Gateway Freedom Integration Guide

Registered in England No: 2081330. VAT Reg. No: 618184140
Pay360 by Capita Gateway Freedom v 3.0
23rd March 2016

Table of Contents

1 Introduction

The Freedom product allows you to use the customisable hosted payment pages, the API, or both. Some merchants like to take the initial payment on the hosted payment pages, and then use the API for tokenised repeats and refunds, as there are fewer PCI compliance implications.

PLEASE SKIP TO **SECTION 13** IF YOU WILL PURELY BE USING THE API, AND NOT REDIRECTING TO THE PAY360 BY CAPITA HOSTED PAGE.

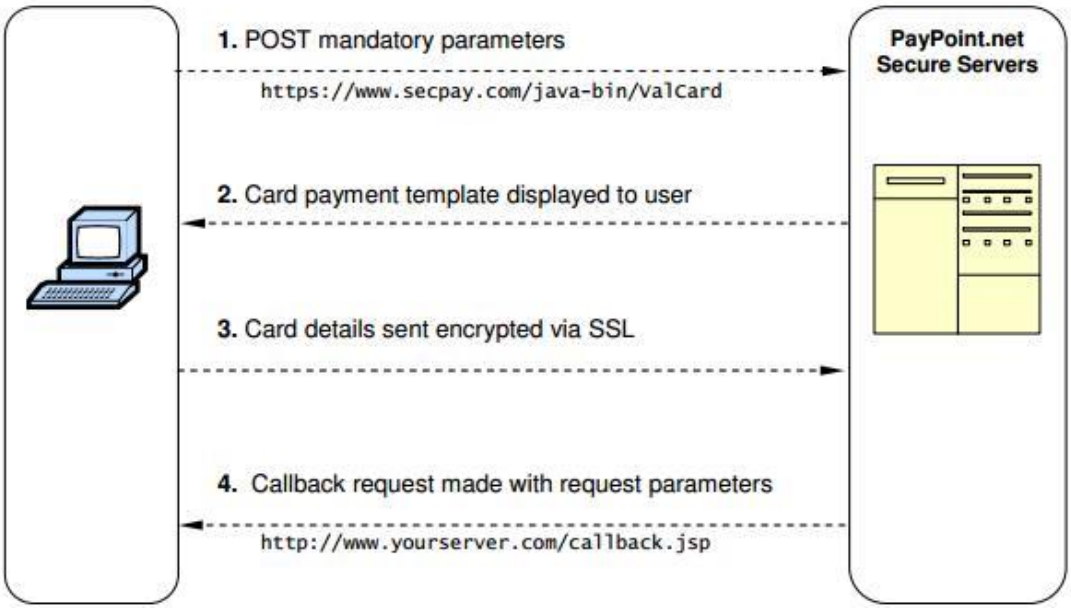
2 Basic setup

Before going in depth about the usage of the API, it's worth explaining the basic functionality and setup process between your servers and the Pay360 by Capita servers.
If you have a Freedom account purely to customise the template, then you will need the following information to enable the redirect between your servers and ours.

The integration works using forms submitted over SSL. A secure server at your end is not necessary, but can be used if you wish. The lifecycle of a typical transaction consists of four steps:

- A POST request containing Mandatory Parameters is made from your server to the Pay360 by Capita server.
- A card payment page, hosted on the Pay360 by Capita server is displayed to the cardholder.
- Card details are sent to Pay360 by Capita
- A GET or POST call-back request is sent back to your server from Pay360 by Capita

Figure.1: Basic Concept



3 POST the Mandatory Parameters

The lifecycle of a typical transaction starts with an HTML form hosted on your server making a POST request to the Pay360 by Capita servers. The action attribute of this form should be https://www.secpay.com/java-bin/ValCard.
An example of the form tag would be:

<form name="myform" method="POST" action=https://www.paypoint.net/java-bin/ValCard>

This form must send a minimum set of parameters. These parameters are known as the "Mandatory Parameters" and are most commonly implemented as hidden input fields.

- merchant
- trans_id
- amount
- callback
- digest

3.1 merchant

This is your Pay360 by Capita username (usually six letters followed by two numbers)

Example

abcdef01

3.2 trans_id

A unique transaction identifier created by yourself. This can be used to refer to a transaction at a later date (to refund it for example). The only constraint placed upon the format of this ID is that you omit any spaces. Should you require spaces, please use a hyphen or underscore ('-'or '_').

