Payment Services

Authorization Interface

Specification

Version 5.2

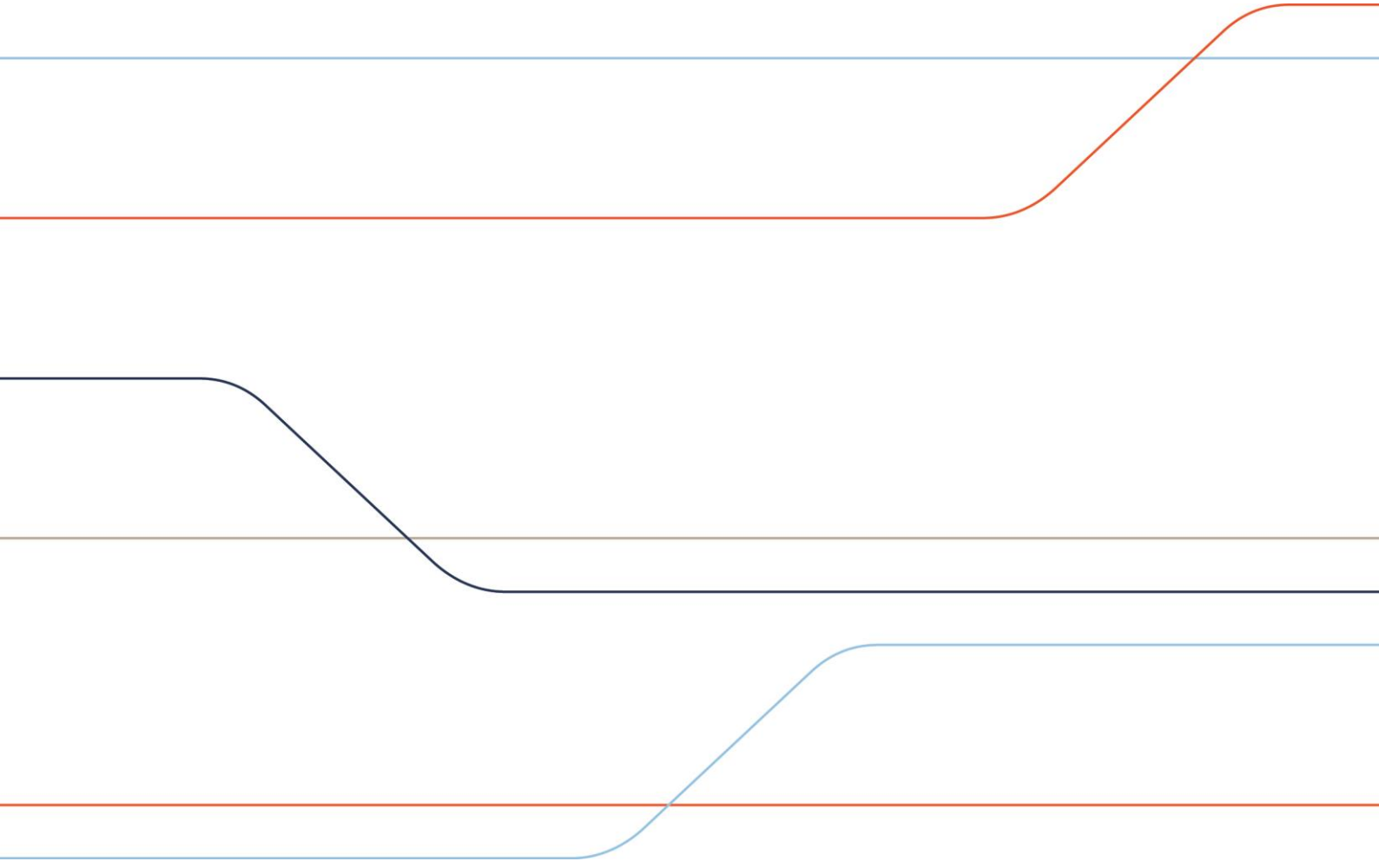


Table of contents

1 Introduction

1.1 Summary

With the Saferpay Authorization Interface, in the following also called AI, online transactions can be processed in the background. The AI is convenient for shop-systems, call center solutions, merchandise planning-, ERP- and CRM-systems. This document describes the integration of the AI in existing systems with the Saferpay Client Library, in the following also called LIB and the Saferpay https Interface in the following also called HI.

1.2 Requirements

The use of the AI with the LIB requires the fulfillment of the following conditions

- A corresponding license and with that the existence of a valid identifier with username and password for the Saferpay system.
- At least one active Saferpay terminal, allowing the payment processing, and the associated TERMINALID, respectively the concerned Saferpay ACCOUNTID, is available.
- The existence of a valid acceptance contract for credit cards and other payment means.

  In order to be able to use the HI, a HI configuration with the merchant data must be setup on Saferpay side. The keys for the signature of the (SSL secured) communication with Saferpay are provided by this configuration. The setup is free of charge but must be individually requested for each Saferpay business account. Please send a formless e-mail requesting the setup to onlinepayment@six-group.com if you have concluded your contract in Switzerland or to service.saferpay@six-payment-services.comservice.saferpay@six-payment-services.com if you have a contract for another country (D, NL, A, etc.).

Important! Please take care to specify your Saferpay customer number (ACCOUNTID) and the IP-address(es) of your server(s) making the requests to Saferpay.

1.3 Data Security and PCI DSS

The credit card organizations have initiated a security program called PCI DSS (Payment Card Industry Data Security Standard) to prevent fraud and abuse of credit cards.

Please take care to respect the PCI DSS guidelines in the design of your payment processes and the usage of the Saferpay Authorization Interface. In combination with the optional Saferpay Secure Card Data service the payment process can be designed so safely that no credit card number is processed, stored or transferred via your (web) servers. The risk of abuse of the credit card data is thereby reduced and it precludes the necessity of an expensive PCI DSS review of the merchant system.

If you have any questions regarding PCI DSS, please contact your acquirer or a qualified security provider (see https://www.pcisecuritystandards.org/pdfs/pci_pa-dss_list.pdf).

