Your step-by-step guide to Connect

Version 2021-1 (IPG)

Contents

Getting Support

There are different manuals available for Clover's eCommerce solutions. This Integration Guide will be the most helpful for integrating hosted payment forms or a Direct Post.

For information about settings, customization, reports and how to process transactions manually (by keying in the information) please refer to the User Guide Virtual Terminal.

If you have read the documentation and cannot find the answer to your question, please contact your local support team.

1. Introduction

The Connect solution provides a quick and easy way to add payment capabilities to your website.

Connect manages the customer redirections that are required in the checkout process of many payment methods or authentication mechanisms and gives you the option to use secure hosted payment pages which can reduce the burden of compliance with the Data Security Standard of the Payment Card Industry (PCI DSS).

This document describes how to integrate your website using Connect and provides step by step instructions on how to quickly start accepting payments from your webshop.

When making decisions on your way of integration, please consider that we do not recommend to use the hosted payment forms inside an iFrame since some Internet browsers do not allow cookies to be sent to the 3rd party hosts, moreover some features (e.g.: 3D Secure authentications) and some Alternative Payment methods that involve redirections to the 3rd party services (e.g. PayPal) do not allow displaying their screens within an iFrame.

Depending on your business processes, it can also make sense to additionally integrate our Web Service API solution (see Web Service API Integration Guide).

2. Payment process options

The Connect solution provides a number of different options for the payment process to support integrations where you handle most of the customer interactions on your own website up to integrations where you use ready-made form pages for the entire payment process.

2.1 Hosted Payments Page

If you want to fully outsource the payment process in order not to have any sensitive cardholder data on your systems, you can use our ready-made hosted pages for your customers to enter their payment information.

The most important aspect around the usage of hosted payment page is the security of sensitive cardholder data. When you decide to let your customers enter their credit card details on the page that we provide and host on our servers for this purpose, it facilitates your compliance with the Data Security Standard of the Payment Card Industry (PCI DSS) as the payment processing is completely hosted by Clover.

For a standard hosted payment page integration, you should use the checkout option 'combinedpage' that consolidates the payment method choice and the typical next step (e.g.: entry of card details or selection of bank) in a single page, which gets automatically optimized for different kinds of user devices (e.g.: PC, smartphone, tablet, etc.).

The hosted page is localized in many languages and can be easily customized with your merchant's logo, colors, and font types to make it fit to the look and feel of your shop environment (refer to the User Guide Virtual Terminal to learn more). It also shows your merchant's name (i.e.: legal name) and allows you to display a summary of the purchased items to your customer in the 'Your Order' box.

If you do not want to let your customer select the payment method on our hosted page but want to handle that part upfront within your shop environment, you should submit a value for the parameter 'paymentMethod' in your request to the gateway. In addition, if you do not want to distinguish between

different card brands (but just card vs. alternative payment methods), you can send a valid card brand value for the parameter 'paymentMethod' and your customer will see a hosted page for the card details entry with no card brand logo shown. Please contact your local support team if you want to enable this feature. This will be managed with a specific setting performed on your account (store) ('hideCardBrandLogoInCombinedPage').

If you do not submit a value for the parameter 'paymentMethod', the gateway will take your customer to a hosted page to choose from the payment methods activated for your store.

If you do not include in your request the fields like e.g.: the card number or the expiry date for a card payment, the gateway will take your customer to a hosted page to collect this information as being mandatory for a transaction processing.

2.2 Direct Post

In the scenarios where you prefer not to use a hosted payment page, you can submit the required customer data directly from your own form to Clover, but please be aware that if you store or process sensitive cardholder data within your own application, you must ensure that your system components are compliant with the Data Security Standard of the Payment Card Industry (PCI DSS).

You create the payment form and display it within your website or app. When your customer has entered the card details and presses the "continue button", the customer's device sends the payment information directly to the gateway.

If you choose the Direct Post option and create your own forms, there are additional fields that must be included in your transaction request to the gateway, which are listed in the chapter on using your own forms to capture the data.

3. Getting Started

This section provides a simple example on how to integrate your website using the "combinedpage" checkout option. Examples are provided using ASP and PHP. This section assumes that the developer has a basic understanding of his chosen scripting language.

3.1 Checklist

In order to integrate with the payment gateway, you must have the following items:

- Store Name

  This is the ID of the store that was given to you by Clover.
  For example: 10123456789
-
- Shared Secret

  This is the shared secret provided to you by Clover.
  This is used when constructing the hash value (see below).

3.2 ASP Example

The following ASP example demonstrates a simple page that will communicate with the payment gateway.

When the cardholder clicks *Submit*, they are redirected to the Clover secure page to enter the card details. After payment has been completed, the user will be redirected to the merchant's receipt page. The location of the receipt page can be configured.

```
<html>
<head><title>IPG Connect Sample for ASP</title></head>
<body>
<p><h1>Order Form</h1></p>
<form method="post" action=" https://test.ipg-online.com/connect/gateway/processing
">
<input type="hidden" name="txntype" value="sale">
<input type="hidden" name="timezone" value="Europe/Berlin"/>
<input type="hidden" name="txndatetime" value="<% getDateTime() %>"/>
<input type="hidden" name="hash_algorithm" value="HMACSHA256"/>
<input type="hidden" name="hashExtended" value="<% call
createExtendedHash("13.00","978") %>"/>
<input type="hidden" name="storename" value="10123456789" />
<input type="hidden" name="checkoutoption" value="combinedpage"/>
<input type="hidden" name="paymentMethod" value="M"/>
<input type="text" name="chargetotal" value="13.00" />
<input type="hidden" name="currency" value="978"/>
<input type="submit" value="Submit">
</form>
</body>
</html>
```

The code presented in Appendix II represents the included file ipg-util.asp. It includes code for generating a hash as is required by Clover. The provision of a hash in the example ensures that this merchant is the only merchant that can send in transactions for this store.

Note, the POST URL used is for integration testing only. When you are ready to go into production, please contact Clover and you will be provided with the live production URL.

Note, the included file, ipg-util.asp uses a server side JavaScript file to build the hash. This file can be provided on request. To prevent fraudulent transactions, it is recommended that the hash is calculated within your server and JavaScript is not used like shown in the samples mentioned.

3.3 PHP Example

The following PHP example demonstrates a simple page that will communicate with the payment gateway.

When the cardholder clicks *Submit*, they are redirected to the Clover secure page to enter the card details. After payment has been completed, the user will be redirected to the merchant's receipt page. The location of the receipt page can be configured.

```
<html>
<head><title>IPG Connect Sample for PHP</title></head>
<body>
<p><h1>Order Form</h1>
<form method="post" action="https://test.ipg-
online.com/connect/gateway/processing">
<input type="hidden" name="txntype" value="sale">
<input type="hidden" name="timezone" value="Europe/Berlin"/>
<input type="hidden" name="txndatetime" value="<?php echo getDateTime() ?>"/>
```

```
<input type="hidden" name="hash_algorithm" value="HMACSHA256"/>
<input type="hidden" name="hashExtended" value="<?php echo
createExtendedHash("13.00","978") ?>"/>
<input type="hidden" name="storename" value="10123456789"/>
<input type="hidden" name="checkoutoption" value="combinedpage"/>
<input type="hidden" name="paymentMethod" value="M"/>
<input type="text" name="chargetotal" value="13.00"/>
<input type="hidden" name="currency" value="978"/>
<input type="submit" value="Submit">
</form>
</body>
</html>
```

Note that the POST URL used in this example is for integration testing only. When you are ready to go into production, please contact Clover and you will be provided with the live production URL.

The code presented in Appendix III represents the included file ipg-util.php. It includes code for generating a hash as is required by Clover. The provision of a hash in the example ensures that this merchant is the only merchant that can send in transactions for this store.

