	Capture() Methode .....	9
<b>4</b>	<b>Saferpay https Interface</b> .....	<b>10</b>
4.1	Voraussetzungen .....	10
4.2	IP-Zugriff und Passwort für das Saferpay https Interface .....	10
4.3	https Interface Adressen .....	10
4.4	Übertragung von Nachrichten .....	11
<b>5</b>	<b>Verarbeitungsschritte</b> .....	<b>12</b>
5.1	Übersicht .....	12
5.2	Prozessbeschreibung .....	13
<b>6</b>	<b>Parameter</b> .....	<b>14</b>
6.1	Autorisationsanfrage .....	14
6.1.1	Optionale Warenkorbparameter .....	16
6.5	Autorisationsantwort .....	19
6.6	Verbuchungsanfrage .....	20
6.7	Verbuchungsantwort .....	20
<b>7</b>	<b>Antwort-Codes</b> .....	<b>21</b>
7.1	RESULT-Werte .....	21
7.2	AUTHRESULT-Werte .....	22
<b>8</b>	<b>Beispiele</b> .....	<b>24</b>
8.1	Wichtiger Hinweis .....	24



8.2	C# mit der .NET LIB .....	24
8.3	Java mit der Java LIB .....	27
8.4	https Interface .....	30
9	Kontakt .....	32
9.1	Saferpay Integration Team .....	32
9.2	Saferpay Support Team .....	32

# 1 Einleitung

## 1.1 Zusammenfassung

Dieses Dokument beschreibt die Integration von "Lastschrift mit Zahlungsgarantie" und "Kauf auf Rechnung" des Zahlungsmittelanbieters Billpay mit der Saferpay Client Library (LIB) und dem Saferpay https Interface (HI).

## 1.2 Voraussetzungen

Die Abwicklung von Billpay-Zahlungen mit Saferpay setzt Folgendes voraus:

- Eine entsprechende Lizenz und somit das Vorhandensein einer gültigen Kennung mit Benutzernamen 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 Billpay liegt vor.
- Der Einsatz einer Saferpay Client Library ab Version 2.0. Ältere LIBs werden nicht unterstützt. Alternativ kann auch das https Interface verwendet werden.
- Bevor im Händler-Shop Billpay-Zahlungen erfolgen können muss eine Abnahme durch Billpay erfolgen. Billpay führt hierfür zusammen mit dem Händler einen Test durch, der eine Zahlung im Shop und den anschließenden Rechnungslauf beinhaltet. Eine detaillierte Beschreibung der Abnahme stellt Billpay hier zur Verfügung:

<https://www.billpay.de/haendler/service/integration/abnahmetest/saferpay/>

## 1.3 Akzeptanz der Billpay Allgemeine Geschäfts- und Datenschutzbestimmungen

Billpay weist darauf hin, dass die Billpay Allgemeine Geschäfts- und Datenschutzbestimmungen vor dem Absenden der Autorisationsanfrage durch den Endkunden bestätigt werden müssen. Das Einblenden der AGB und Datenschutzbestimmungen muss nach folgendem Schema geschehen:

Mit der &Uuml;bermittlung der f&uuml;r die Abwicklung des Rechnungsaufs und einer Identit&auml;ts und Bonit&auml;tspr&uuml;fung erforderlichen Daten an die [Billpay GmbH](https://billpay.de/endkunden) bin ich einverstanden. Es gelten die [\[Link auf Datenschutzbestimmungen\]](#) von Billpay.

Der Platzhalter für den Verweis auf die Billpay-Datenschutzbestimmungen muss durch den entsprechenden Wert aus der folgenden Tabelle ersetzt werden. Dabei ist das Land, in dem die Rechnungsadresse des Kunden liegt maßgeblich für die Auswahl des Verweises.

Dokument	Webseite	Popup
Datenschutz (Deutschland)	<a href="https://www.billpay.de/kunden/agb#datenschutz">https://www.billpay.de/kunden/agb#datenschutz</a>	<a href="https://www.billpay.de/api/agb#datenschutz">https://www.billpay.de/api/agb#datenschutz</a>
Datenschutz (Schweiz)	<a href="https://www.billpay.de/kunden/agb-ch#datenschutz">https://www.billpay.de/kunden/agb-ch#datenschutz</a>	<a href="https://www.billpay.de/api/agb-ch#datenschutz">https://www.billpay.de/api/agb-ch#datenschutz</a>
Datenschutz (Österreich)	<a href="https://www.billpay.de/kunden/agb-at#datenschutz">https://www.billpay.de/kunden/agb-at#datenschutz</a>	<a href="https://www.billpay.de/api/agb-at#datenschutz">https://www.billpay.de/api/agb-at#datenschutz</a>

Die Datenschutzbestimmungen liegen in verschiedenen Sprachen vor. Soll zum Beispiel das Schweizer Dokument in französischer Sprache angezeigt werden, nutzen Sie dafür folgenden Link:

<https://www.billpay.de/kunden/agb-ch?lang=fr/#datenschutz>

Werden die Datenschutzbestimmungen vom Kunden nicht akzeptiert muss ein entsprechender Hinweis angezeigt und der Bestellvorgang darf nicht fortgesetzt werden.



#### 1.4 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](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 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"/>	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 API 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 CreatePayComplete(), um ein leeres 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 CreatePayComplete(String id, String token);
    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 Execute() Methode**

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

### **3.6 CreatePayComplete() Methode**

Mit CreatePayComplete() kann eine Reservation verbucht werden. CreatePayComplete() benötigt für eine Verbuchung immer ID und ACCOUNTID.