1.4 Supported Payment Means

The Saferpay Batch Processing actually allows the processing of transactions for the following payment means:

- Visa
- MasterCard
- Maestro international
- V PAY
- American Express
- Diners Club
- J.C.B.
- Union Card
- ELV electronic direct debits (Germany only)

All other payment means, requiring a data input on the site of the payment provider can be processed via the Saferpay Payment Page. For further questions please contact integration.saferpay@six-payment-services.com

1.5 Format Information

The following abbreviations for format information are used in this document:

a	letters (a - z, A - Z)
n	numeric characters (0 - 9)
an	alphanumeric characters (a - z, A - Z, 0 - 9)
s	special characters (:?,-(+‘.)/ and space)
ans	alphanumeric and special characters

2 Saferpay Client Library

The Saferpay LIB is to be installed on the server that hosts the application of the merchant.
After the installation Saferpay classes and methods are available on the server.

Root- respectively administrator rights on the destination server are required to install the LIB and to generate a new configuration (generation of keys).

The LIB is available as .NET- or Java-version. The corresponding installation files can be downloaded in the download area of the Saferpay Backoffice via the following address:

https://www.Saferpay.com/download/

If neither the .NET LIB nor the Java LIB can be used or if a local installation is not possible the Saferpay https Interface can be used as an alternative.

2.1 Requirements

2.1.1 .NET Client Library

The Saferpay .NET Client LIB is compiled with .Net Framework 2.0. So it is mandatory to have installed this version on the target server, too.

2.1.2 Java Client Library

On the target server a Sun Java Runtime Environment (JRE) version 1.3.1 or newer has to be installed. Other Java environments from IBM or OpenJDK are not compatible with Saferpay Java LIB.

2.2 Installation

2.2.1 .NET Client Library

Please execute the downloaded file and follow the instructions of the setup assistant.

2.2.2 Java Client Library

For the integration in java please unpack the downloaded file and copy the included “Saferpay.jar” into the directory jre/lib/ext.

For the integration in other programming- or script languages the “Saferpay.jar” can be copied in any directory.

2.3 Proxy Server Configuration

In case, outgoing connections in your network use a proxy server, the appropriate configuration data of the proxy server is needed to establish the Saferpay LIBs.

2.3.1 .NET Client Library

To use a proxy server a few parameters have to be added to „config.xml“. The file is located in the installation directory of the .NET Client, for instance in `C:\Programm files\Saferpay\Client\`.

Proxy server with user authentication

For communication through a proxy that requires user authentication the following parameters have to be added to „config.xml“. In doing so the order of the parameters does not matter.

```
PROXYPASSWORD="secret"
PROXYUSERNAME="MyProxyUser"
PROXYADDRESS="http://localhost:8080"
USEPROXY="True"
USEDEFAULTCREDENTIALS="False"
```

Proxy server without user authentication

For communication through a proxy without user authentication the following parameters have to be added to „config.xml“.:

```
PROXYADDRESS="http://localhost:8080"
USEPROXY="True"
USEDEFAULTCREDENTIALS="True"
```

Depending on the proxy configuration the content of „config.xml“ could look like:

```
<IDP MSGTYPE="SetupResponse" GXID="6216B171-B449-4D02-A114-D42AB58D42AE"
CUSTOMERID="99867" VERSION="47"
VTAUTOURL="https://www.saferpay.com/user/setup.asp"
VTURL="https://www.saferpay.com/vt2/Pay.aspx" VTKEYID="1-0"
CAPTUREURL="https://www.saferpay.com/scai2/index.aspx"
VTSCRIPTURL="http://www.saferpay.com/OpenSaferpayScript.asp"
USEDEFAULTCREDENTIALS="True" USEPROXY="True"
PROXYADDRESS="http://localhost:8080" />
```

2.3.2 Java Client Library

For the Java LIB the configuration of a proxy server can be done either by a file „settings.xml“ or by command line call. For the usage of „settings.xml“ the file must be created within the same directory where “saferpay.jar“ is located, for instance in `jre/lib/ext`.

Proxy server with user authentication

Sample „settings.xml“:

```
<IDP PROXYHOST="10.23.209.100" PROXYPORT="8080" PROXYUSERNAME="myUserId"
PROXYPASSWORD="myPassword" TRACEOPT="rawhttp" VERSION="1" USEPROXY="1" />
```

Sample command line call:

```
--proxyHost 10.23.209.100 --proxyPort 8080 --proxyUser myUserId
--proxyPassword myPassword
```

Proxy server without user authentication

Sample „settings.xml“:

```
<IDP PROXYHOST="10.23.209.100" PROXYPORT="8080" TRACEOPT="rawhttp"
VERSION="1" USEPROXY="1" />
```

Sample command line call:

```
--proxyHost 10.23.209.100 --proxyPort 8080
```

2.4 Key Generation

Besides the SSL encrypted communication between the LIB and the Saferpay servers the data of a Saferpay account is also protected by a digital signature according to the PGP (Pretty Good Privacy) policy. For this purpose a key pair must be generated and stored on the merchant server for every Saferpay account. The generation of the keys requires a valid login and password for the Saferpay Backoffice. After successfully generating the keys the password can be changed via the Saferpay Backoffice since the generation of the key is only necessary once and does not need to be repeated. Generated key pairs remain valid and should therefore be kept safe with restricted access.

2.4.1 .NET Client Library

After installation of the .NET LIB a GUI (Graphical User Interface) is available for the key generation. The GUI can be found at:

The Saferpay Client Setup opens. Please follow the subsequent instructions.

2.4.2 Java Client Library

The key generation with the java LIB is done via the command line. Therefore please change to the directory with the Saferpay.jar and enter the following command:

```
java -jar Saferpay.jar -conf -p . -r
https://www.Saferpay.com/user/setup.asp -u <YOUR-ACCOUNT> -w <YOUR-
PASSWORD>
```

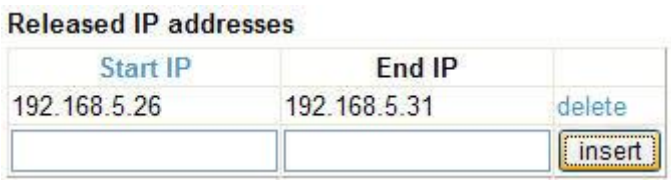
The example uses the access data of the Saferpay test account.