3.4 Amounts for test transactions

When using our test system for integration, odd amounts (e. g. 13.01 EUR or 13.99 EUR) can cause the transaction to decline as these amounts are sometimes used to simulate unsuccessful authorizations.

We therefore recommend using even amounts for testing purpose, e. g. 13.00 EUR like in the example above.

4. Mandatory Fields

Depending on the transaction type, the following form fields must be present in the form being submitted to the payment gateway (X = mandatory field). Please refer to this Integration Guide's Appendixes for implementation details in relation to alternative payment methods and the other product options.

Field Name	Description, possible values and format	Sale transaction	PreAuth*	PostAuth*	Void	PayerAuth**
txntype	'sale', 'preauth', 'postauth', 'void' or 'payer_auth' (the transaction type – please note the descriptions of transaction types in the User Guide Virtual Terminal) The possibility to send a 'void' using the Connect interface is restricted. Please contact your local support team if you want to enable this feature.	X (sale)	X (preauth)	X (postauth)	X (void)	X (payer_auth)
timezone	Time zone of the transaction in Area/Location format, e.g. Africa/Johannesburg America/New_York America/Sao_Paulo Asia/Calcutta Australia/Sydney	X	X	X	X	X

	Europe/Amsterdam Europe/Berlin Europe/Dublin Europe/London Europe/Rome					
txndatetime	YYYY:MM:DD-hh:mm:ss (exact time of the transaction)	X	X	X	X	X
hash_algorithm	This is to indicate the algorithm that you use for hash calculation. The possible values are: ▪ HMACSHA256 ▪ HMACSHA384 ▪ HMACSHA512 Only one algorithm value should be used.	X	X	X	X	X
hashExtended	The extended hash needs to be calculated using all non-empty gateway specified request parameters in ascending order of the parameter names, where the upper-case characters come before the lower case (based on ASCII value) and the shared secret must be used as the secret key for calculating the hash value. When you are using Direct Post, there is also an option where you do not need to know the card details (PAN, CVV and Expiry Date) for the hash calculation. This will be managed with a specific setting performed on your store. Please contact your local support team if you want to enable this feature. An example of how to generate a hash is given in Appendix I.	X	X	X	X	X
storename	This is the ID of the store provided by Clover.	X	X	X	X	X
chargetotal	This is the total amount of the transaction using a dot or comma as decimal separator, e. g. 12.34 for an amount of 12 Euro and 34 Cent. Group separators like1,000.01 / 1.000,01 are not allowed.	X	X	X	X	X
checkoutoption	Set the value for this parameter to 'combinedpage' for a standard hosted payment page integration.	X	X			X
currency	The numeric ISO code of the transaction currency, e. g. 978 for Euro	X	X	X	X	X
oid	The order ID of the initial action a PostAuth shall be initiated for.			X		
ipgTransactionId *or* merchantTransactionId	Exact identification of a transaction that shall be voided. You receive this value as result parameter, 'ipgTransactionId' of the corresponding transaction. Alternatively, 'merchantTransactionId' can be used for the Void in case the merchant has assigned one.				X	

* The transaction types 'preauth' and 'postauth' only apply to the payment methods credit card, PayPal.
** The transaction type 'payer_auth' is only required if you want to split the 3D Secure authentication process from the payment transaction (authorization) process. See more information in the 3D Secure section of this guide.

5. Optional Form Fields

Field Name	Description, possible values and format
cardFunction	This field allows you to indicate the card function in case of combo cards which provide credit and debit functionality on the same card. It can be set to 'credit' or 'debit'. The field can also be used to validate the card type in a way that transactions where the submitted card function does not match the card's capabilities will be declined. If you e.g. submit "cardFunction=debit" and the card is a credit card, the transaction will be declined.
comments	Place any comments here about the transaction.
customerid	This field allows you to transmit any value, e. g. your ID for the customer.
dccInquiryId	Inquiry ID for a Dynamic Pricing request. Used to send the Inquiry ID you have obtained via a Web Service API call ('RequestMerchantRateForDynamicPricing'). This value will be used to retrieve the currency conversion information (exchange rate, converted amount) for this transaction.
dccSkipOffer	If the cardholder declines the currency conversion offer within your environment, the request parameter 'dccSkipOffer' can be set to 'true' so that the hosted consumer dialogue will automatically be skipped.
dynamicMerchantName	The name of the merchant to be displayed on the cardholder's statement. The length of this field should not exceed 25 characters. If you want to use this field, please contact your local support team to verify if this feature is supported in your country.
hideOrderDetails	Set this parameter to 'true' when you want to hide (remove) the 'Your Order' box from our hosted payment page.
invoicenumber	This field allows you to transmit any value, e. g. an invoice number or class of goods. Please note that the maximum length for this parameter is 48 characters.
item1 up to item999	Line items are regular Connect integration key-value parameters (URL-encoded), where: ▪ the name is a combination of the keyword item and a number, where the number indicates the list position e.g.: item1 ▪ the value is represented by a semicolon-separated list of values, where the position indicates the meaning of the list item property e.g.: <1>;<2>;<3>;<4>;<5>;<6>;<7> The 'item1' to 'item999' parameters allow you to send basket information in the following format: *id;description;quantity;item_total_price;sub_total;vat_tax;shipping* 'shipping' always has to be set to '0' for single line item. If you want to include a shipping fee for an order, please use the predefined *id* IPG_SHIPPING. For other fees that you may want to add to the total order, you can use the predefined *id* IPG_HANDLING. When you want to apply a discount, you should include an item with a negative amount and change accordingly the total amount of the order. Do not forget to regard the 'quantity' when calculating the values e.g.: subtotal and VAT since they are fixed by items. Examples: A;Product A;1;5;3;2;0

<table>
<tr><td></td><td>B;Product B;5;10;7;3;0
C;Product C;2;12;10;2;0
D;Product D;1;-1.0;-0.9;-0.1;0
IPG_SHIPPING;Shipping costs;1;6;5;1;0
IPG_HANDLING;Transaction fee;1;6.0;6.0;0;0</td></tr>
<tr><td>language</td><td>This parameter can be used to override the default payment page language configured for your merchant store.
The following values are currently possible:
<table>
<tr><th>Language</th><th>Value</th></tr>
<tr><td>Chinese (simplified)</td><td>zh_CN</td></tr>
<tr><td>Chinese (traditional)</td><td>zh_TW</td></tr>
<tr><td>Czech</td><td>cs_CZ</td></tr>
<tr><td>Danish</td><td>da_DK</td></tr>
<tr><td>Dutch</td><td>nl_NL</td></tr>
<tr><td>English (USA)</td><td>en_US</td></tr>
<tr><td>English (UK)</td><td>en_GB</td></tr>
<tr><td>Finnish</td><td>fi_FI</td></tr>
<tr><td>French</td><td>fr_FR</td></tr>
<tr><td>German</td><td>de_DE</td></tr>
<tr><td>Greek</td><td>el_GR</td></tr>
<tr><td>Hungarian</td><td>hu_HU</td></tr>
<tr><td>Italian</td><td>it_IT</td></tr>
<tr><td>Japanese</td><td>ja_JP</td></tr>
<tr><td>Norwegian (Bokmål)</td><td>nb_NO</td></tr>
<tr><td>Polish</td><td>pl_PL</td></tr>
<tr><td>Portuguese (Brazil)</td><td>pt_BR</td></tr>
<tr><td>Serbian (Serbia)</td><td>sr_RS</td></tr>
<tr><td>Slovak</td><td>sk_SK</td></tr>
<tr><td>Spanish (Spain)</td><td>es_ES</td></tr>
<tr><td>Spanish (Mexico)</td><td>es_MX</td></tr>
<tr><td>Swedish</td><td>sv_SE</td></tr>
</table></td></tr>
<tr><td>merchantTransactionId</td><td>Allows you to assign a unique ID for the transaction. This ID can be used to reference to this transaction in a PostAuth or Void request (referencedMerchantTransactionId).</td></tr>
<tr><td>mobileMode</td><td>If your customer uses a mobile device for shopping at your online store you can submit this parameter with the value ‘true’, when using the ‘classic’ checkout option. This will lead your customer to a payment page flow that has been specifically designed for mobile devices.</td></tr>
<tr><td>mode</td><td>The legacy checkout option specific parameter: If you are building a payment request for the Sale, PreAuth or PayerAuth transaction, when using the ‘classic’ checkout option, your request needs to include a value for one of the three different modes to define the range of data that shall be captured by the gateway:
‘payonly’ - shows a hosted page to collect the minimum set of information for the transaction (e. g. cardholder name, card number, expiry date and card code for a credit card transaction),
‘payplus’ - in addition to the above, the payment gateway collects a full set of billing information on an additional page,
‘fullpay’ - in addition to the above, the payment gateway displays a third page to also collect shipping information.</td></tr>
<tr><td>numberOfInstallments</td><td>This parameter allows you to set the number of instalments for a Sale transaction if your customer pays the amount in several parts.</td></tr>
<tr><td>installmentsInterest</td><td>This parameter allows you to choose, if instalment interest should be applied or not, the values “true” or “false” are currently possible.</td></tr>
<tr><td>installmentDelayMonths</td><td>This parameter allows you to delay the first instalment payment for several months, values 2-99 are currently possible.</td></tr>
</table>