### **3.7 CreateRequest() Methode**

Erzeugt ein neues Request MessageObject vom angegebenen Nachrichtentyp (msgtype).

### **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 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 Voraussetzungen

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 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.

**Wichtig!** Bitte vergessen Sie nicht den Kollegen mitzuteilen, dass Sie Billpay-Zahlungen abwickeln möchten und nennen Sie Ihre Saferpay Kundennummer und die IP-Adresse(n) für den Zugriff auf das https Interface.

### 4.2 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.

### 4.3 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>

#### 4.4 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
```

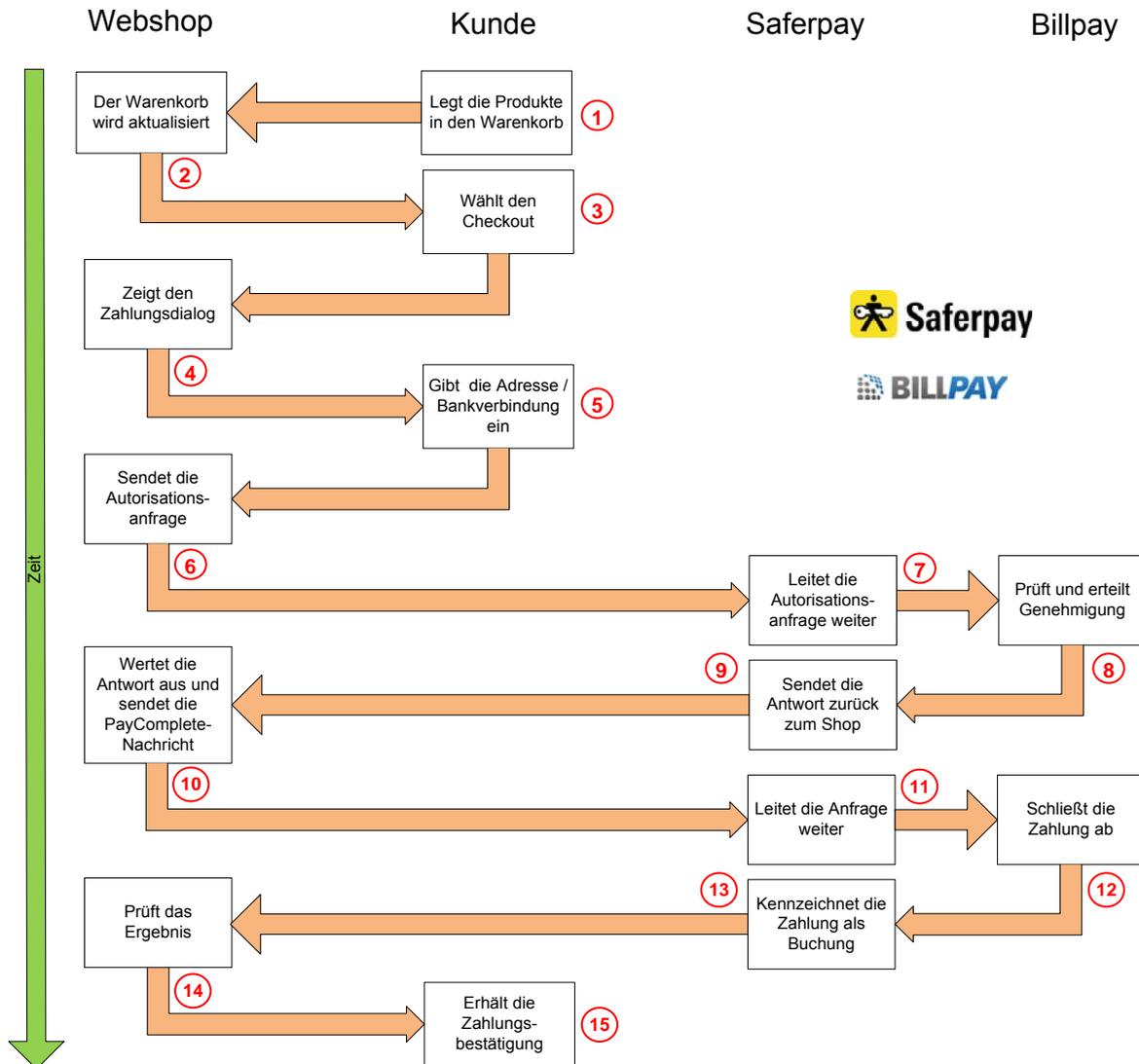
##### Hinweise

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 Billpay-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 seine Adressdaten ein oder übernimmt die vom Shop angezeigte Adresse. Bei "Lastschrift mit Zahlungsgarantie" erfolgt zusätzlich die Eingabe der Bankverbindung.
- ⑥ Der Webshop übermittelt die Zahlungsanfrage an Saferpay.
- ⑦ Saferpay leitet die Zahlungsanfrage zur Autorisation an Billpay weiter.
- ⑧ Billpay 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 Autorisationsantwort zu Auswertung und löst die Verbuchung der Transaktion mit der PayComplete-Nachricht aus.

*Das Verbuchen einer Reservation ist für den Geldfluss obligatorisch. Das Geld wird anschließend in Form einer Sammelposition dem Geschäftskonto des Händlers gutgeschrieben. Vom Zahlungsmittelverarbeiter Billpay erhält der Händler eine Abrechnungsliste.*

*Es wird empfohlen jede Billpay-Transaktion möglichst sofort, aber spätestens nach 21 Tagen zu verbuchen. Unmittelbar nach dem Verbuchen muss die Rechnungsstellung mit dem Zahlungsziel erfolgen. Der Händler tritt mit dem Verbuchen die Forderung an Billpay ab.*

- ⑪ Saferpay leitet den Aufruf zur Verbuchung an Billpay weiter.
- ⑫ Billpay verbucht die Transaktion und schließt damit den Zahlungsvorgang ab. Das Ergebnis wird an Saferpay übermittelt.
- ⑬ Der Status der Transaktion wechselt durch das PayComplete von „Reservation“ auf „Buchung“. Der Transaktionsstatus wird im Saferpay Backoffice Journal angezeigt. Saferpay sendet die Verbuchungsantwort an den Webshop.
- ⑭ Der Webshop überprüft das Ergebnis der Verbuchungsantwort.
- ⑮ Der Kunde erhält vom Shop seine Zahlungsbestätigung.

## 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.
ACTION	a[..6]	<i>Optional</i> Kennzeichnet die Zahlungsanfrage als Buchung oder Gutschrift. Werte: "Debit" (Standard, Kunde bezahlt) und "Credit" (Kunde erhält Geld).
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[..40]	ORDERID enthält die Referenznummer für eine Zahlung. Die ORDERID muss eindeutig vergeben werden. Zahlungsanfragen mit bereits verwendeter ORDERID können von Billpay abgelehnt werden.
REFID	an[28]	<i>Gutschriftparameter</i> Enthält die Saferpay ID der vorausgegangenen Buchung. Es kann wahlweise mit REFID oder REFOID auf eine Buchung referenziert werden. Einer der Parameter ist bei einer Gutschrift obligatorisch.
REFOID	ans[..40]	<i>Gutschriftparameter, Pflicht wenn ACTION="Credit"</i> Enthält die Saferpay ORDERID der vorausgegangenen Buchung. Es kann wahlweise mit REFID oder REFOID auf eine Buchung referenziert werden. Einer der Parameter ist bei einer Gutschrift obligatorisch.