The command line help can be called with:

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

2.5 IP Access Configuration

Even if the communication with Saferpay is already encrypted it is recommended to restrict the access to the Saferpay account via the Client LIB as additional security measure. For this purpose the IP-Access can be configured via the menu point “IP Permissions” in the Saferpay Backoffice (https://www.Saferpay.com/user/login.asp).



After adding one or more IP addresses the access to the Saferpay account is restricted to these. Requests from other IP addresses are then blocked by Saferpay.

3 Classes and Methods of the Client Library

This chapter describes the classes and methods available for the integration of the Saferpay Client LIB.

3.1 Summary

3.1.1 Payment request and authorization response

The payment request is generated with the CreateRequest() method. The generated MessageObject is filled with the transaction parameters and triggered with Execute(). Afterwards the parameters of the authorization response can be evaluated.

1) Creation of a MessageFactory Object.
2) Opening of the corresponding Configuration with Open().
3) Call of CreateRequest(), in order to get an empty MessageObject.
4) Call of SetAttribute() with the MessageObject to set the parameters.
5) Call of Execute() to engage the payment request.
6) Call of GetAttribute() with the MessageObject to read the response parameters.

3.1.2 Settlement, Cancel and Batch Close

1) Creation of a MessageFactory Object.
2) Opening of the corresponding Configuration with Open().
3) Call of CreateRequest("PayComplete"), in order to get a MessageObject.
4) Call of SetAttribute() with the MessageObject to set the parameters.
5) Call of Capture() with the MessageObject.

3.2 MessageFactory Class

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

3.3 MessageObject Class

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

3.4 Open() Method

The reference to the key pair of the merchant account is done by the call of Open(). In order to ensure that the other function calls of this MessageFactory also refer to these keys, Open() must be called before all other methods of the MessageFactory object.

3.5 Execute() Method

The call of Execute() transmits the message of the specified message type (msgtype).

3.6 CreateRequest() Method

Creates a new Request MessageObject of the specified message type (msgtype). For instance with CreateRequest("PayComplete") transactions with status Reservation can be settled or canceled. A reservation can also be settled with a partial amount, a transaction with status payment can be canceled and the Batch Close can be engaged.
CreateRequest("PayComplete") always needs ID and TOKEN for a settlement. For a settlement with reduced amount the additional parameter AMOUNT must be transmitted. For the cancel of a Reservation or a Payment as well as for the start of the Batch Close the additional parameter ACTION is required. Every call must contain the ACCOUNTID.

3.7 SetAttribute() Method

With SetAttribute() the necessary parameters for the message are set. Please take care to respect the case sensitivity of the used parameter names.

3.8 GetAttribute() Method

GetAttribute() returns the value of a parameter of the message. If the parameter is not included in the message the call fails. Please take care to respect the case sensitivity of the used parameter names.

3.9 Capture() Method

The call of Capture() transmits the message of the message type CreatePayComplete.

4 Saferpay https Interface

The Saferpay https Interface can be used as an alternative to the Saferpay Client Library.
This might be the case if for example the LIB cannot be installed or used on the destination system.

4.1 IP Access and Password of the Saferpay https Interface

Saferpay ensures that the data exchanged with the Merchant application cannot be manipulated. Possible manipulations by experienced internet users are recognized and reported to the merchant application.

The access to the Saferpay https Interface is only possible if

- the calling IP Address(es) has been explicitly activated on the Saferpay Server.
- the https Interface password is transmitted within the authorization and settlement requests.

4.2 https Interface Addresses

The Saferpay https Interface can be accessed via the following web addresses:

Authorization and Refund
https://www.saferpay.com/hosting/Execute.asp

Settlement
https://www.saferpay.com/hosting/PayCompleteV2.asp

Attention! Most frameworks verify the server certificate automatically. Nevertheless when using the Saferpay https interface, we recommend to make sure that your application verifies the www.saferpay.com server certificate to prevent man-in-the-middle attacks..

4.3 Transmission of Messages

The HI submits an answer to every request, except in case of technical problems that do not allow the return of the response message.

Request
The request data (parameters) to the HI can be transmitted via POST or GET.

Request example:
https://www.Saferpay.com/hosting/Execute.asp?spPassword=hfJK43SA&AMOUNT=1295
&CURRENCY=EUR*...further parameters*

Response
The HI answers are text messages. A successful executed request is answered with “OK” followed by a colon and the response data. The standard format of the response is XML. The signalization with “OK” indicates the successful processing of the request - the merchant application must then evaluate the response data, like checking the value of the parameter RESULT.