oid	This field allows you to assign a unique ID for your order. If you choose not to assign an order ID, the Clover system will automatically generate one for you.
parentUri	If you plan to embed our hosted payment pages inside an iFrame you must use this parameter, with the maximum length of 30 characters, to specify an URL of a page, where the hosted payment page will be embedded. However, note that we do not recommend using the hosted payment forms inside an iFrame since some Internet browsers do not allow cookies to be sent to the 3rd party hosts, moreover some features (e.g.: 3D Secure authentications) and some Alternative Payment methods that involve redirections to the 3rd party services (e.g.: PayPal) do not allow displaying their screens within an iFrame.
paymentMethod	If you let the customer select the payment method (e. g. MasterCard, Visa,) in your shop environment or want to define the payment type yourself, transmit the parameter 'paymentMethod' along with your Sale or PreAuth transaction. If you do not submit this parameter, the payment gateway will display a drop-down menu to the customer to choose from the payment methods available for your shop.
ponumber	This field allows you to submit a Purchase Order Number with up to 50 characters.
refer	This field describes who referred the customer to your store.
referencedMerchantTransactionID	This field allows to reference to a merchantTransactionId of a transaction when performing a Void. This can be used as an alternative to ipgTransactionId if you assigned a merchantTransactionId in the original transaction request.
referencedSchemeTransactionId	Credentials on file (COF) specific parameter. This field allows you to include in your request 'schemeTransactionId' that has been returned in the response of the initial transaction in order to provide a reference to the original transaction, which stored the credentials for the first time.
responseFailURL	The URL where you wish to direct customers after a declined or unsuccessful transaction (your Sorry URL) – only needed if not setup in Virtual Terminal / Customisation.
responseSuccessURL	The URL where you wish to direct customers after a successful transaction (your Thank You URL) – only needed if not setup in Virtual Terminal / Customisation.
shipping	This parameter can be used to submit the shipping fee, in the same format as 'chargetotal'. If you submit 'shipping', the parameters 'subtotal' and 'vattax' have to be submitted as well. Note that the 'chargetotal' has to be equal to 'subtotal' plus 'shipping' plus 'vattax'.
trxOrigin	This parameter allows you to use the secure and hosted payment form capabilities within your own application. Possible values are: ▪ 'MOTO' (For Mail Order or Telephone Order transactions). ▪ 'MAIL' (for transactions where the payment details are captured manually and provided in written form the Card Code entry is not allowed). ▪ 'PHONE' (for transactions where you have received the order over the phone and enter the payment details yourself the Card Code entry is required). ▪ 'ECI' (for standard usage in an eCommerce environment where your customer enters the payment details).
unscheduledCredentialOnFileType	Credentials on file (COF) specific parameter. This field allows you to flag transactions as unscheduled credential on file type. Currently the valid values are: FIRST, CARDHOLDER_INITIATED or MERCHANT_INITIATED to advise the scenario if the credential is stored on your side.
vattax	This field allows you to submit an amount for Value Added Tax or other taxes, e.g.: GST in Australia. Please ensure the sub total amount plus shipping plus tax equals the charge total.

6. Using your own forms to capture the data

If you decide to create your own forms, i.e.: Direct Post (not to use the ones provided and hosted by Clover), there are additional mandatory fields that you need to include. These fields are listed in the following sections.

Using Direct Post allows you to have full control over the look and feel of the form where your customers enter their card details for payment while simultaneously avoiding the need to have sensitive card data within your systems.

It is also important that you check if JavaScript is activated in your customer's browser. If necessary, inform your customer that JavaScript needs to be activated for the payment process.

6.1 Capture payment details

After your customer has decided how to pay, you present a corresponding HTML-page with a form to enter the payment data as well as hidden parameters with additional transaction information. In addition to the mandatory fields, your form needs to contain the following fields (part of them can be hidden).

Field Name	Description, possible values and format	Credit Card (+ Visa Debit/Electron/Delta)	Maestro
cardnumber	Your customer's card number. 12-24 digits.	X	X
expmonth	The expiry month of the card (2 digits)	X	X
expyear	The expiry year of the card (4 digits)	X	X
cvm	The card code, in most cases on the backside of the card (3 to 4 digits)	X	X as an optional field "if on card"
bname	Name of the bank account owner that will be debited (alphanumeric characters, spaces, and dashes limited to 96)		X

6.2 Capture billing information

It is possible to additionally transfer billing information to the payment gateway. The following table describes the format of these additional fields:

Field Name	Possible Values	Description
bcompany	Alphanumeric characters, spaces, and dashes limited to 96	Customers Company
bname	Alphanumeric characters, spaces, and dashes limited to 96	Customers Name
baddr1	Limit of 96 characters, including spaces	Customers Billing Address 1
baddr2	Limit of 96 characters, including spaces	Customers Billing Address 2
bcity	Limit of 96 characters, including spaces	Billing City
bstate	Limit of 96 characters, including spaces	State, Province or Territory
bcountry	2 Letter Country Code	Country of Billing Address
bzip	Limit of 24 characters, including spaces	Zip or Postal Code
phone	Limit of 32 Characters	Customers Phone Number
fax	Limit of 32 Characters	Customers Fax Number
email	Limit of 254 Characters	Customers Email Address

6.3 Capture shipping information

It is possible to additionally transfer shipping information to the payment gateway. The billing information is as specified above. The following table describes the format of the shipping fields:

Field Name	Possible Values	Description
sname	Alphanumeric characters, spaces, and dashes limited to 96	Ship-to Name
saddr1	Limit of 96 characters, including spaces	Shipping Address Line 1
saddr2	Limit of 96 characters, including spaces	Shipping Address Line 2
scity	Limit of 96 characters,	Shipping City

	including spaces	
sstate	Limit of 96 characters, including spaces	State, Province or Territory
scountry	2 letter country code	Country of Shipping Address
szip	Limit of 24 characters, including spaces	Zip or Postal Code

6.4 Validity checks

Prior to the authorization request for a transaction, the payment gateway performs the following validation checks:

- The expiry date of cards needs to be in the future
- The Card Security Code field must contain 3 or 4 digits
- The structure of a card number must be correct (LUHN check)

If the submitted data should not be valid, the payment gateway presents a corresponding data entry page to the customer.