PAYMENTTYPE	a[..11]	Kennzeichnet die Zahlungsanfrage als Rechnung oder Lastschrift. Werte: "POB" (Purchase On Bill, Rechnung) und "DDG" (Direct Debit Guaranteed, Lastschrift mit Zahlungsgarantie)
<b>Kontodaten des Kunden</b>		
NAME	ans[..100]	<i>Optional, Pflichtparameter bei PAYMENTTYPE Lastschrift</i> Kontoinhabername
ACCOUNTNUMBER	n[..10]	<i>Optional, Pflichtparameter bei PAYMENTTYPE Lastschrift</i> Kontonummer
BANK_CODE_NUMBER	n[8]	<i>Optional, Pflichtparameter bei PAYMENTTYPE Lastschrift</i> Bankleitzahl

Rechnungsadresse		
GENDER	a[1]	Geschlecht Werte: "f" (female), "m" (male)
FIRSTNAME	ans[..50]	Vorname
LASTNAME	ans[..50]	Nachname
STREET	ans[..50]	Straße
ADDRESSADDITION	an[..50]	Zusatz zur Adresse
ZIP	an[..10]	Postleitzahl
CITY	ans[..50]	Ort
COUNTRY	a[2]	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]	Email-Adresse
PHONE	ns[..50]	Telefonnummer
DATEOFBIRTH	n[8]	Geburtsdatum (YYYYMMDD)
LANGID	a[2]	Sprachen-Code gemäß ISO 639-1. Gibt die Sprache des Kunden vor. Eine Liste möglicher Sprachen-Codes findet sich unter <a href="http://support.saferpay.de/download/LanguageList.pdf">http://support.saferpay.de/download/LanguageList.pdf</a> .
COMPANY	ans[..50]	Firmenname Indikator für B2B-Zahlungen. Bei Anfragen ohne COMPANY handelt es sich automatisch um B2C-Zahlungen.
IP	ns[..15]	Enthält die IP-Adresse des Kunden zur Ermittlung des Herkunftslandes mittels Geotargeting.
LEGALFORM	a[..4]	Rechtsform der Firma Sofern GENDER nicht "f" oder "m" bei der Anfrage gesetzt wird, ist LEGALFORM ein Pflichtparameter für B2B-Zahlungen. Ansonsten handelt es sich automatisch um B2C-Zahlungen. Werte: "gmbh" (GmbH), "ag" (AG), "misc" (sonstige)
Lieferadresse*		
DELIVERY_GENDER	a[1]	Geschlecht Werte: "f" (female), "m" (male)
DELIVERY_FIRSTNAME	ans[..50]	Vorname
DELIVERY_LASTNAME	ans[..50]	Nachname
DELIVERY_STREET	ans[..50]	Straße
DELIVERY_ADDRESSADDITION	an[..50]	Zusatz zur Adresse
DELIVERY_ZIP	an[..10]	Postleitzahl
DELIVERY_CITY	ans[..50]	Ort
DELIVERY_COUNTRY	a[2]	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> .
DELIVERY_PHONE	ns[..50]	Telefonnummer

\* Weicht die Lieferadresse von der Rechnungsadresse ab, wird die Zahlung von Billpay in der Regel abgelehnt. Sollte das für Ihr Geschäftsmodell nicht akzeptabel sein, klären Sie diesen Punkt bitte direkt mit Billpay.

### 6.1.1 Optionale Warenkorbparameter

Optional können jeder Buchung oder Gutschrift XML-Dokumente mit den Warenkorbdetails der Bestellung übergeben werden. Die Warenkorbdetails kann Billpay verwenden, um Forderungen gegenüber dem Shopper bei Zahlungsschwierigkeiten geltend zu machen. Liegen Billpay die Warenkorbdetails nicht vor, müssen sie zunächst vom Händler angefordert werden, was für diesen mitunter Mehraufwand bedeuten kann. Es folgt eine Liste der XML-Dokumente und deren XML-Attribute:

Parameter	Beschreibung	
BASKETDATA	<i>Buchungsparameter (ACTION="Debit")</i> Enthält die Bestelldaten des Artikels. Es können beliebig viele Artikel übergeben werden.	
XML-Attribut	Format	Beschreibung
articleid	an[..20]	Artikelnummer
articlequantity	n[..7]	Artikelmenge im Warenkorb
articlename	an[..50]	Artikelname
articledescription	an[..50]	Artikelbeschreibung
articleprice	n[..7]	Artikelpreis netto in kleinster Währungseinheit. Zum Beispiel: "1000" entspricht Betrag 10,00 in Euro.
articlepricegross	n[..7]	Artikelpreis brutto in kleinster Währungseinheit. Zum Beispiel: "1190" entspricht Betrag 11,90 in Euro.

Beispiel:

BASKETDATA =