Example of a response message:
OK:<IDP RESULT="65" ...further parameters... />

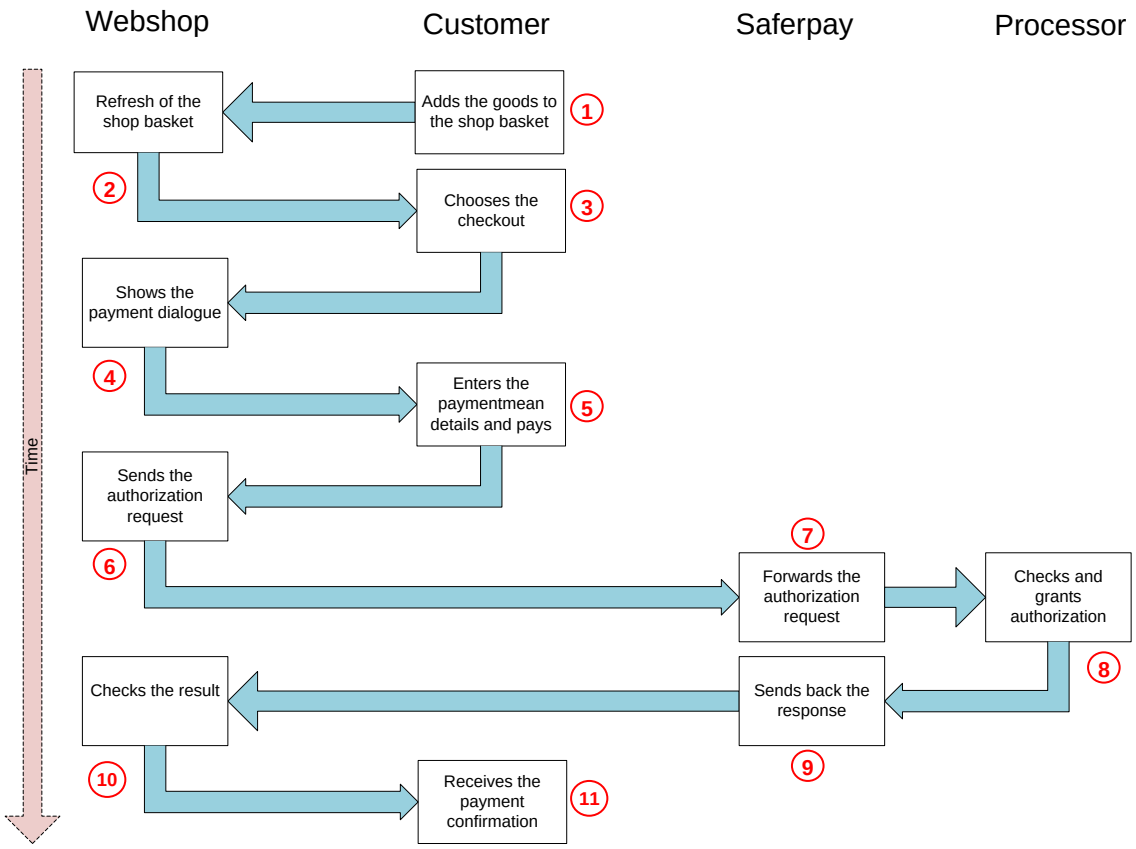
In case of an (application) error the HI will answer with “ERROR” followed by a colon and an optional error description.
Example:
ERROR: Hosting: Merchant not configured or unknown

Notes
The request processing is always sequential. In case of multiple parallel requests to the HI the requests are at latest serialized at the Saferpay calculation centre based on the Saferpay Accountid. To process multiple transactions in a parallel way a pool of Saferpay Accountid’s, must be setup and used alternately (round robin).

5 Processing Steps

5.1 Overview

The following chart shows the process flow of a successful online payment via the Saferpay Authorization Interface:



5.2 Process Description

1. The customer puts the goods in the basket of the shop.
2. In the webshop the basket is refreshed and the price to pay is displayed.
3. The customer goes to the checkout.
4. The payment dialogue of the shop is displayed.
5. The customer enters the needed data for the chosen payment mean and clicks on “Pay”.
6. The webshop transmits the payment request to Saferpay.
7. Saferpay forwards the payment request directly to the processor.
8. The processor checks the payment request, grants the authorization and returns the authorization response to Saferpay.
9. Saferpay forwards the response to the shop.
10. The shop gets the authorization response data and checks the result.

11 The customer gets the payment confirmation from the shop.

The following points are not shown on the chart since their execution via the shop system is not mandatory:

12 The amount is settled (PayComplete) and the order can be processed. The settlement can be done directly after receiving the authorization response by the webshop or at a later time for example from within an ERP system.

> *The settlement of a Reservation is mandatory for the Batch Close. The Batch Close only processes transactions with the status "Payment" and forwards them to the Processor in order to engage the financial transfer. The money is than subsequently credited as compound item to the merchants business account. The merchant gets a settlement list from the payment mean processor.*
>
> *Depending on the business case the settlement can also be done at a later time, normally within 6 days since that is the normal lifetime of a reservation. Since this value might vary depending on processor and payment mean please ask your processor for further information.*
>
> *The Batch Close can be initiated manually or automatically.*

13 Via the PayComplete call the status of a transaction changes from "Reservation" to "Payment". The transaction status is shown in the Saferpay Backoffice Journal.

6 Parameter

6.1 Authorization Request

The following table lists the available parameters for the message type “Authorization”
If not specified as *Optional* the parameters are mandatory.

Parameter	Format	Description
spPassword	ans[..40]	*https Interface parameter* Password required for communication with the HI. The password is provided separately.
ACCOUNTID	ns[..15]	The Saferpay account number of the merchant for this transaction. e.g. “401860-17795278” for the Saferpay Test Account.
PAN	n[..19]	The credit card number (Primary Account Number). PAN needs the additional parameters EXP and CVC (for initial payments).
EXP	n[4]	Expiry date as printed on the card. The format is MMYY, e.g. “1215” for 12/2015.
CVC	n[..4]	3- or 4-digit credit card verification value also known as CID/4DBC (American Express) CVC2 (MasterCard) CVV2 (Visa) CAV (JCB)
	⚠	The use of the parameter is mandatory for initial payments with a credit card. For subsequent payments the CVC is not available since electronic storage is strictly forbidden.
IBAN	an[22]	International Bank Account Number, SEPA bank data Only German IBAN are supported. Replaces the parameter PAN, EXP and CVC for payments with German direct debits. Do not use together with TRACK2 parameter. Format: "DE[checksum, 2 digits][bank code, 8 digits] [bank account number, 10 digits]"
CARDREFID	ans[..40]	*Optional* Reference number for credit card number and expiry date or bank account information (German direct debits). The use implies the service “Saferpay Secure Card Data”.
AMOUNT	n[..8]	Authorization amount in minor currency unit, e.g. 1230 in EUR means EUR 12,30.
CURRENCY	a[3]	ISO 4217 three letter currency code e.g. CHF or EUR