To avoid this hosted page when using your own input forms for the payment process, you can transmit the following additional parameter along with the transaction data:

full_bypass=true

In that case you get the result of the validity check back in the transaction response and can display your own error page based on this.

Please note, if the transaction is eligible for DCC (your store is configured for DCC and the customer is paying by credit card capable of DCC), your customer will be presented the DCC page despite having full_bypass set to true. This is due to regulatory reasons. You can avoid displaying of DCC choice pages by doing the DCC Inquiry yourself via our Web Service API (RequestMerchantRateForDynamicPricing).

7. Additional Custom Fields

You may want to use further fields to gather additional customer data geared toward your business specialty, or to gather additional customer demographic data which you can then store in your own database for future analysis. You can send as many custom fields to the payment gateway as you wish, and they will get returned along with all other fields to the response URL.

Up to ten custom fields can be submitted in a way that they will be stored within the gateway so that they appear in the Virtual Terminal's Order Detail View as well as in the response to Inquiry Actions that you send through our Web Service API.

Field Name	Description, possible values and format
customParam_key	If you want to use this feature, please send the custom fields in the format customParam_key=value. The maximum length of a custom parameter is 100 characters. Example:<input type="hidden" name="customParam_color" value="green"/>

8. 3D Secure

The Connect solution includes the ability to authenticate transactions using Verified by Visa, MasterCard SecureCode, American Express SafeKey, JCB J/Secure and Diners ProtectBuy to provide an additional security layer for online card transactions.

If your store is enabled for 3D Secure, all Sale or preAuth transactions that you initiate by posting an HTML form will by default go through the 3D Secure process without the need for you to do anything, i.e. cardholders with an enrolled card will see a page from the card issuer to enter the password unless the card issuer decides not to check it.

The generic fields to be considered:

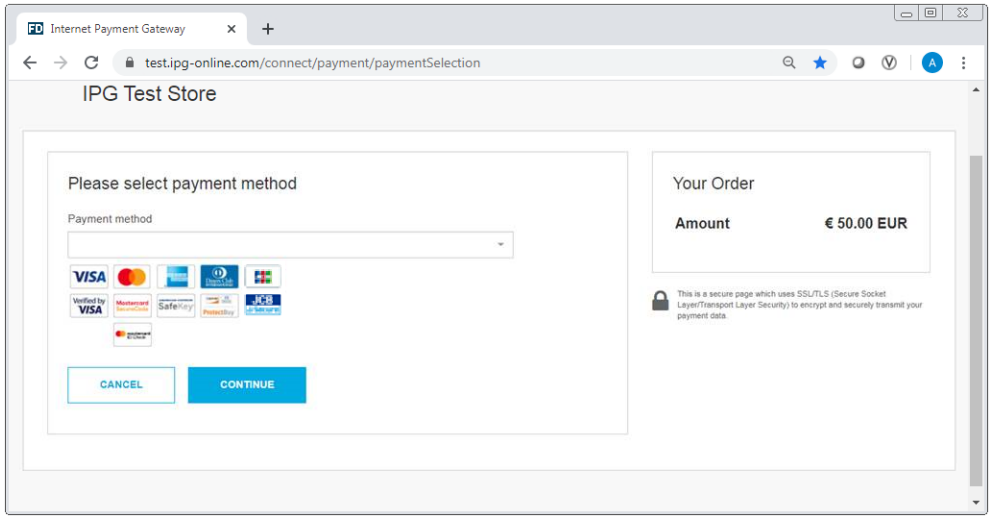
Field Name	Description, possible values and format
authenticateTransaction	Optional parameter to be set either to 'true' or 'false' to enable or disable 3D Secure authentication on a Transaction-by-Transaction basis. Example for a transaction with 3D Secure: <input type="hidden" name="authenticateTransaction" value="true"/> Example for a transaction without 3D Secure: <input type="hidden" name="authenticateTransaction" value="false"/>
threeDSRequestorChallengeIndicator	Optional parameter for EMV 3D Secure (2.0) to be set to: 01,02,03,04 in order to indicate the preferred type of authentication: ▪ 01 - no preference (set as default value) ▪ 02 - no challenge requested ▪ 03 - challenge requested 3DS requestor preference ▪ 04 - challenge requested mandate
threeDSTransType	The parameter for EMV 3D Secure (2.0) represents the type of purchased item, mandatory for Visa and Brazilian market, otherwise optional. If no specific value present in the transaction request, default value is used. ▪ 01 - Goods/ Service Purchase (default value) ▪ 03 - Check Acceptance ▪ 10 - Account Funding ▪ 11 - Quasi-Cash Transaction ▪ 28 - Prepaid Activation and Load
scaExemptionIndicator1	Optional parameter to request an exemption from Strong Customer Authentication (SCA) without the need to perform 3-D Secure authentication. Currently available values: ▪ Low Value Exemption

	▪ TRA Exemption ▪ Trusted Merchant Exemption ▪ SCP Exemption Note this parameter is relevant only for the European merchants impacted by the PSD2 requirements.
skipTRA	This optional parameter allows you to use 3D Secure even if the transaction has been evaluated as low risk and would be eligible for an exemption. Currently available values: ▪ true ▪ false When your store has been set up with Transaction Risk Analysis (TRA) service, but you do want to force 3D Secure authentication for a certain transaction, set 'skipTRA' to 'true'. Note this parameter is relevant only for the European merchants impacted by the PSD2 requirements.
oid	Use this optional parameter to assign an identifier for your order; in case you plan to authenticate the transaction using EMV 3DS protocol (aka 3DS 2.1) only the following characters are allowed: ▪ A-Z, a-z, 0-9, "-"

In principle, it may occur that 3D Secure authentications cannot be processed successfully for technical reasons. If one of the systems involved in the authentication process is temporarily not responding, the payment transaction will be processed as a "regular" eCommerce transaction (ECI 7). A liability shift to the card issuer for possible chargebacks is not warranted in this case. If you prefer that such transactions shall not be processed at all, our technical support team can block them for your Store on request.

Credit card transactions with 3D Secure hold in a pending status while cardholders search for their password or need to activate their card for 3D Secure during their shopping experience. During this time when the final transaction result of the transaction is not yet determined, the payment gateway sets the Approval Code to „?:waiting 3dsecure". If the session expires before the cardholder returns from the 3D Secure dialogue with his bank, the transaction will be shown as "N:-5103:Cardholder did not return from ACS".

Please note that the technical process of 3D Secure transactions differs in some points compared to a normal transaction flow. If you already have an existing shop integration and plan to activate 3D Secure subsequently, we recommend performing some test transactions on our test environment.



8.1 3DSecure Split Authentication

If your business or technical processes require the cardholder authentication to be separated from the payment transaction (authorization), you can use the transaction type 'payer_auth'. This transaction type only performs the authentication (and stores the authentication results).