```
<article_data>
  <article articleid="article001" articlequantity="12"
    articlename="Kaffetasse 08/15" articledescription="Eine Kaffetasse aus Porzellan"
    articleprice="120" articlepricegross="195" />
  <article articleid="article002" articlequantity="4" articlename="Kuchenteller 07-11"
    articledescription="Kuchenteller - weiss" articleprice="85"
    articlepricegross="100" />
</article_data>
```

Parameter	Beschreibung	
BASKETTOTAL	<i>Buchungsparameter (ACTION="Debit")</i> Enthält die Bestelldaten insgesamt.	
XML-Attribut	Format	Beschreibung
shippingname	an[..50]	Versandart Zum Beispiel: "Express-Versand"
shippingprice	n[..7]	Nettowert aller Zusatzkosten für den Kunden (Lieferkosten, Zuschläge, ...) in kleinster Währungseinheit. Zum Beispiel: "20000 " entspricht Betrag 200,00 in Euro.
shippingpricegross	n[..7]	Bruttowert aller Zusatzkosten für den Kunden (Lieferkosten, Zuschläge, ...) in kleinster Währungseinheit. Zum Beispiel: "23800 " entspricht Betrag 238,00 in Euro.
rebate	n[..7]	Nettowert aller Betrag mindernden Posten für den Kunden (Rabatte, Gutscheine, ...) in kleinster Währungseinheit. Zum Beispiel: "405 " entspricht Betrag 4,05 in Euro.
rebategross	n[..7]	Bruttowert aller Betrag mindernden Posten für den Kunden (Rabatte, Gutscheine, ...) in kleinster Währungseinheit. Zum Beispiel: "500 " entspricht Betrag 5,00 in Euro.
carttotalprice	n[..7]	Nettowert der Bestellung ohne Zusatzkosten in kleinster Währungseinheit. Zum Beispiel: "60000 " entspricht Betrag 600,00 in Euro.
carttotalpricegross	n[..7]	Bruttowert der Bestellung ohne Zusatzkosten in kleinster Währungseinheit. Zum Beispiel: "71400 " entspricht Betrag 714,00 in Euro.
currency	a[3]	Dreistelliger ISO 4217 Währungs-Code. Zum Beispiel: "CHF" oder "EUR"

Beispiel:

BASKETTOTAL =

```
<total shippingname="Express-Versand" shippingprice="500" shippingpricegross="650"
  rebate="500" rebategross="595" carttotalprice="1780" carttotalpricegross="2795"
  currency="EUR" />
```

Parameter	Beschreibung	
BASKETDATAREFUND	<i>Gutschriftparameter (ACTION="Credit")</i> Enthält die Teilretourdaten der Artikel.	
XML-Attribut	Format	Beschreibung
articleid	an[..20]	Artikelnummer
articlequantity	n[..7]	Anzahl der Artikel, die storniert werden sollen.

Beispiel:

BASKETDATAREFUND =

```
< canceled_articles >
  < article articleid="article001" articlequantity="1" />
  < article articleid="article002" articlequantity="1" />
</ canceled_articles >
```

Parameter	Beschreibung	
BASKETTOTALREFUND	<i>Gutschriftparameter (ACTION="Credit")</i> Enthält die Parameter der Teilretourdaten.	
XML-Attribut	Format	Beschreibung
rebatedecrease	n[..7]	Nettoreduzierung aller Betrag mindernden Posten für den Kunden in kleinster Währungseinheit. Zum Beispiel: "405 " entspricht Betrag 4,05 in Euro.
rebatedecreasegross	n[..7]	Bruttoreduzierung aller Betrag mindernden Posten für den Kunden (Rabatte, Gutscheine, ...) in kleinster Währungseinheit. Zum Beispiel: "500 " entspricht Betrag 5,00 in Euro.
shippingdecrease	n[..7]	Nettoreduzierung aller Zusatzkosten für den Kunden (Lieferkosten, Zuschläge, ...) in kleinster Währungseinheit. Zum Beispiel: "20000 " entspricht Betrag 200,00 in Euro.
shippingdecreasegross	n[..7]	Bruttoreduzierung aller Zusatzkosten für den Kunden (Lieferkosten, Zuschläge, ...) in kleinster Währungseinheit. Zum Beispiel: "23800 " entspricht Betrag 238,00 in Euro.

Beispiel:

BASKETTOTALREFUND =