Parameter	Format	Description
ORDERID	ans[..80]	*Optional, (mandatory for the giropay payment method)* ORDERID contains the reference number for a payment. The ORDERID reference must be unique to ensure clear allocation. Saferpay can process 80 characters for the ORDERID, however this is normally not possible on the processor side. Excessively long character chains are generally truncated to 80 characters. In practice, a length of 12 characters has been proven to be a good value. In case of doubt please ask your processor how many characters can be processed.
MANDATEID	an[..35]	*Optional* Mandate reference of the payment. Needed for German direct debits (ELV) only. The value has to be unique. As a default Saferpay transaction identifier is used.
NAME	ans[..50]	*Optional* Contains the cardholder's name. Special characters in the name must be transmitted HTML encoded as Entity.
MPI_SESSIONID	an[28]	*Optional* The session of the VerifyEnrollment process is needed for the authorization request to flag the transaction as 3-D Secure (only needed for "Verified by Visa" and "MasterCard SecureCode").
PRE-AUTH	a) 3	*Optional* Flags a pre-authorized transaction. Pre-authorized transactions can be settled up to 30 days after authorization. If no parameter is specified, a "final authorization" (default) is processed. Values: "yes" or "no" (default) **Attention!** Pre-authorizations are not supported by all processors. Pre-authorizations are currently possible via Saferpay in the case of the processors SIX, B+S CardService, ConCardis, Airplus, and, after consultation, with American Express.
IP	ns[..15]	*Optional* IP address of the customer in order to determine the country of origin by geo targeting.
AUTHCODE	an[..64]	*Optional* Authorization code of the processor if e.g. the request has been preauthorized by telephone..
AUTHFLAGS	n[..2]	*Optional* Following values are allowed: 0 = Standard value, the payment is done with authorization. 4 = Authorization already done (AUTHCODE). 16 = At own risk. Payment is processed without authorization.

Parameter	Format	Description
ACTION	a[..6]	*Optional* Flags the payment request as payment or refund Values: "Debit" (Standard, card holder is charged) and "Credit" (Card holder is credited).
RECURRING	a[..3]	*Optional* Flags the payment request as recurring payment. Must be set for initial recurring payment and all following recurring payments. Values: "yes" or "no" (default) Not to use in combination with INSTALLMENT!
RECFREQ	n[..3]	*Optional* Comes with initial recurring payment and declares the minimum of days between recurring payments, for instance "28" conforms to one month. Must be used together with RECEXP!
RECEXP	n[8]	*Optional* For the initial payment declares the date from when on no more recurring payments will follow. The format is YYYYMMJJ, for instance "20151231" 31.12.2015. For 3-D Secure requests the ACS checks whether the card has a valid expiry date. Must be used together with RECFREQ!
INSTALLMENT	a[..3]	*Optional* Flags the payment request as installment payment. Must be set for initial installment payment and all following installment payments. Values: "yes" or "no" (default) Not to use in combination with RECURRING!
INSTCOUNT	n[..2]	*Optional* Number of installments as agreed between merchant and customer. The minimum value is "2". INSTCOUNT is mandatory for the initial installment payment and not necessary for following installment payments!
REFID	an[28]	*Optional** **Payment**: Uses the transaction identifier of the initial payment to refer to following recurring or installment payments. Value: ID of the initial payment **Refund**: Uses the transaction identifier to refer to the origin payment for a refund. Reservations have to be captured with PayComplete first or the refund will be declined! Value: ID of the payment

Parameter	Format	Description
REFOID	ans[..80]	*Optional** **Payment**: Uses the reference number of the initial payment to refer to following recurring or installment payments. Value: ORDERID of the initial payment **Refund**: Uses the reference number to refer to the origin payment for a refund. Reservations have to be captured with PayComplete first or the refund will be declined! Value: ORDERID of the payment

** Referring to the origin payment for refunds is optional for most acquirers. Known exceptions are Yapi Kredi, Alfa Bank und Cielo. In case of doubt, the usage of the parameters REFID or REFOID for refunds is recommended.*

With some processors, referenced refunds may be declined if **the batch close was not processed** previously.
Please therefore ensure that the batch close was processed in the case of the referenced transaction; otherwise please discard the transaction with **PayComplete** (chapter 6.3) and **ACTION="cancel"!**

If you wish to use the automatic batch close, this is processed daily at approx. 10.00 pm. Comparing the transactions by time stamp is recommended.

Alternatively, you can initiate the batch close yourself via **PayComplete** (chapter 6.3) with **ACTION="CloseBatch"**. However, please ensure here that the automatic batch close is deactivated in Saferpay Backoffice and the transactions flagged accordingly in your system.

6.2 Authorization Response

The following table lists the parameters which can be returned within the authorization response.

Parameter	Format	Description
MSGTYPE	a[..30]	Always contains the value „AuthorizationResponse“.
RESULT	n[..4]	Contains the result of the authorization request 0 = Request successfully processed ≠0 = Request NOT successfully processed A list of possible RESULT values is to be found in the Chapter RESULT values.
ACCOUNTID	ns[..15]	The Saferpay account number of the merchant for this transaction. e.g. 99867-94913159 for the Saferpay Test Account.
ID	an[28]	Unique Saferpay transaction identifier.
TOKEN	ans[..40]	May contain additional information concerning the transaction processing. Standard value: "(unused)"
PROVIDERID	n[..4]	Contains the Provider ID of the payment means processor.
PROVIDERNAME	ans[..40]	Contains the name of the payment means processor.
AUTHRESULT	n[..3]	Contains the response code of the processor. If no connection to the processor was established the value of RESULT will indicate this. The values vary depending on the used payment mean protocol.
AUTHCODE	n[6]	Contains the authorization code of the credit card processor in case of successful authorization.
PAYMENT_PROTOCOL	ans[..30]	The name of the payment means protocol used for the connection.
CAVV	ans[28]	*3-D Secure Parameter** Cardholder Authentication Verification Value For a MasterCard the UCAF value is contained. Saferpay, independently from the credit card type, uses the value CAVV
MPI_LIABILITYSHIFT	a[..3]	*3-D Secure parameter** Indicates whether technically formal liability shift is granted. Values: "yes" or "no" **Attention!** Not all processors can check the liability shift during the authorization and can already exclude it within the authorization response. Therefore it is possible that even if MPI_LIABILITYSHIFT and ECI indicate an existing liability shift the processor might refuse it for contractual reasons. In case of questions concerning this please contact your processor for further information.
XID	ans[28]	*3-D Secure Parameter** Extra identifier This base64 string is generated by the MPI and references to the instance within the 3-D Secure protocol.