Example of a 'payer_auth' request:

```
<!-- #include file="ipg-util.asp"-->

<html>
<head><title>IPG Connect Sample for ASP</title></head>
<body>
<p><h1>Order Form</h1></p>

<form method="post" action=" https://test.ipg-online.com/connect/gateway/processing
">
    <input type="hidden" name="txntype" value="payer_auth">
                <input type="hidden" name="timezone" value="Europe/Berlin"/>
                <input type="hidden" name="txndatetime" value="<% getDateTime()
%>"/>
                <input type="hidden" name="hash_algorithm" value="HMACSHA256"/>
                <input type="hidden" name="hashExtended" value="<% call
createExtendedHash( "13.00","978" ) %>"/>
                <input type="hidden" name="storename" value="10123456789" />
    <input type="hidden" name="checkoutoption" value="combinedpage"/>
                <input type="hidden" name="paymentMethod" value="M"/>
    <input type="text" name="chargetotal" value="13.00" />
    <input type="hidden" name="currency" value="978"/>
                <input type="hidden" name="authenticateTransaction" value="true"/>
<input type="submit" value="Submit">
</form>
</body>
</html>
```

Example of a 'payer_auth' response:

```
{txndate_processed=17/04/20 17:17:32,
ccbin=542606,
timezone=Europe/Berlin,
oid=C-2101f68a-45e9-4f3c-a6da-1337d5574717,
cccountry=N/A,
expmonth=12,
hash_algorithm=HMACSHA256
```

```
currency=978,
chargetotal=13.00,
approval_code=Y:ECI2/5:Authenticated,
hiddenSharedsecret=sharedsecret,
hiddenTxndatetime=2020:04:17-17:32:41,
expyear=2024,
response_hash=LarWYFSNgEToq13HlvyslX6hywi2T/nMn8jMY+1kxkI=,
response_code_3dsecure=1,
hiddenStorename=10123456789,
transactionNotificationURL=https://test.ipg-
online.com/webshop/transactionNotification,
tdate=1491824253,
ignore_refreshTime=on,
ccbrand=MASTERCARD,
txntype=payer_auth,
paymentMethod=M,
txndatetime=2020:04:17-17:32:41,
cardnumber=(MASTERCARD) ... 4979,
ipgTransactionId=84120276797,
status=APPROVED}
```

In a second step, you need to submit a payment transaction ('sale' or 'preauth') via the IPG Web Service API and reference it to the prior authentication. To review an example of a 'sale' transaction that refers to a previous 'payer_auth' transaction, please review the 3DSecure Split Authentication section, in the Web Service API integration guide.

8.2 Dynamic 3D Secure based on the card issuer's country

With the Dynamic 3D Secure product option, you can exclude specific card transactions from the 3D Secure authentication based on a certain country selection (i.e.: issuing country) e.g.: Germany, Switzerland and Austria, while apply the standard 3D Secure authentication process for other transactions with card from other countries.

You can improve the consumer experience for the cardholders from the selected countries, while the chargeback risk for such transactions is still with you.

If you have ordered this product option, the countries that should be excluded from the 3D Secure authentication process can be set up for you by your local support team.

In case of some specific high-risk transactions, you can override this setting on transaction level and force the 3D Secure authentication on a Transaction-by-Transaction basis, even if the card used is issued in a country, which has been defined by you as a country where 3D Secure authentication should not be applied. In order to do it, you have to send the parameter 'override3dsCountryExclusion' set to "true" then the country setting will be ignored and the 3D Secure authentication process applied.

Field Name	Description, possible values and format
override3dsCountryExclusion	Optional parameter to be set either to 'true' or 'false'. Set to 'true' if for a transaction you would like to enforce 3D Secure authentication, despite this country possibly being exempted from authentication due to the merchant configured list of countries, where 3D Secure is not required.

9. MCC 6012 Mandate in UK

For UK-based Financial Institutions with Merchant Category Code 6012, Visa and MasterCard have mandated additional information of the primary recipient of the loan to be included in the authorization message.

If you are a UK 6012 merchant use the following parameters for your transaction request:

Field Name	Description, possible values and format
mcc6012BirthDay	Date of birth in format dd.mm.yyyy
mcc6012AccountFirst6	First 6 digits of recipient PAN (where the primary recipient account is a card)
mcc6012AccountLast4	Last 4 digits of recipient PAN (where the primary recipient account is a card)
mcc6012AccountNumber	Recipient account number (where the primary recipient account is not a card)
mcc6012Surname	Surname
mcc6012Zip	Post Code

If you are a UK 6051 and 7299 merchant, you can reuse the MCC 6012 parameters to send the optional data to be included in the authorization message. However, please note that you have to either populate all the parameters or none otherwise the transaction will be declined.

10. Data Vault

With the Data Vault product option, you can store sensitive cardholder data in an encrypted database in Clover's data center to use it for subsequent transactions without the need to store this data within your own systems.

If you have ordered this product option, the Connect solution offers you the following functions:

- Store or update payment information when performing a transaction

  Additionally send the parameter 'hosteddataid' together with the transaction data as a unique identification for the payment information in this transaction. Depending on the payment type, credit card number and expiry date will be stored under this ID if the transaction has been successful. In cases where the submitted 'hosteddataid' already exists for your store, the stored payment information will be updated.

  If you want to assign multiple IDs to the same payment information record, you can submit the parameter 'hosteddataid' several times with different values in the same transaction.

  If you prefer not to assign a token yourself but want to let the gateway do this for you, send the parameter 'assignToken' and set it to 'true'. The gateway will then assign a token and include it in the transaction response as 'hosteddataid'.

  If you have use cases where you need some of the tokens for single transactions only (e.g.: for consumers that check out as a "guest", use the additional parameter 'tokenType' with the values 'ONETIME' (card details will only be stored for a short period of time) or 'MULTIPAY' (card details will be stored for use in future transactions).

- Initiate payment transactions using stored data

If you stored cardholder information using the Data Vault option, you can perform transactions using the 'hosteddataid' without the need to pass the credit card or bank account data again. Please note that it is not allowed to store the card code (in most cases on the back of the card) so that for credit card transactions, the cardholder still needs to enter this value. If you use Clover's hosted payment forms, the cardholder will see the last four digits of the stored credit card number, the expiry date and a field to enter the card code.

When using multiple Store IDs, it is possible to access stored card data records of a different Store ID then the one that has been used when storing the record. In that way you can for example use a shared data pool for different distributive channels. To use this feature, submit the Store ID that has been used when storing the record as the additional parameter 'hosteddatastoreid'.

- Avoid duplicate cardholder data for multiple records

  To avoid customers using the same cardholder data for multiple user accounts, the additional parameter 'declineHostedDataDuplicates' can be sent along with the request. The valid values for this parameter are 'true'/'false'. If the value for this parameter is set to 'true' and the cardholder data in the request is already found to be associated with another 'hosteddataid', the transaction will be declined.

See further possibilities with the Data Vault product in the Integration Guide for the Web Service API.

11.Recurring Payments

For credit card and PayPal transactions, it is possible to install recurring payments using Connect. To use this feature, the following additional parameters will have to be submitted in the request:

Field Name	Possible Values	Description
recurringInstallmentCount	Number between 1 and 999	Number of installments to be made including the initial transaction submitted
recurringInstallmentPeriod	day week month year	The periodicity of the recurring payment
recurringInstallmentFrequency	Number between 1 and 99	The time period between installments
recurringComments	Limit of 100 characters, including spaces	Any comments about the recurring transaction

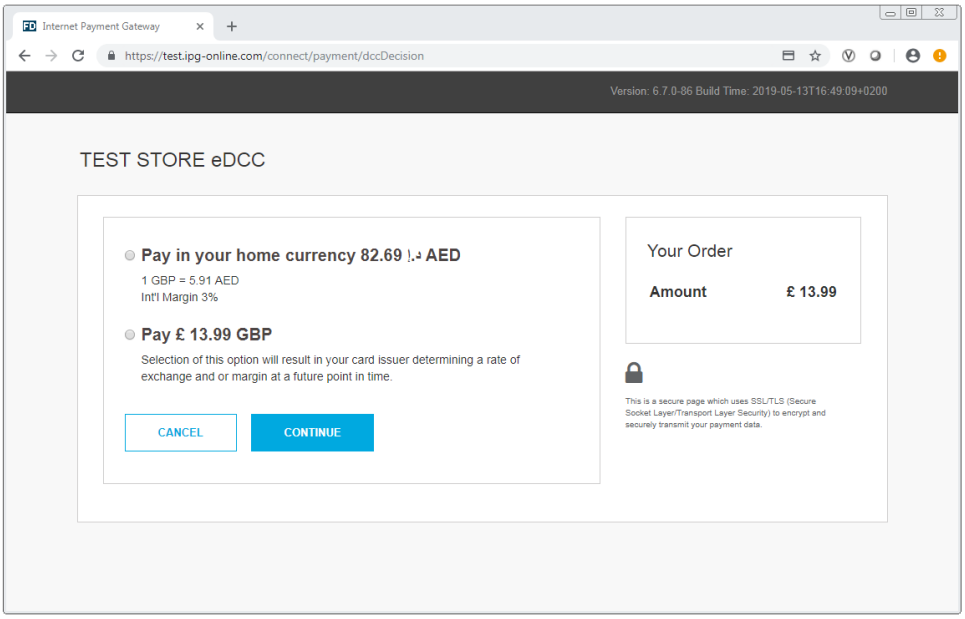
Note that the start date of the recurring payments will be the current date and will be automatically calculated by the system.