```
<cancel_params reference="123456789" rebatedecrease="0" rebatedecreasegross="0"
    shippingdecrease="0" shippingdecreasegross="0" />
```

## 6.5 Autorisationsantwort

Die folgende Tabelle listet die Parameter, die in der Autorisationsantwort bei "Lastschrift mit Zahlungsgarantie" und "Kauf auf Rechnung" enthalten sind.

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.
AUTHRESULT	n[..3]	Enthält den Antwort-Code von Billpay. Ist keine Verbindung zum Verarbeiter zustande gekommen, weist der Wert in RESULT darauf hin.
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.
PROVIDERID	n[..4]	Enthält die Provider ID des Zahlungsmittelverarbeiters.
PROVIDERNAME	ans[..30]	Enthält den Namen des Zahlungsmittelverarbeiters.
ACQUIRER_TERMINALID	an[..10]	Die Terminalkennung des Zahlungsmittelverarbeiters.
AUTHDATE	ns[17]	Das Autorisationsdatum (YYYYMMDD hh:mm:ss).
AUTHCODE	ans[..40]	Enthält bei erfolgreicher Autorisation den Autorisations-Code des Zahlungsmittelverarbeiters.
STREET*	ans[..50]	Korrigierte Straße
ZIP*	an[..10]	Korrigierte Postleitzahl
CITY*	ans[..50]	Korrigierter Ort
COUNTRY*	a[2]	Korrigiertes Land nach ISO 3166.
PAYMENT_PROTOCOL	ans[..30]	Der Name des Zahlungsmittelprotokolls, das der Verbindung zugrunde liegt.
CARDHOLDERMESSAGE	ans[..255]	Kann im Fehlerfall eine Antwort für den Kunden enthalten.
AUTHMESSAGE	ans[..30]	Enthält eine Antwort zur Autorisation in Textform.
POB_ACCOUNTHOLDER	ans[..50]	<i>Nur bei "Kauf auf Rechnung"</i> Kontoinhaber für die Forderung (in der Regel "Billpay GmbH").
POB_ACCOUNTNUMBER	n[..10]	<i>Nur bei "Kauf auf Rechnung"</i> Billpay-Kontonummer für die Forderung.
POB_BANKCODE	n[8]	<i>Nur bei "Kauf auf Rechnung"</i> Billpay-Bankleitzahl für die Forderung.
POB_BANKNAME	ans[..50]	<i>Nur bei "Kauf auf Rechnung"</i> Billpay-Bankinstitut für die Forderung.
POB_PAYERNOTE	ans[..80]	<i>Nur bei "Kauf auf Rechnung"</i> Verwendungszweck

\* Billpay korrigiert mitunter die Adresse, übernimmt aber für die Richtigkeit dieser Korrektur keine Haftung. Der Händler kann selbst entscheiden, ob er die korrigierte Adresse verwendet.

## 6.6 Verbuchungsanfrage

Durch folgende Parametern wird die Verbuchung mit der CreatePayComplete Nachricht ausgelöst:

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
AMOUNT	n[.8]	Zahlungsbetrag in kleinster Währungseinheit. Der Parameter wird nur für eine betragsreduzierte Verbuchung benötigt und darf keinen größeren Wert, als bei der Autorisation enthalten.
ACCOUNTID	ns[.15]	Die Saferpay Kontonummer des Händlers für diese Transaktion. Zum Beispiel "99867-94913159" für das Saferpay Testkonto.
POB_DELAY	n[.2]	<i>Optional, nur für "Kauf auf Rechnung" verfügbar</i> Gibt die Anzahl Tage an, um die sich das Zahlungsziel zusätzlich zu den üblichen 20 oder 30 Tagen verschiebt. Der Standardwert ist "0".

## 6.7 Verbuchungsantwort

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.
POB_DUEDATE	n[8]	<i>Nur bei "Kauf auf Rechnung"</i> Enthält das Zahlungsziel des Kunden (YYYYMMDD).

## 7 Antwort-Codes

### 7.1 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
5	Access denied	Der Zugriff mit dieser IP-Adresse ist für den Account nicht freigegeben.
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. Das Feld AUTHRESULT kann nähere Hinweise liefern.
104	PAN black listed	Die Bankverbindung 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.

## 7.2 AUTHRESULT-Werte

Für den Fall, dass eine Autorisationsanfrage von Billpay abgelehnt wird, enthält die Autorisationsantwort den Parameter AUTHRESULT. Einige Ablehnungen werden von Billpay mit dem Status "DENIED" versehen. In diesen Fällen darf dem Kunden die Zahlart Billpay nicht für eine Wiederholung angeboten werden. Die folgende Tabelle liefert die Erklärungen für die AUTHRESULT-Werte für den Händler und den Kunden. Den Erklärungen für den Händler ist jeweils ein „(m)“ vorangestellt, den Erklärungen für den Kunden ein „(c)“. Die Ablehnungen mit dem Status "DENIED" finden sich am Ende der Tabelle.

Wert	Erklärung
9	(c) Bitte überprüfen Sie Ihre Daten. Der Vorname Ihrer Rechnungsadresse muss aus mindestens zwei Buchstaben bestehen. (m) Bitte überprüfen Sie Ihre Daten. Der Vorname Ihrer Rechnungsadresse muss aus mindestens zwei Buchstaben bestehen.
10	(c) Bitte überprüfen Sie Ihre Daten. Der Nachname Ihrer Rechnungsadresse muss aus mindestens zwei Buchstaben bestehen. (m) Bitte überprüfen Sie Ihre Daten. Der Nachname Ihrer Rechnungsadresse muss aus mindestens zwei Buchstaben bestehen.
11	(c) Bitte überprüfen Sie Ihre Daten. Die Straße Ihrer Rechnungsadresse muss aus mindestens zwei Buchstaben bestehen. (m) Bitte überprüfen Sie Ihre Daten. Die Straße Ihrer Rechnungsadresse muss aus mindestens zwei Buchstaben bestehen.