Example

transaction01, transaction_01 or transaction-01

3.3 amount

This is the amount for the transaction. This should contain no currency symbols or formatting (for example, do not send an amount with a comma in). A decimal point may be used, as in the example below, but it is not mandatory: "50.00" is equal to "50".

Example

50.00

3.4 callback

This is the location on your server that Pay360 by Capita is required to call when we have completed the authorisation process (see section 4. Callback).

Example

http://www.example.com/callback.jsp

3.5 digest

In order for Pay360 by Capita to be able to ensure that a request to process a transaction actually came from your web application, you need to authenticate yourselves to us. This is done by POSTING the digest parameter to Pay360 by Capita (along with the other mandatory parameters. Please see section 11: Security for more details.

Example

<input type="hidden" name="digest" value="7cbe0b4606943c6a76b38ebef74c237">

4 The Callback

After payment has occurred, we need to send the cardholder back to your server and also need to inform them (and you) of whether or not the transaction was authorised by the bank. This is done using the callback mechanism.
These are the features of a typical callback:

- A GET request is made to your callback page.
- This request contains the callback request parameters
- The HTML output from your callback page is read by the Pay360 by Capita server and displayed to the cardholder's browser.

The resulting HTML output from the callback to your server is presented relative to the Pay360 by Capita server, and therefore you should not include any relative path graphics or links, as these will not work. Images and links which are in your callback page should be referenced absolutely.

Example

A relative link excludes the main url; any links within the code will automatically reference the root url.
(e.g. …/images/logo.jpg)

An absolute link includes the entire path.
(e.g. http://www.example.com/images/logo.jpg)

Note: Your callback page must actually provide some HTML response when called by our server. This response must include both an <html> start and end tag, and a <body> start and end tag.

4.1 Referencing your Callback Page

The obvious question is; how does Pay360 by Capita know where your callback page is? You specify the location of your callback when you make the initial POST request to our servers. This is through the use of the mandatory parameter callback which is mentioned in section 3. POST the Mandatory Parameters.

Example

<input type="hidden" name="callback" value =http://www.example.com/callback.jsp>

4.2 Using Two Static Callback Pages

It is possible to specify two callback pages when sending a transaction to Pay360 by Capita; one for display in the event of an authorised transaction, and one for display for a declined transaction. This is useful if you do not run server side code such as PHP or Java on your web server.
In order to specify two callback locations, you implement a hidden form field as follows:

<input type="hidden" name="callback"
value="http://www.example.com/authorised.html;http://www.example.com/declined.html>

The page to be displayed for an authorised transaction should be before the page to be displayed for a declined transaction.

NOTE: the authorised and declined URLs should be separated by a semi-colon.

4.3 Using One, Dynamic, Callback Page

When a callback occurs, Pay360 by Capita send request parameters known as the callback parameters. If you specify a single dynamic callback page (such as a PHP or JSP file) your page can read these request parameters and determine whether or not the transaction was authorised.

Example

<input type="hidden" name="callback" value=http://www.example.com/callback.jsp>

NOTE:

By default, Pay360 by Capita makes a GET request to your callback page. You can configure your transaction to have a POST request callback by including the optional parameter "cb_post=true" in the options field. POST requests do not include a query string.

4.4 SSL Callbacks

In the event that you do have a secure server set up, and therefore your callback page is running over HTTPS, you will need to specify this by sending the (optional) ssl_cb=true parameter.

Example

<input type="hidden" name="options" value="ssl_cb=true">

or

<input type="hidden" name="ssl_cb" value="true">

…can be used with a callback of https://www.example.com/callback.jsp

4.5 Callback Parameters

The request parameters sent back to your server with the callback are as follows.

- trans_id
- valid
- auth_code
- cv2avs
- message
- esp_code
- amount
- code
- test_status
- hash

NOTE: The following three callback parameters are only supplied if you sent "repeat=true" (allowing repeat transactions) in the options field:

- expiry
- card_no
- customer

NOTE: The following callback parameter is only supplied when a currency other than the default (GBP) is used:

- currency

NOTE: The following callback parameter is only supplied if you sent "cv_card_type=true" in the options field:

- card_type

4.5.1 trans_id

We send you back the same ID that you sent us, with your mandatory parameters so that you can update your system appropriately.