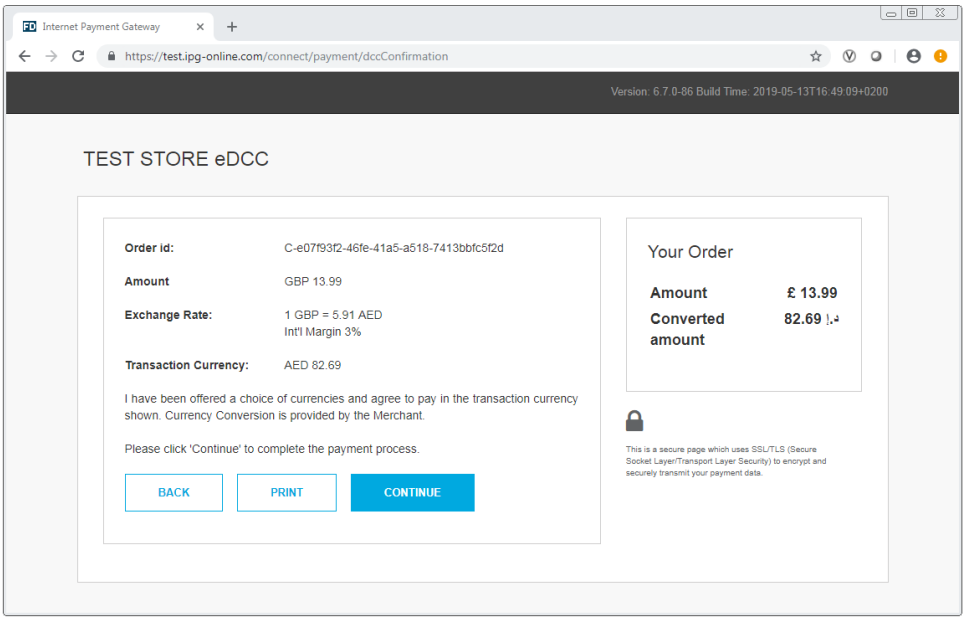
The recurring payments installed using Connect can be modified or cancelled using the Virtual Terminal or Web Service API.

12. Dynamic Currency Conversion (Global Choice™) and Dynamic Pricing

Foreign customers have the choice to pay for goods and services purchased online in their home currency when using their Visa or MasterCard credit card for the payment. The currency conversion is quick and eliminates the need for customers to mentally calculate the estimated cost of the purchase in their home currency. International Visa and MasterCard eCommerce customers can make informed decisions about their online purchases and eradicate any unexpected pricing or foreign exchange conversions on receipt of their monthly statements.

If your Store has been activated for this product option, the Connect solution automatically offers a currency choice to your customers if the card they use has been issued in a country with a currency that is different to your default currency.





Please note that for compliance reasons Clover's Global Choice can only be offered on transactions that take place in full at that time (e.g.: Sale, Refund) and not on any delayed settlement (e.g.: pre/post auth, recurring) due to the fluctuation of the rate of exchange.

Another option for your foreign customers is to display all pricing within your online store in their home currency using our Dynamic Pricing solution. This solution removes the need for your company to set pricing in any other currency other than your home currency.

Please see the Integration Guide for our Web Service API for details on how to request the exchange rates.

If your Store has been activated for this product option and you want to submit the payment transaction via our Connect solution, you need to send the DCC Inquiry ID that you have received along with the exchange rate request in the parameter 'dccInquiryId'.

You can also use the 'dccInquiryId' for cases where Global Choice is being offered and handled on your side (e.g.: within a mobile app). If the cardholder declines the currency conversion offer within your environment, the request parameter 'dccSkipOffer' can be set to 'true' so that the hosted consumer dialogue will automatically be skipped.

13. Purchasing Cards

Purchasing Cards offer businesses the ability to allow their employees to purchase items with a credit card while providing additional information on sales tax, customer code etc. When providing specific details on the payment being made with a Purchasing card favourable addendum interchange rates are applied.

There are three levels of details required for Purchasing Cards:

- Level I - The first level is the standard transaction data; no enhanced data is required at this level.
- Level II - The second level requires that data such as tax amount and customer code be supplied in addition to the standard transaction date. (Visa only have a level II option)
- Level III - The third level allows a merchant to pass a detailed accounting of goods and services purchased to the buyer. All the data for Level I and Level II must also be passed to participate in Level III. (Visa and MasterCard).

You can submit Level II and Level III data in your transaction request using the following parameters:

Field Name	Description, possible values and format
pcCustomerReferenceID	Merchant-defined reference for the customer that will appear on the customer's statement.
pcSupplierInvoiceNumber	Merchant-defined reference for the invoice, e.g. invoice number.
pcSupplierVATRegistrationNumber	The Identification number assigned by the taxing authorities to the merchant.
pcTotalDiscountAmount	The total discount amount applied to a transaction (i.e. total transaction percentage discounts, fixed transaction amount reductions or summarization of line item discounts).
pcTotalDiscountRate	The rate of the discount for the whole transaction.
pcVatShippingRate	The total freight/shipping amount applied to the transaction. Merchants can choose to deliver the contents of a single transaction in multiple shipments and this field reflects the total cost of those deliveries.
pcVatShippingAmount	The total freight/shipping amount applied to the transaction. Merchants can choose to deliver the contents of a single transaction in multiple shipments and this field reflects the total cost of those deliveries.
pcLineItemsJson	Line Item Details in JSON format.

	See table below for more information.

Purchasing Cards Line Item Details in JSON format:

Field Name	Description, possible values and format
CommodityCode	A reference to a commodity code used to classify purchased item.
ProductCode	A reference to a merchant product identifier, the Universal Product Code (UPC) of purchased item.
Description	Represents a description of purchased item.
Quantity	Represents a quantity of purchased items.
UnitOfMeasure	Represents a unit of measure of purchased items.
UnitPrice	Represents mandatory data for Level III transactions.
VATAmountAndRate	Represents a rate of the VAT amount, e.g. 0.09 (means 9%).
DiscountAmountAndRate	Represents a rate of the discount amount, e.g. 0.09 (means 9%).
LineItemTotal	This field is a calculation of the unit cost multiplied by the quantity and less the discount per line item. The calculation is reflected as: [Unit Cost * Quantity] - Discount per Line Item = Line Item Total.

14. Transaction Response

14.1 Response to your Success/Failure URLs

Upon completion, the transaction details will be sent back to the defined 'responseSuccessURL' or 'responseFailURL' as hidden fields. You can define these URLs in your transaction request. Alternatively, you can define them once in the Customisation section of our Virtual Terminal.