12	(c) Bitte überprüfen Sie Ihre Daten. Die Hausnummer Ihrer Rechnungsadresse muss aus mindestens einer Ziffer bestehen. (m) Bitte überprüfen Sie Ihre Daten. Die Hausnummer Ihrer Rechnungsadresse muss aus mindestens einer Ziffer bestehen.
13	(c) Bitte überprüfen Sie Ihre Daten. Die PLZ Ihrer Rechnungsadresse muss aus genau [Länge] Ziffern bestehen. (m) Bitte überprüfen Sie Ihre Daten. Die PLZ Ihrer Rechnungsadresse muss aus genau [Länge] Ziffern bestehen.
14	(c) Bitte überprüfen Sie Ihre Daten. Der Ort Ihrer Rechnungsadresse muss aus mindestens zwei Buchstaben bestehen. (m) Bitte überprüfen Sie Ihre Daten. Der Ort Ihrer Rechnungsadresse muss aus mindestens zwei Buchstaben bestehen.
15	(c) Bitte überprüfen Sie Ihre Daten. Die Telefonnummer Ihrer Rechnungsadresse ist nicht korrekt. (m) Bitte überprüfen Sie Ihre Daten. Die Telefonnummer Ihrer Rechnungsadresse ist nicht korrekt.
16	(c) Bitte überprüfen Sie Ihre Daten. Die Handynummer Ihrer Rechnungsadresse ist nicht korrekt. (m) Bitte überprüfen Sie Ihre Daten. Die Handynummer Ihrer Rechnungsadresse ist nicht korrekt.
17	(c) Bitte überprüfen Sie Ihre Daten. Ihre E-Mail-Adresse ist nicht korrekt. (m) Bitte überprüfen Sie Ihre Daten. Ihre E-Mail-Adresse ist nicht korrekt.

Wert	Erklärung
18	(c) Bitte überprüfen Sie Ihre Daten. Ihr Geburtsdatum darf nicht in der Zukunft liegen. (m) Bitte überprüfen Sie Ihre Daten. Ihr Geburtsdatum darf nicht in der Zukunft liegen.
19	(c) Bitte wählen Sie eine andere Zahlart. Sie müssen mind. 18 Jahre alt sein, um diese Zahlart von Billpay zu nutzen. (m) Bitte wählen Sie eine andere Zahlart. Sie müssen mind. 18 Jahre alt sein, um diese Zahlart von Billpay zu nutzen.
<b>Status "DENIED" Antwort-Codes:</b>	
29	(c) Bitte wählen Sie eine andere Zahlart. Leider können wir Ihnen für diese Transaktion die ausgewählte Zahlart von Billpay nicht anbieten. (m) Die IP-Adresse kommt aus einem Land, das auf der Blacklist steht.
32	(c) Bitte wählen Sie eine andere Zahlart. Leider können wir Ihnen für diese Transaktion die ausgewählte Zahlart von Billpay nicht anbieten. (m) Der Kunde wurde aufgrund fehlgeschlagener Identitäts- bzw. Bonitätsprüfung abgelehnt.
41	(c) Bitte wählen Sie eine andere Zahlart. Leider können wir Ihnen für diese Transaktion die ausgewählte Zahlart von Billpay nicht anbieten. (m) Von der IP-Adresse wurden zu viele Anfragen gesendet.
42	(c) Bitte wählen Sie eine andere Zahlart. Leider können wir Ihnen für diese Transaktion die ausgewählte Zahlart von Billpay nicht anbieten. (m) Der Kunde hat offene Forderungen.
<b>Status "DENIED" Antwort Codes:</b>	
43	(c) Bitte wählen Sie eine andere Zahlart. Leider können wir Ihnen für diese Transaktion die ausgewählte Zahlart von Billpay nicht anbieten. (m) Das Rechnungslimit des Kunden bei Billpay ist erreicht.
44	(c) Bitte wählen Sie eine andere Zahlart. Leider können wir Ihnen für diese Transaktion die ausgewählte Zahlart von Billpay nicht anbieten. (m) Der Kunde hat offene Forderungen beim Händler.
45	(c) Bitte wählen Sie eine andere Zahlart. Leider können wir Ihnen für diese Transaktion die ausgewählte Zahlart von Billpay nicht anbieten. (m) Externe Identitäts- oder Bonitätsprüfung fehlgeschlagen.
70	(c) Bitte wählen Sie eine andere Zahlart. Leider können wir Ihnen für diese Transaktion die ausgewählte Zahlart von Billpay nicht anbieten. (m) Die angegebene E-Mail-Adresse ist ungültig.

## 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");

`##### Parameter "Lastschrift mit Zahlungsgarantie" #####
request.SetAttribute("PAYMENTTYPE", "DDG");
request.SetAttribute("NAME, Server.HtmlEncode("Stefanie Müller"));
request.SetAttribute("ACCOUNTNUMBER", "12345674");
request.SetAttribute("BANK_CODE_NUMBER", "76026000");
`##### Parameter "Kauf auf Rechnung" #####
request.SetAttribute("PAYMENTTYPE", "POB");
`##### Die restlichen Parameter sind identisch #####
request.SetAttribute("ACCOUNTID", "99867-94913159");
request.SetAttribute("AMOUNT", "12500");
request.SetAttribute("CURRENCY", "EUR");
request.SetAttribute("ORDERID", "123456789");
request.SetAttribute("GENDER", "f");
request.SetAttribute("FIRSTNAME, Server.HtmlEncode("Stefanie"));
request.SetAttribute("LASTNAME, Server.HtmlEncode("Müller"));
request.SetAttribute("STREET, Server.HtmlEncode("Hohe Chaussee 254"));
request.SetAttribute("ZIP", "20225");
request.SetAttribute("CITY", Server.HtmlEncode("Hamburg"));
request.SetAttribute("EMAIL", "stefanie.mueller@saferpay.com");
request.SetAttribute("LANGID", "de");
request.SetAttribute("COUNTRY", "DE");
request.SetAttribute("DATEOFBIRTH", "19170511");
`##### Optionale Warenkorbparameter #####
request.SetAttribute("BASKETDATA", "<article_data><article[XML-Attribute]/></article_data>");
request.SetAttribute("BASKETTOTAL", "<total[XML-Attribute]/>");

MessageObject response = request.Execute();