4.5.2 valid

This is 'true' or 'false' and indicates the acceptance or not of the card details.

4.5.3 auth_code

This is the authorisation code obtained from the bank for this transaction. This is only returned if valid=true. For a transaction sent with the optional parameter 'test_status=true', the auth_code with always be 9999.

4.5.4 cv2avs

The Apacs approved text that is supplied as a result of the CV2 and AVS anti-fraud checks. There are five core values defined which are:

ALL MATCH	All the data provided matched that which the card issuer had on record.
SECURITY CODE MATCH ONLY	Only the security code (CV2) matched.
ADDRESS MATCH ONLY	Only the address matched.
NO DATA MATCHES	None of the data matched
DATA NOT CHECKED	The cv2avs system is unavailable or not supported by this card issuer.

With these core codes, an address is only understood to match only if both the address and the postcode match at the same time. This is a little strict for some people, as the banks have a very defined address layout to adhere to, so the following codes have also been introduced.

PARTIAL ADDRESS MATCH / POSTCODE	The postcode matched, but the address did not (following on from above, this does not mean the address is wrong, it could just be that it does not match exactly the one on record at the bank)
PARTIAL ADDRESS MATCH / ADDRESS	The address matched, but the postcode did not.
SECURITY CODE MATCHED / POSTCODE	The security code and postcodes matched, but the address did not.
SECURITY CODE MATCH / ADDRESS	The security code and address matched, but the postcode did not.

Codes are only supplied when CV2 and/or Billing Address data is supplied. It is in your interests to supply this data to us.

4.5.5 message

This parameter is only returned when a transaction is declined and code=N. It is a failure message sent from the bank and should not be displayed to the cardholder.

Example

DECLINED

4.5.6 resp_code

This parameter is only returned when a transaction is declined and code=N. This is the bank's failure code for your information only, do not show it to the customer.

- 2 or 83 Referral It may be possible to obtain an authorisation code from the bank.
- 5 or 54 Not Authorised
- 30 General Error (retrying after 1 minute may succeed, depending on the error)

4.5.7 amount

This is the amount actually authorised by the bank.

4.5.8 code

The code field is a short code giving extensive details of failure states. This is the parameter that should be used to programmatically determine whether or not a particular transaction was authorised. e.g.

- Code=A Authorised
- Code=N Not Authorised

Below is a list of all possible response codes, along with their meanings.

Code	Definition
A	Transaction authorised by bank. 'auth_code' available as bank reference
N	Transaction not authorised. Failure text available to merchant (see 4.5.6)
C	Communication problem. Trying again later may work.
F	The Pay360 by Capita system has detected a fraud condition and rejected the transaction. The message field will contain more details.
P:A	Pre-bank checks. Amount not supplied or invalid

P:X	Pre-bank checks. Not all mandatory parameters supplied.
P:P	Pre-bank checks. Same payment presented twice.
P:S	Pre-bank checks. Start date invalid.
P:E	Pre-bank checks. Expiry Date Invalid.
P:I	Pre-bank checks. Issue number invalid.
P:C	Pre-bank checks. Card number fails LUHN check. (The card number is wrong)
P:T	Pre-bank checks. Card type invalid. i.e. does not match the card number prefix.
P:N	Pre-bank checks. Customer name not supplied.
P:M	Pre-bank checks. Merchant does not exist or not registered yet.
P:B	Pre-bank checks. Merchant account for card type does not exist.
P:D	Pre-bank checks. Merchant account for this currency does not exist.
P:V	Pre-bank checks. CV2 security code mandatory and not supplied/invalid.
P:R	Pre-bank checks. Transaction timed out awaiting a virtual circuit. Merchant may not have enough virtual circuits for the volume of business.
P:#	Pre-bank checks. No MD5 hash/token key set up against account

NOTE:

Pre-auth checks can have several errors, e.g. P:NEC means the name, expiry date and card number fields are all invalid or not supplied.

4.5.9 test_status

This is returned if the options parameter test_status was set to either ‘true’ or ‘false’.

4.5.10 hash

This parameter is always returned on the callback and can be used to ensure that a callback was not spoofed and actually came from Pay360 by Capita (Please see section 14: Security for more information).

Example

1e12858a757ef15b03da3beb5c59f0af

4.5.11 expiry

This is the expiry data of the credit card used for payment as entered by the cardholder. This parameter is only returned in the callback if the “repeat=true” parameter was sent with the transaction.