Field Name	Description, possible values and format
approval_code	Approval code for the transaction. The first character of this parameter is the most helpful indicator for verification of the transaction result. 'Y' indicates that the transaction has been successful 'N' indicates that the transaction has not been successful "?" indicates that the transaction has been successfully initialized, but a final result is not yet available since the transaction is now in a waiting status. The transaction status will be updated at a later stage.
oid	Order ID
refnumber	Reference number
status	Transaction status, e.g. 'APPROVED', 'DECLINED' (by authorization endpoint or due to fraud prevention settings), 'FAILED' (wrong transaction message content/parameters, etc.) or 'WAITING' (asynchronous Alternative Payment Methods).
txndate_processed	Time of transaction processing
ipgTransactionId	Transaction identifier assigned by the gateway, e.g. to be used for a Void
tdate	Identification for the specific transaction
fail_reason	Reason the transaction failed
response_hash	Hash-Value to protect the communication (see more below)
processor_response_code	The response code provided by the backend system. Please note that response codes can be different depending on the used payment type and backend system. While for credit card payments, the response code '00' is the most common response for an approval, the backend for giropay transactions for example returns the response code '4000' for successful transactions.
fail_rc	Internal processing code for failed transactions
terminal_id	Terminal ID used for transaction processing
ccbin	6 digit identifier of the card issuing bank

cccountry	3 letter alphanumeric ISO code of the cardholder's country (e.g. USA, DEU, ITA, etc.) Filled with "N/A" if the cardholder's country cannot be determined or the payment type is not credit card
ccbrand	Brand of the credit or debit card: MASTERCARD VISA AMEX DINERSCLUB MAESTRO Filled with "N/A" for any payment method which is not a credit card or debit card
schemeTransactionId	Credentials on file (COF) specific parameter. Returned in the response by a scheme for stored credentials transactions to be used in subsequent transaction request for future reference.

For 3D Secure transactions only:

response_code_3dsecure	Return code indicating the classification of the transaction: 1 – Successful authentication (VISA ECI 05, MasterCard ECI 02) 2 – Successful authentication without AVV (VISA ECI 05, MasterCard ECI 02) 3 – Authentication failed / incorrect password (transaction declined) 4 – Authentication attempt (VISA ECI 06, MasterCard ECI 01) 5 – Unable to authenticate / Directory Server not responding (VISA ECI 07) 6 – Unable to authenticate / Access Control Server not responding (VISA ECI 07) 7 – Cardholder not enrolled for 3D Secure (VISA ECI 06) 8 – Invalid 3D Secure values received, most likely by the credit card issuing bank's Access Control Server (ACS) Please see note about blocking ECI 7 transactions in the 3D Secure section of this document.

For Global Choice™ transactions only:

dcc_foreign_amount	Converted amount in cardholder home currency. Decimal number with dot (.) as a decimal separator
dcc_foreign_currency	ISO numeric code of the cardholder home currency. This transaction is performed in this currency String
dcc_margin_rate_percentage	Percent of margin applied to the original amount. Decimal number with dot (.) as a decimal separator
dcc_rate_source	Name of the exchange rate source (e.g. Reuters Wholesale Inter Bank) String
dcc_rate	Exchange rate. Decimal number with dot (.) as a decimal separator.
dcc_rate_source_timestamp	Exchange rate origin time. Integer - Unix timestamp (seconds since 1.1.1970)
dcc_accepted	Indicates if the card holder has accepted the conversion offer (response value 'true') or declined the offer (response value 'false')

For merchants using the Clover Global Merchant Acquiring model only:

associationResponseCode	The raw association value tells exactly how the issuer has responded to the transaction without any mapping done either by the authorization platform or the gateway. It will be returned only for Visa, MasterCard, Amex, and Discover

Additionally, when using your own error page for negative validity checks (full_bypass=true):

fail_reason_details	Comma separated list of missing or invalid variables.

	Note that 'fail_reason_details' will not be supported in case of payplus and fullpay mode
invalid_cardholder_data	**true** – if validation of card holder data was negative **false** – if validation of card holder data was positive but transaction has been declined due to other reasons

In addition, your custom fields and billing/shipping fields will also be sent back to the specific URL.

Please consider when integrating that new response parameters may be added from time to time in relation to product enhancements or new functionality.

14.2 How to generate a hash for a response

Make sure to use the parameter 'response_hash' to recheck if the received transaction response has really been sent by Clover to protect you from fraudulent manipulations. The value is created with a HMAC Hash using the following parameter string:

approval_code|chargetotal|currency|txndatetime|storename

Shared secret ('sharedsecret') will be used as a key in HMAC to calculate the hash with the above hash string. The hash algorithm is the same as the one that you have set in the transaction request.

Please note that you have to implement the response hash validation, when doing so remember to store the 'txndatetime' that you have submitted with the transaction request in order to be able to validate the response hash. Furthermore, you must always use the https-connection (instead of http) to prevent eavesdropping of transaction details.

14.3 Server-to-Server Notification

In addition to the response you receive in hidden fields to your 'responseSuccessURL' or 'responseFailURL', the payment gateway can send server-to-server notifications with the above result parameters to a defined URL. This is especially useful to keep your systems in synch with the status of a transaction. To use this notification method, you can specify an URL in the Customisation section of the Virtual Terminal or submit the URL in the following additional transaction parameter 'transactionNotificationURL'.

Please note that:

- The Transaction URL is sent as received therefore please don't add additional escaping (e.g. using %2f for a Slash (/).
- No SSL handshake, verification of SSL certificates will be done in this process.
- The Notification URL needs to listen on port 443 (https) – other ports are not supported.

The response hash parameter for validation (using the same algorithm that you have set in the transaction request) 'notification_hash' is calculated as follows:

chargetotal|currency|txndatetime|storename|approval_code

Shared secret ('sharedsecret') will be used as a key in HMAC to calculate the hash with the above hash string.

Such notifications can also be set up for the recurring payments that get automatically triggered by the gateway. Please contact your local support team to get a shared secret ('rcpSharedSecret') agreed for these notifications. You can configure your Recurring Transaction Notification URL ('rcpTransactionNotificationURL') in the Customisation section of the Virtual Terminal.

In case of the recurring transactions the response hash parameter 'notification_hash' is calculated differently as follows:

chargetotal+rcpSharedSecret+currency+txndatetime+storename+approval_code

The shared secret ('rcpSharedSecret') is part of the string (it is not used as a key in HMAC to calculate the hash with the hash string). Moreover, the response hash parameter for the recurring transaction notifications is calculated with the SHA256-value (as the default value).

Appendix I – How to generate a hash for a request

If you are using an HTML form to initiate a transaction, your request needs to include a security hash for verification of the message integrity.

The hash (parameter 'hashExtended') needs to be calculated using all non-empty gateway specified request parameters in ascending order of the parameter names, where the shared secret (parameter 'sharedsecret') must be used as the secret key for calculating the hash value. The gateway sorts the request parameters in the "natural order". For strings this means the "Lexicographic Order", thus the upper-case characters come before the lower case (based on ASCII value).

The request parameters that are not specified in our solution can still be submitted in your request to the gateway, but they must be excluded from the hash calculation. They will be ignored during processing and returned in the response.

When you are using Direct Post, there is also an option where you do not need to know the card details (PAN, CVV and Expiry Date) for the hash calculation. This will be managed with a specific setting performed on your store. Please contact your local support team if you want to enable this feature.

Creating the hash with all parameters

Transaction request values used for the hash calculation can be considered as a set of mandatory as well as optional gateway specified request parameters depending on the way you decide to build your request. See an example below:

- chargetotal= 13.00
- checkoutoption = combinedpage
- currency= 978
- hash_algorithm=HMACSHA256
- paymentMethod=M
- responseFailURL=https://localhost:8643/webshop/response_failure.jsp
- responseSuccessURL=https://localhost:8643/webshop/response_success.jsp
- storename=10123456789
- timezone= Europe/Berlin
- transactionNotificationURL=https://localhost:8643/webshop/transactionNotification
- txndatetime= 2021:09:06-16:43:04
- txntype=sale
- sharedsecret=sharedsecret (to be used as the secret key for calculating the hash value)

The steps below provide the guidelines on how to calculate a hash, while using the values from our example.