```



### Autorisationsanfrage Gutschrift

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

request.SetAttribute("ACTION", "Credit");
`##### "Lastschrift mit Zahlungsgarantie" #####
request.SetAttribute("PAYMENTTYPE", "DDG");
`##### "Kauf auf Rechnung" #####
request.SetAttribute("PAYMENTTYPE", "POB");
`##### Die restlichen Parameter sind identisch #####
request.SetAttribute("ACCOUNTID", "99867-94913159");
request.SetAttribute("AMOUNT", "12500");
request.SetAttribute("CURRENCY", "EUR");
`##### Entweder Referenzierung auf die ORDERID #####
request.SetAttribute("REFOID", "123456789");
`##### Oder Referenzierung auf die ID #####
request.SetAttribute("REFID", "zh8G9pAj89fUUA70StpzAOSYKldb");
`##### Optionale Warenkorbparameter #####
request.SetAttribute("BASKETDATAREFUND", "<anceled_articles><article[XML-Attribute]/>
</article_data>");
request.SetAttribute("BASKETTOTALREFUND", "<cancel_params[XML-Attribute]/>");

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 response = mo_paycomplete.Capture();
```



### Verbuchungssantwort

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

### 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", "7500");

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();
```

### 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");

`##### Parameter "Lastschrift mit Zahlungsgarantie" #####
request.SetAttribute("PAYMENTTYPE", "DDG");
request.SetAttribute("NAME, StringEscapeUtils.escapeHtml ("Stefanie Müller"));
request.SetAttribute("ACCOUNTNUMBER", "12345674");
request.SetAttribute("BANK_CODE_NUMBER", "76026000");
`##### Parameter "Kauf auf Rechnung" #####
request.SetAttribute("PAYMENTTYPE", "POB");
`##### Die restlichen Parameter sind identisch #####
request.SetAttribute("ACCOUNTID", "99867-94913159");
request.SetAttribute("AMOUNT", "12500");
request.SetAttribute("CURRENCY", "EUR");
request.SetAttribute("ORDERID", "123456789");
request.SetAttribute("GENDER", "f");
request.SetAttribute("FIRSTNAME, StringEscapeUtils.escapeHtml ("Stefanie"));
request.SetAttribute("LASTNAME, StringEscapeUtils.escapeHtml ("Müller"));
request.SetAttribute("STREET, StringEscapeUtils.escapeHtml ("Hohe Chaussee 254"));
request.SetAttribute("ZIP", "20225");
request.SetAttribute("CITY", StringEscapeUtils.escapeHtml ("Hamburg"));
request.SetAttribute("EMAIL", "stefanie.mueller@saferpay.com");
request.SetAttribute("LANGID", "de");
request.SetAttribute("COUNTRY", "DE");
request.SetAttribute("DATEOFBIRTH", "19170511");
`##### Optionale Warenkorbparameter #####
request.SetAttribute("BASKETDATA", "<article_data><article[XML-Attribute]/></article_data>");
request.SetAttribute("BASKETTOTAL", "<total[XML-Attribute]/>");