4.5.12 card_no

This is the last four digits of the card number used for this transaction. This parameter is only returned in the callback if the ‘repeat=true’ parameter was sent with the transaction.

4.5.13 customer

This is the name of the cardholder as entered by the cardholder. This parameter is only returned in the callback if the “repeat=true” parameter was sent with the transaction.

4.5.14 currency

This is the currency that the transaction was processed in. This parameter is only returned in the callback when a currency other than the default (GBP) is used.

4.5.15 card_type

This is useful for customers that charge for credit card transaction and charge differently by type. This parameter is only returned in the callback if the “cb_card_type=true” parameter was sent with the transaction.
Potential values for the card_type callback parameter are:

- American Express
- Delta
- Diners Card
- JCB

- Master Card Credit
- Debit Master Card
- Solo
- Maestro
- Visa
- Laser

NOTE:

Historically you would have received MasterCard as a card type, however since the release of Debit Mastercards, we have split the card type into two separate values as above.

4.6 Mandating CV2

To mandate the CV2 code you can supply req_cv2=true in the options parameter of your form POST. Alternatively you can set the CV2-AVS options within the Extranet under Account then Account Options. To make the CV2 mandatory via the Extranet you must set it to Strict.

4.7 Supply your own callback parameters

You can also supply your own parameters with the callback and then use them when the script is called. Anything can be supplied. If you do supply your own parameters than you must ensure they are URL encoded.

NOTE:

You will only receive these parameters back in the query string If your callback takes the default form; GET. If you use the cb_post parameter to make your callback a POST, you will not receive a query string.

Example

"callback=http://www.example.com/cgi-bin/callback.pl?bruce=one&wayne=two+three"

Note the '+'

The 'bruce' and 'wayne' parameters will be returned to you as given. This mechanism could be used to get back your own order and address parameters if you don't want to save them on your site and then look them up.

Our server side extensions can also be used to expand variables originally supplied to us in any form variable (except 'card_no'!) so for example, the addresses or order details can be retrieved via the callback.

Example

http://www.example.com/callback?order=${order}&{shipping=${shipping}&billing={billing}

The above will put the appropriate information onto the callback.

4.8 Customer redirection: -redir and -jredir

As standard, the Pay360 by Capita hosted page will GET the information specified in your callback page and display it on the same url (https://www.secpay.com/java-bin/ValCard).
However, there may be instances where you would rather the customer was redirected back to your site. This can be achieved in two ways, either by a server-side redirect or by a client-side redirect.

Server-side redirect can be achieved by adding -redir to the merchant id in the merchant field.

Example

<input type="hidden" name="merchant" value="testin12-redir">

For client-side redirect -jredir needs to be added to the merchant id in the merchant field.

Example

<input type="hidden" name="merchant" value="testin12-jredir">

5 Multi-Currency Transaction

Using Pay360 by Capita it is possible to authorise transactions in any currency. If you wish to authorise transactions in a foreign currency, you simply obtain permission from your bank and provide to Pay360 by Capita a Merchant Account number for each of the currencies you wish to use (please send this number to support@paypoint.net).

By default, all transactions sent to Pay360 by Capita are assumed to be GBP, but you can specify the currency of any transaction at run-time on a per-transaction basis using the currency optional parameter.

Example

<input type="hidden" name="currency" value="USD">

Below are the currency codes for some of the most common currencies in use. For a full list of ISO currency codes, please visit http://www.xe.com/iso4217.htm

- AUD Australia, Dollars.
- CAD Canada, Dollars
- EUR Euro Member Countries, Euros
- GBP United Kingdom, Pound Sterling.
- HKD Hong Kong, Dollars
- JPY Japan, Yen.
- USD United States of America, Dollars

NOTE:

These currency codes, like all other parameters that Pay360 by Capita use, are case sensitive.

NOTE:

Test transactions will always return a value of 'GBP' even if an alternate value is provided.

Pay360 by Capita **does not** perform any form of currency conversion. If you would like to accept transactions through your website in USD, but would like the funds from these transactions to be settled into your bank account in GBP, the currency conversion necessary for this is performed automatically by your bank (the bank with whom you have your Internet Merchant Account (IMA) with).