Step 1. Extended hash needs to be calculated using all non-empty gateway specified request parameters in ascending order of the parameter names, where the upper-case characters come before the lower case (based on ASCII value). Join the parameters' values to one string with pipe separator (use only parameters' values and not the parameters' names).

```
stringToExtendedHash =
13.00|combinedpage|978|HMACSHA256|M|https://localhost:8643/webshop/response
_failure.jsp|https://localhost:8643/webshop/response_success.jsp|1012345678
9|Europe/Berlin|https://localhost:8643/webshop/transactionNotification|2021
:09:06-16:43:04|sale
```

Corresponding hash string does not include 'sharedsecret', which has to be used as the secret key for the HMAC instead.

Step 2. Pass the created string to the HMACSHA256 algorithm and using shared secret as a key for calculating the hash value.

```
HmacSHA256(stringToExtendedHash, sharedsecret)
```

Step 3. Encode the result of HMACSHA256 with Base64 and pass it to the gateway as part of your request.

```
Base64:
EapafBqqOF6N/kch8USkHPGh+fwSko24h6FpQnQHfQ8=

<input type="hidden" name="hashExtended" value="
EapafBqqOF6N/kch8USkHPGh+fwSko24h6FpQnQHfQ8="/>
```

Appendix II – ipg-util.asp

```
<!-- google CryptoJS for HMAC -->
<script LANGUAGE=JScript RUNAT=Server src="script/cryptoJS/crypto-
js.min.js"></script>
<script LANGUAGE=JScript RUNAT=Server src="script/cryptoJS/enc-
base64.min.js"></script>
<script LANGUAGE=JScript RUNAT=Server>
    var today = new Date();
    var txndatetime = today.formatDate("Y:m:d-H:i:s");

    /*
        Function that calculates the hash of the following parameters as an
example:
- chargetotal
- checkoutoption
- currency
- hash_algorithm
- paymentMethod
- responseFailURL
- responseSuccessURL
- storename
- timezone
- transactionNotificationURL
- txndatetime
- txntype
- and sharedsecret as the secret key for calculating the hash value
     */

    function createExtendedHash(chargetotal, currency) {
        // Please change the storename to your individual Store Name
        var storename = "10123456789";
        // NOTE: Please DO NOT hardcode the secret in that script. For example read
it from a database.
        var stringToExtendedHash =
chargetotal|checkoutoption|currency|hash_algorithm|paymentMethod|responseFailURL|re
sponseSuccessURL|storename|timezone|transactionNotificationURL|txndatetime|txntype;

        var hashHMACSHA256 = CryptoJS.HmacSHA256(stringToExtendedHash,
sharedSecret);
        var extendedhash = CryptoJS.enc.Base64.stringify(hashHMACSHA256);

        Response.Write(extendedhash);
    }

    function getDateTime() {
        Response.Write(txndatetime);
    }
</script>
```

Appendix III – ipg-util.php

```
<!DOCTYPE HTML>
<html>
<head><title>IPG Connect Sample for PHP</title></head>
<body>
<p><h1>Order Form</h1>

<form method="post" action="https://test.ipg-
online.com/connect/gateway/processing">


<fieldset>
        <legend>IPG Connect Request Details</legend>
        <p>
            <label for="storename">Store ID:</label>
            <input type="text" name="storename" value="10123456789"
readonly="readonly" />
        </p>
        <p>
            <label for="timezone">Timezone:</label>
            <input type="text" name="timezone"
value="Europe/London"  readonly="readonly"/>
        </p>
        <p>
            <label for="chargetotal">Transaction Type:</label>
            <input type="text" name="txntype" value="sale" readonly="readonly" />
        </p>
        <p>
            <label for="chargetotal">Transaction Amount:</label>
            <input type="text" name="chargetotal" value="13.00" readonly="readonly"
/>
        </p>
        <p>
            <label for="currency">Currency (see ISO4217):</label>
            <input type="text" name="currency" value="978" readonly="readonly" />
        </p>
        <p>
            <label for="txndatetime">Transaction DateTime:</label>
            <input type="text" name="txndatetime" value="<?php echo getDateTime();
?>"/>
        </p>
        <p>
            <label for="hashExtended">Hash Extended:</label>
               <input type="text" name="hashExtended" value="<?php echo
createExtendedHash('13.00', '978'); ?>"  readonly="readonly" />
        </p>
        <p>
            <label for="hashExtended">Hash Algorithm :</label>
            <input type="text" name="hash_algorithm" value="HMACSHA256"
readonly="readonly" />
        </p>
        <p>
            <label for="hashExtended">Checkout option :</label>
            <input type="text" name="checkoutoption" value="combinedpage"
readonly="readonly" />
        </p>
        <p>
            <input type="submit" id="submit" value="Submit" />
        </p>
    </fieldset>

</form>

<?php
```

```
function getDateTime() {
    return date("Y:m:d-H:i:s");
}

function createExtendedHash($chargetotal, $currency) {
// Please change the store Id to your individual Store ID
// NOTE: Please DO NOT hardcode the secret in that script. For example read it from
a database.
$sharedSecret = "sharedsecret";
$separator = "|";
$storeId= "10123456789";
$timezone= "Europe/London";
$txntype= "sale";
$checkoutoption = "combinedpage";

$stringToHash = $chargetotal . $separator . $checkoutoption . $separator .
$currency . $separator . "HMACSHA256" . $separator . $storeId . $separator .
$timezone. $separator . date("Y:m:d-H:i:s") . $separator . $txntype;

$hash = base64_encode(hash_hmac('sha256', $stringToHash, $sharedSecret, true));
return $hash;
}

?>
</body>
</html>
```

The above is the working PHP example, to run it you can copy the above and paste it on https://www.w3schools.com/php/phptryit.asp?filename=tryphp_function1

Appendix IV – PayPal

Refer to the following information when integrating PayPal as a payment method.

Transaction types mapping

Connect Transaction Type (txntype)	PayPal operation
Sale	SetExpressCheckoutPayment (sets *PaymentAction* to *Authorization* in *SetExpressCheckout* and *DoExpressCheckoutPayment* requests)
Preauth	GetExpressCheckoutDetails
sale – with additional parameters for installing a Recurring Payment	DoExpressCheckoutPayment*
Postauth	DoCapture (,DoReauthorization)
Void	DoVoid

Address handling

If you pass a complete set of address values within your request to Connect (name, address1, zip, city and country within billing and/or shipping address), these values will be forwarded to PayPal, setting the PayPal parameter 'addressOverride' to '1'.

Please note that it is an eligibility requirement for PayPal's Seller Protection that the shipping address will be submitted to PayPal.

If you submit no or incomplete address data within the Connect request, no address data will be forwarded to PayPal and the PayPal parameter 'addressOverride' will not be set.

Regardless of that logic, the payment gateway will always store the shipTo address fields received from PayPal in the GetDetails request in the ShippingAddress fields, possibly overwriting values passed in the request to Connect (such overwriting depends on the above logic).

* If you want to use PayPal's Reference Transactions feature for recurring payments, please contact PayPal upfront to verify if your PayPal account meets their requirements for this feature.

Recurring Payment Transaction

You have to submit a SALE transaction request with the corresponding parameters to install the recurring payments. The first transaction is always conducted immediately along with the request.

The subsequent transactions are executed by the Gateway's scheduler, via the API Web Service, as defined during the initial SALE transaction with the installation.