Parameter	Format	Description
ECI	n[1]	*3-D Secure Parameter** Electronic Commerce Indicator Is needed for the flagging of 3-D Secure transactions (“Verified by Visa“, “MasterCard SecureCode“): 0 = SSL secure internet payment, no liability shift 1 = SSL secure internet payment with 3DS and liability shift, customer is taking part in the process 2 = SSL secure internet payment with 3DS and liability shift, customer is not taking part in the process.
AUTHDATE	ns[17]	Contains the timestamp of the authorization. format: YYYYMMDD hh:mm:ss
EXP	n[4]	Contains the expiry date of the requested card. format: MMYY
PAN	ans[..23]	Contains the masked credit card number of the request. e.g. "xxxx xxxx xxxx 0111".
IBAN	an[22]	Contains the IBAN of the request. e.g. "DE77970000010123456789".
CARDREFID	ans[..40]	Contains the reference value of the payment mean the authorization request was done with.
REFERRAL	ans[..30]	Depending on the processor, contains a phone number or textual information for a pre-authorization by telephone.
ACQUIRER_TERMINALID	n[..10]	Contains the terminal identifier of the direct debit processor (Germany).
BANK_CODE_NUMBER	n[8]	Contains the BIC of the requested bank connection.
PROTOCOL_AID	n[8]	In case of successful authorization contains the authorization code of the direct debit processor (Germany).
PROTOCOL_STAN	n[..9]	Contains the sequence number of the direct debit terminal (Germany).
MANDATEID	ans[..35]	Contains the mandate reference of an ELV payment.
CREDITORID	ans[..35]	Contains the creditor identifier of an ELV payment.
AUTHMESSAGE	ans[..30]	Contains a textual response to the authorization.
IP	ns[..15]	*Fraud Prevention Parameter* ** Contains the committed IP address of the customer.
IPCOUNTRY	a[2]	*Fraud Prevention Parameter* ** 2-letter ISO 3166 country code, e.g. CH, DE, AT… Contains the IP geo location country of the customer’s IP address. If the country cannot be retrieved, IPCOUNTRY will be empty or contain “IX”.
CCCOUNTRY	a[2]	*Fraud Prevention Parameter* ** 2-letter ISO 3166 country code, e.g. CH, DE, AT…of the country of the card issuing bank. If the country cannot not be retrieved CCCOUNTRY will not be contained in the response.

**Required condition is the participation in the 3D-Secure process. (“Verified by Visa“, “MasterCard SecureCode“, “American Express SafeKey“))*
*** Only available if Safetpay Risk Management is activated.*

6.3 PayComplete Parameters

The following parameters are available for a CreatePayComplete message:

Parameter	Format	Description
spPassword	ans[..40]	*https Interface parameter* Password requested for communication with the HI. The password is provided separately.
ID	an[28]	Saferpay transaction identifier from the PayConfirm message.
ORDERID	ans[..80]	*Optional* Can be used instead of ID, but has to be unique then.
TOKEN	ans[..40]	The token from the PayConfirm message.
AMOUNT	n [..8]	Amount to settle in minor currency unit e.g. 1230 in EUR means EUR 12,30.
ACCOUNTID	ns[..15]	Saferpay's merchant account identifier for this transaction. e.g. 99867-94913159 for the Saferpay Test Account.
ACTION		*Optional* Is used for special processing options. Possible values are: "Settlement", "CloseBatch", "Cancel" Settlement Instructs the Saferpay system to change the status of the transaction from "Reservation" to "Payment". With that the transaction will be forwarded by the next Batch Close to the concerned processor in order to engage the actual fund transfer. With the parameter AMOUNT an amount inferior than reserved can be settled. The difference between reserved and settled amount is canceled. The reserved amount is the maximum amount that can be settled. CloseBatch Instructs the Saferpay system to engage the Batch Close for the specified ACCOUNTID. If no ACCOUNTID is specified the call fails. Cancel With that call a reservation can be discarded or a payment canceled as long as it has not been processed by the Batch Close. If a reservation is discarded it will remain visible for 6 days under „Discarded Reservations". After that it will be erased from the database, whereas canceled payments will remain visible in the back office flagged as "Canceled Payment". If the ACTION parameter is not submitted the default value ACTION="Settlement" is used.

6.4 PayComplete Response

The following table lists the parameters of the PayComplete response:

Parameter	Format	Description
MSGTYPE	a[..30]	Always contains the value “PayConfirm“.
ID	an[28]	Saferpay transaction identifier
RESULT	n[..4]	Contains the result of the request processing 0 = request successfully processed. ≠0 = request not successfully processed.
MESSAGE	ans[..30]	Contains a textual response to the request
AUTHMESSAGE	ans[..30]	Can contain a textual response to the request

7 Saferpay Test Environment

For the integration phase and in order to be able to test Saferpay, we can offer you our External Test Environment (ETU).
In this environment, which is isolated from the operational environment, you can test Saferpay with simulations for all current payment means in your own test account.

All the details on our test environment can be found at the following address:

https://www.six-payment-services.com/de/site/saferpay-support/testaccount

8 Examples

8.1 Important Notice

Please note that own values should always be HTML encoded, either as HTML entity or as Unicode in order to ensure that all special characters are transmitted to Saferpay correctly.

8.2 C# with the .NET LIB

Authorization Request Payment

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

request.SetAttribute("ACCOUNTID", "401860-17795278");
request.SetAttribute("AMOUNT", "12500");
request.SetAttribute("CURRENCY", "EUR");
request.SetAttribute("PAN", "9451123100000111");
request.SetAttribute("EXP", "1214");
request.SetAttribute("CVC", "123");
request.SetAttribute("NAME, Server.HtmlEncode("Stefanie Müller"));
request.SetAttribute("ORDERID", "123456789");

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

Authorization Request Refund

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

request.SetAttribute("ACCOUNTID", "401860-17795278");
request.SetAttribute("AMOUNT", "12500");
request.SetAttribute("CURRENCY", "EUR");
request.SetAttribute("PAN", "9451123100000111");
request.SetAttribute("EXP", "1214");
request.SetAttribute("CVC", "123");
request.SetAttribute("ACTION", "Credit");
request.SetAttribute("ORDERID", "123456789");

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

Authorization Response

```
int result = Convert.ToInt32(response.GetAttribute("RESULT"));
if (result == 0)
{
        String id = response.GetAttribute("ID");
        Console.WriteLine("Authorisation successful!");
}
else
{
        Console.WriteLine("Authorization failed! RESULT=" + result);
        return;
}
```