6 CV2 / AVS: Security Code and Address Checks

Pay360 by Capita provides as standard, transmission for AVS (address & postcode verification) and CV2 (last 3 digits on the reverse of the card) for ALL acquiring banks. These checks are not actually performed by Pay360 by Capita, but by the credit/debit card issuers and are therefore dependent upon whether or not the issuer for the particular credit card being used, supports it. The results of these CV2 and AVS checks are sent back to you in the cv2avs callback parameter (see section 5.5.4: cv2avs)

6.1 CV2

The cv2 number, or 'security code' is found on the back of a credit card and is the last three digits displayed on the signature strip. AMEX cards are an exception to this rule as their cv2 number is found on the front of the card and is four digits long.

The cv2 number is mandatory by default and bears no relationship to the card number: therefore it cannot be 'calculated'. Due to this, it can be asserted that if the cv2 number provided matches that which the card issuer has on record, then the person who provided this information has physically seen the credit card, or a copy or the credit card.

6.2 AVS: Address Verification System

Like the CV2 check, the AVS check is done by the credit card issuer. The results of this check are sent back to Pay360 by Capita and Pay360 by Capita send them back to your website as part of the cv2avs callback parameter.
Billing address details may be captured on your server and then sent to Pay360 by Capita along with the Mandatory Parameters, the fields are available on the payment page for the customer to use. There are three different ways in which Billing Address parameters can be sent:

6.2.1 As individual parameters:

- bill_name Contact or Customer's name
- bill_company Company (if sold to a company)
- bill_addr_1 Generic address lines
- bill_addr_2 Generic address lines
- bill_city The city or town
- bill_state The country, state, province or other area
- bill_country The country
- bill_post_code The country post code or zip
- bill_tel The contact telephone number
- bill_fax The contact fax number
- bill_email Email address. This is used for order confirmation and should always be supplied.
- bill_url URL of website.

Individual Billing Address parameters can be implemented as separate request parameters.

Example

```
<input type="hidden" name="bill_city" value="London">
<input type="hidden" name="bill_country" value="England">
```

6.2.2 As one parameter in string format

- All billing address details can be combined into one individual parameter called billing.

Example

```
<input type="hidden" name="billing" value="name=Fred Bloggs,company=Pay360 by
Capita,county=kent,country=England">
```

As you can see from the above example, the 'bill' prefix has been removed from each of the individual parameters specified in the value of the hidden field input field. Also note that each individual item specified in the comma-separated string of billing parameters must be HTML encoded.

6.2.3 As One Parameter in XML Format

Billing address parameters can also be sent as an XML 'snippet'.

Example

```
<billing class='com.secpay.seccard.Address'>
        <name>Fred Bloggs</name>
        <company>Pay360 by Capita</company>
        <addr_1>23 The Road</addr_1>
        <addr_2>The Crescent</addr_2>
        <city>Metropolis</city>
        <state>Happiness</state>
        <country>Atlantis</country>
        <post_code>AA23 BB1</post_code>
        <tel>01234 567890</tel>
        <email>sales@Pay360 by Capita</email>
        <url>http://www.Pay360 by Capita</url>
```

</billing>
Again, note that the text contained within each XML element must be HTML encoded. Once constructed, this XML string can be sent within a hidden input field.

Please note as HTML and XML have the same quoting conventions, you must ensure that in these fields, they use the opposite to each other.

Example

<input type="hidden" name="billing" value='XML GOES HERE using double quotes'>

or

<input type="hidden" name="billing" value="XML GOES HERE using single quotes">

6.2.4 Sending Shipping Address Details

Shipping address fields have exactly the same naming convention as Billing Address fields, but they are prefixed with "ship_" instead of "bill_". See 11.2.1 for the possible values.

6.3 Configuring Automatic CV2/AVS Transaction screening

Within the extranet you can configure your account to automatically fail transactions depending on the results of the CV2/AVS checks done by the bank. This can be done under "Account" (on the top menu), then "Account options" (on the side), then under 'General Settings'.

6.3.1 CV2

Defines whether or not a CV2 code is required

6.3.2 House no.

Defines whether or not the house number is required

6.3.3 Postcode

Defines whether or not the postcode is required.

6.3.4 Strict/Non-Strict

Defines whether to adhere exactly to the bank's specifications or to allow a small amount of flexibility. (see 4.5.4)

If you are unable to see this option, please contact support@paypoint.net,

7 Sending Order Details

It is possible to send product or service information to Pay360 by Capita with any given transaction. This information is stored on the Pay360 by Capita servers for your convenience and is available for reference through reports at a later date.