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



### Autorisationsanfrage Gutschrift

```
import Saferpay.*;

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

request.SetAttribute("ACTION", "Credit");
`##### "Lastschrift mit Zahlungsgarantie" #####
request.SetAttribute("PAYMENTTYPE", "DDG");
`##### "Kauf auf Rechnung" #####
request.SetAttribute("PAYMENTTYPE", "POB");
`##### Die restlichen Parameter sind identisch #####
request.SetAttribute("ACCOUNTID", "99867-94913159");
request.SetAttribute("AMOUNT", "12500");
request.SetAttribute("CURRENCY", "EUR");
`##### Entweder Referenzierung auf die ORDERID #####
request.SetAttribute("REFOID", "123456789");
`##### Oder Referenzierung auf die ID #####
request.SetAttribute("REFID", "zh8G9pAj89fUUA70StpzAOSYK1db");
`##### Optionale Warenkorbparameter #####
request.SetAttribute("BASKETDATAREFUND", "<canceled_articles><article[XML-Attribute]>
</article_data>");
request.SetAttribute("BASKETTOTALREFUND", "<cancel_params[XML-Attribute]>");

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 response = mo_paycomplete.Capture();
```



### Verbuchungssantwort

```
int result = response.GetAttribute("RESULT");
if (result == 0)
{
    String ddate = response.GetAttribute("POB_DUEDATE");
    Console.WriteLine("Verbuchung erfolgreich!");
}
else
{
    Console.WriteLine("Verbuchung fehlgeschlagen! RESULT=" + result);
    return;
}
```

### 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();
```

## 8.4 https Interface

Aufgrund der Datenmenge empfiehlt sich für die Autorisationsanfrage die Übergabe der Daten an das HI mittels POST.

### Autorisationsanfrage Buchung

```
<form method="post"
action="https://www.saferpay.com/hosting/execute.asp">
  <input type="hidden" name="spPassword" value="XAJc3Kna">
  <input type="hidden" name="ACCOUNTID" value="99867-94913159">
  <input type="hidden" name="AMOUNT" value="1000">
  <input type="hidden" name="CURRENCY" value="EUR">
  <input type="hidden" name="ORDERID" value="Billpay Lastschrift">
  <input type="hidden" name="PAYMENTTYPE" value="DDG">
  <input type="hidden" name="LANGID" value="de">
  <input type="hidden" name="COUNTRY" value="DE">
  <p>
  <input type="text" size="40" name="NAME"></p>
  <p>
  <input type="text" size="10" name="ACCOUNTNUMBER"></p>
  <p>
  <input type="text" size="8" name="BANK_CODE_NUMBER"></p>
  <p>
  <select name="GENDER" size="2">
  <option value="m">männlich</option>
  <option value="f">weiblich</option>
  </select></p>
  <input type="text" size="40" name="FIRSTNAME">
  <input type="text" size="40" name="LASTNAME">
  <input type="text" size="40" name="STREET">
  <input type="text" size="5" name="ZIP">
  <input type="text" size="40" name="CITY">
  <input type="text" size="40" name="EMAIL">
  <input type="text" size="8" name="DATEOFBIRTH">

  <input type="submit" name="SUBMIT" value="Bezahlen...">
</form>
```

### Autorisationsanfrage Gutschrift

```
https://www.saferpay.com/hosting/execute.asp?spPassword=XAJc3Kna_
&ACTION=Credit_
&PAYMENTTYPE=DDG_
&ACCOUNTID=99867-94913159_
&AMOUNT=1000_
&CURRENCY=EUR_
&REFOID="Billpay Lastschrift" _
```

### Autorisationsantwort

```
OK:<IDP RESULT="0" MSGTYPE="AuthorizationResponse" ID="Gp6UWObtSifMSAvfr9E0AfEEQdUb"
AUTHMESSAGE="request was processed successfully" AUTHCODE="1d452e4b-e285-485d-87d0-
cdfce54c970d" PROVIDERID="1199" PROVIDERNAME="Billpay Lastschrift Test" ORDERID="Billpay
Lastschrift" IP="127.0.0.1" AUTHDATE="20110511 07:51:29" STREET=" Hohe Chaussee 254"
ZIP="20225" CITY="Hamburg" COUNTRY="DE" />
```



### Verbuchung einer Zahlung mit CreatePayComplete:

[https://www.saferpay.com/hosting/paycomplete.asp?spPassword=XAjc3Kna\\_&ACCOUNTID=99867-94913159\\_&ID="Gp6UWObtSifMSAvfr9E0AfEEQdUb"](https://www.saferpay.com/hosting/paycomplete.asp?spPassword=XAjc3Kna_&ACCOUNTID=99867-94913159_&ID=)

### Verbuchungsantwort:

OK:<IDP RESULT="0" MSGTYPE="PayConfirm" MESSAGE="request was processed successfully" AUTHMESSAGE=" " ID="Gp6UWObtSifMSAvfr9E0AfEEQdUb" POB\_DUEDATE="20110608" />

### Betragsreduzierte Verbuchung einer Zahlung mit CreatePayComplete

<https://www.saferpay.com/hosting/paycompletev2.asp?spPassword=XAjc3Kna&ACCOUNTID=99867-94913159&ID=8IK98fbtS4d7UAzMGCEAEQ12vbA&AMOUNT=500>

### Stornierung einer Zahlung mit CreatePayComplete

<https://www.saferpay.com/hosting/paycompletev2.asp?spPassword=XAjc3Kna&ACCOUNTID=99867-94913159&ID=EvrKOEApM3YtSapnE0MlAU28nCYb&ACTION=Cancel>

## 9 Kontakt

### 9.1 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)

### 9.2 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!*