Settlement of a Payment with CreatePayComplete

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

mo_paycomplete.SetAttribute("ID", id);
mo_paycomplete.SetAttribute("ACCOUNTID", "401860-17795278");

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

Settlement of a Payment with reduced Amount with CreatePayComplete

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

mo_paycomplete.SetAttribute("ID", id);
mo_paycomplete.SetAttribute("ACCOUNTID", "401860-17795278");
mo_paycomplete.SetAttribute("AMOUNT", "10000");

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

Cancel of a Payment with CreatePayComplete

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

mo_paycomplete.SetAttribute("ID", id);
mo_paycomplete.SetAttribute("ACCOUNTID", "401860-17795278");
mo_paycomplete.SetAttribute("ACTION", "Cancel");

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

Initiating the Batch Close with CreatePayComplete

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

mo_paycomplete.SetAttribute("ACCOUNTID", "401860-17795278");
mo_paycomplete.SetAttribute("ACTION", "CloseBatch");

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

PayComplete Response

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

8.3 Java with the Java LIB

Authorization Request Payment

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

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

request.SetAttribute("ACCOUNTID", "401860-17795278");
request.SetAttribute("AMOUNT", "12500");
request.SetAttribute("CURRENCY", "EUR");
request.SetAttribute("PAN", "9451123100000111");
request.SetAttribute("EXP", "1214");
request.SetAttribute("CVC", "123");
request.SetAttribute("NAME", StringEscapeUtils.escapeHtml("Stefanie Müller"));
request.SetAttribute("ORDERID", "123456789");

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

Authorization Request Refund

```
import Saferpay.*;

MessageFactory mf = new MessageFactory();
mf.Open("");
MessageObject request = mf.CreateRequest("Authorization");
request.SetAttribute("ACCOUNTID", "401860-17795278");
request.SetAttribute("AMOUNT", "12500");
request.SetAttribute("CURRENCY", "EUR");
request.SetAttribute("PAN", "9451123100000111");
request.SetAttribute("EXP", "1214");
request.SetAttribute("CVC", "123");
request.SetAttribute("ACTION", "Credit");
request.SetAttribute("ORDERID", "123456789");

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

Authorization Response

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

Settlement of a Payment with CreatePayComplete

```
import Saferpay.*;

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

mo_paycomplete.SetAttribute("ID", id);
mo_paycomplete.SetAttribute("ACCOUNTID", "401860-17795278");

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

Amount reduced settlement of a payment with CreatePayComplete

```
import Saferpay.*;

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

mo_paycomplete.SetAttribute("ID", id);
mo_paycomplete.SetAttribute("ACCOUNTID", "401860-17795278");
mo_paycomplete.SetAttribute("AMOUNT", "10000");

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

Cancelation of a Payment with CreatePayComplete

```
import Saferpay.*;

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

mo_paycomplete.SetAttribute("ID", id);
mo_paycomplete.SetAttribute("ACCOUNTID", "401860-17795278");
mo_paycomplete.SetAttribute("ACTION", Cancel);

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

Initiating the Batch Close with CreatePayComplete

```
import Saferpay.*;

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

mo_paycomplete.SetAttribute("ACCOUNTID", "401860-17795278");
mo_paycomplete.SetAttribute("ACTION", "CloseBatch");

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

PayComplete Response

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

8.4 Command line calls with the Java LIB

Authorization request payment

```
java -jar Saferpay.jar -exec -p c:/programs/soplex/Saferpay/keys/99867 -m Authorization -a
ACCOUNTID 401860-17795278 -a PAN 9451123100000004 -a EXP 1214 -a CVC 123 -a AMOUNT 12500 -a
CURRENCY EUR -a ORDERID 123456789 -a NAME "Stefanie M&uuml;ller" -of cai.txt
```

Authorization request refund

```
java -jar Saferpay.jar -exec -p "c:/programs/soplex/Saferpay/keys/99867 -m Authorization -a
ACCOUNTID 401860-17795278 -a PAN 9451123100000004 -a EXP 1214 -a CVC 123 -a AMOUNT 12500 -a
CURRENCY EUR -a ORDERID 123456789 -a NAME "Stefanie M&uuml;ller" -a ACTION Credit -of
cai_refund.txt
```

Authorization Response

Payment (contents of cai.txt):

```
<IDP PAN="xxxx xxxx xxxx 0004" CCCOUNTRY="XX" EXP="1214"
REFERRAL="017772357" AUTHRESULT="1" AUTHCODE="745000"
ACCOUNTID="401860-17795278" RESULT="0" AUTHDATE="20110418 14:12:31"
PAYMENT_PROTOCOL="CARCDS" PROVIDERNAME="Saferpay Test Card"
PROVIDERID="90" ID="Ctp7OpbnQ8phSA13Ev9Wb512S0bA"
MSGTYPE="AuthorizationResponse" AUTHMESSAGE="request was processed successfully" ECI="0"
CONTRACTNUMBER="123456789" TOKEN="(unused)"/>
```

Refund (contents of cai_refund.txt):

```
<IDP PAN="xxxx xxxx xxxx 0004" CCCOUNTRY="XX" EXP="1214"
REFERRAL="017772357" AUTHRESULT="1" AUTHCODE="      "
ACCOUNTID="401860-17795278" RESULT="0" AUTHDATE="20110418 14:24:14"
PAYMENT_PROTOCOL="CARCDS" PROVIDERNAME="Saferpay Test Card"
PROVIDERID="90" ID="xI4lvCAClSz2vAKY9YGOA7dlI9Ub"
MSGTYPE="AuthorizationResponse" AUTHMESSAGE="request was processed successfully" ECI="0"
CONTRACTNUMBER="123456789" TOKEN="(unused)"/>
```

Settlement of a payment with CreatePayComplete

```
java -jar Saferpay.jar -capt -p c:/programs/soplex/Saferpay/keys/99867 -a ACCOUNTID 401860-
17795278 -if cai.txt -of capt.txt
```

Alternatively the parameters can be transmitted directly:

```
java -jar Saferpay.jar -capt -p c:/programs/soplex/Saferpay/keys/99867 -i
Ctp7OpbnQ8phSA13Ev9Wb512S0bA -a ACCOUNTID 401860-17795278 -of capt.txt
```

Settlement of a payment with reduced amount with CreatePayComplete

```
java -jar Saferpay.jar -capt -p c:/programs/soplex/Saferpay/keys/99867 -a ACCOUNTID 401860-
17795278 -if cai.txt -a AMOUNT 10000 -of capt.txt
```

Alternatively the parameters can be transmitted directly:

```
java -jar Saferpay.jar -capt -p c:/programs/soplex/Saferpay/keys/99867 -i
8xMY2hbb33dGtA8x96Ylb27jjOfb -a ACCOUNTID 401860-17795278 -a AMOUNT 10000 -of capt.txt
```

Cancelation of a payment with CreatePayComplete

```
java -jar Saferpay.jar -capt -p c:/programme/soplex/Saferpay/keys/99867 -a ACCOUNTID 401860-
17795278 -if cai.txt -a ACTION Cancel -of cancel.txt
```

Alternativ können die Parameter direkt übergeben werden:

```
java -jar Saferpay.jar -capt -p c:/programme/soplex/Saferpay/keys/99867 -i
Ctp7OpbnQ8phSA13Ev9Wb512S0bA -a ACCOUNTID 401860-17795278 -a ACTION Cancel -of cancel.txt
```

Initiating the Batch Close with CreatePayComplete

```
java -jar Saferpay.jar -capt -p c:/programme/soplex/Saferpay/keys/99867 -a ACTION CloseBatch
-a ACCOUNTID 401860-17795278 -of daily.txt
```

8.5 https Interface

Authorization Request Payment

```
https://test.saferpay.com/hosting/execute.asp?spPassword=8e7Yn5yk&ACCOUNTID=401860-
17795278&ORDERID=123456789-001&AMOUNT=1000&CURRENCY=EUR&PAN=9451123100000004&EXP=1214&CVC=123
```

Authorization Request Refund

```
https://test.saferpay.com/hosting/execute.asp?spPassword=8e7Yn5yk&ACTION=Credit&ACCOUNTID=4018
60-17795278&ORDERID=123456789-001&AMOUNT=1000&CURRENCY=EUR&PAN=9451123100000004&EXP=1214
```

Authorization Response

```
OK:<IDP RESULT="0" MSGTYPE="AuthorizationResponse" ID="EvrKOEApM3YtSApnE0MlAU28nCYb"
TOKEN="(unused)" AUTHRESULT="1" AUTHMESSAGE="request was processed successfully"
AUTHCODE="500000" PROVIDERID="90" PROVIDERNAME="Saferpay Test Card" ECI="0" CCCOUNTRY="XX"
CONTRACTNUMBER="123456789" ORDERID="123456789-001" AUTHDATE="20110621 10:07:26" EXP="1214"
PAN="xxxx xxxx xxxx 0004"/>
```

Settlement of a Payment with CreatePayComplete

```
https://test.saferpay.com/hosting/paycompletev2.asp?spPassword=8e7Yn5yk&ACCOUNTID=401860-
17795278&ID=EvrKOEApM3YtSApnE0MlAU28nCYb
```

Settlement of a Payment with reduced Amount with CreatePayComplete

```
https://test.saferpay.com/hosting/paycompletev2.asp?spPassword=8e7Yn5yk&ACCOUNTID=401860-
17795278&ID=8IK98fbtS4d7UAzMGGCEAEQ12vbA&AMOUNT=500
```

Cancelation of a Payment with CreatePayComplete

```
https://test.saferpay.com/hosting/paycompletev2.asp?spPassword=8e7Yn5yk&ACCOUNTID=401860-
17795278&ID=EvrKOEApM3YtSApnE0MlAU28nCYb&ACTION=Cancel
```

Initiating the Batch Close with CreatePayComplete

```
https://test.saferpay.com/hosting/paycompletev2.asp?spPassword=8e7Yn5yk&ACCOUNTID=401860-
17795278&ACTION=CloseBatch
```

PayComplete Response

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

9 RESULT Values

An Authorization has only been processed successfully if RESULT=0 is returned. With all other RESULT values the request failed.

Wert	Description	Explanation
Authorization response:		
5	Access denied	Access for IP address is not allowed for this account.
61	Invalid card	This card failed the validity check.
62	Invalid date	Invalid expiration date.
63	Card expired	The card has expired.
64	Unknown card	The card is unknown; it could not be allocated to a card type.
65	Authorization declined	The processor has denied the transaction request. The refusal reason code provided by the acquirer is returned within the field AUTHRESULT.
67	No contract available	On the concerned Saferpay terminal there is no contract for the requested card/currency combination.
70	Geo IP not white listed	The origin country of the transmitted IP is not permitted in the Saferpay Risk Management.
83	Invalid currency	The specified currency code is invalid.
84	Invalid amount	The specified amount is invalid.
85	No credits available	No more transaction points available.
102	Function not supported	The processor does not support the requested function.
104	PAN black listed	The card was blocked by Saferpay Risk Management
105	Card country not white listed	Card country not listed in the Saferpay Risk Management.
113	CVC wrong value	The CVC contains an invalid value.
114	CVC mandatory	The submission of the CVC is mandatory.
8100	Referenced transaction not found	The submitted MPI_SESSIONID is unknown.

10 Contact

10.1 Saferpay Integration Team

Do you have questions about this document or problems with the integration of Saferpay or do you need assistance? Then please contact our integration team:

Saferpay Switzerland
SIX Payment Services AG
Hardturmstrasse 201
8021 Zürich
+41 848 66 44 44
www.six-payment-services.com/saferpay
integration.saferpay@six-payment-services.com

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

10.2 Saferpay Support Team

Do you have questions about error messages or do you encounter problems with your running system? Then please contact our support team:

Saferpay Switzerland
SIX Payment Services AG
Hardturmstrasse 201
8021 Zürich
+41 848 66 44 44
www.six-payment-services.com/saferpay
support.saferpay@six-payment-services.com

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

The Saferpay team wishes you every success with your Saferpay e-payment solution!