Please do not fill this field with very large blocks of text as there are limitations in any piece of software as to field size. Do structure it in the way documented below. Do not include HTML or JavaScript within it. Failure to comply with this can temporarily affect our service and inconvenience other customers.

This field is provided as a convenience for the majority of customers that have simple requirements. It isn't suitable for everyone. If you have more complex requirements, please capture, store and present your own order details to the customer and do not send them to us.

There are two methods of sending order details parameters to Pay360 by Capita

7.1 Order Details in String Format

The order field has the following internal structure. It consists of a set of product identifier and amount pairs with a multiplier attached to the amount as "prod.amountxquantity". The string requires some means of being delimited. Delimited

MUST NOT appear in the actual data (e.g. in the product code) or your string will not be parsed correctly and you will lose information.
There are four delimiters: ';' (semi-colon), ',' (comma), '=' and 'x'.

Example

<input type="hidden" name="order" value="prod=funny_book,item_amount=18.50;prod=sad_book,item_amount=16.50">

or

<input type="hidden" name="order"
value="prod=funny_book,item_amount=18.50x2;prod=sad_book,item_amount=16.50x3">

Notice that in the second example we have specified a quantity for each of the products.

An optional parameter 'delimit' is available at the start of the order string, and can be used to change the default delimiters (this can be useful if all your products contain one of our delimiters in their product codes!).

Example

prod=funny_book,item_amount=18.50x1 ;prod=sad_book,item_amount=16.50x2
…could be fully specified as:

delimit=;,=x;prod=funny_book,item_amount=18.50x1;prod=sad_book;item_amount=16.50x2

…and you could change these delimiters like this:

delimit=#+:*;prod:funny,;=book+item_amount:18.50*1#prod:sad,book+item_amount:16.50*2

…giving silly characters in the product description. It is still your responsibility to ensure the resulting string is well formed. The only way to be sure is to use XML.

There are three keywords (SHIPPING, DISCOUNT, TAX) which can be used within the order details string to specify that a particular order item refers to shipping, discount or tax information. Tax can be specified as TAX or for Example TAX@20%

Example

prod=funny_book,item_amount=18.50x1;prod=sad_book,item_amount=16.50x2;prod=SHIPPING,item_amount=5.00;prod=TAX@20%,amount=6.00

prod=funny_book,item_amount=18.50x1;prod=sad_book,item_amount=16.50x2;prod=SHIPPING,item_amount=5.00;prod=TAX@20%,item_amount=6.00;prod=DISCOUNT,item_amount=5.00

7.2 Order Details in XML Format

Order details can also be sent as an XML 'snippet'.

Example

```
<order class='com.secpay.seccard.Order'>
        <orderLines class='com.secpay.seccard.OrderLine>
                <OrderLine>
                <prod_code>funny_book</prod_code>
                <item_amount>18.50</item_amount>
                <quantity>1</quantity>
                </OrderLine>
                <OrderLine>
                <prod_code>sad_book</prod_code>
                <item_amount>10.00</item_amount>
                <quantity>5</quantity>
                </OrderLine>
                </OrderLines>
        </order>
```

Once constructed, this XML string can be sent within a hidden input field.

Example

<input type="hidden" name="order" value="XML GOES HERE using double quotes'>

or

<input type="hidden" name="order" value="XML GOES HERE using single quotes'>

8 Sending Custom Parameters

You can create your own custom callback parameters that you would like sent through to your callback page after a transaction has either been authorised or declined.

Example

If you included 3 custom fields in the form hosted on your server that POST's to http://www.secpay.com/java-bin/ValCard...

<input type="hidden" name="customfld1" value="customval1">
<input type="hidden" name="customfld2" value="customval2">
<input type="hidden" name="customfld3" value="customval3">

…and you wanted these fields and their values to be made available to your callback page, you would use the cb_flds optional parameter as follow:

<input type="hidden" name="options" value="cb_flds=customfld1:customfld2:customfld3">

or

<input type="hidden" name="cb_flds" value="customfld1:customfld2:customfld3">

9 Optional Parameters

There is also a range of other optional parameters which can be sent that haven't yet been mentioned. All options can be sent either in a comma separated string, or as their own individual parameters. The following Example shows three options parameters implemented as a comma separated string:

<input type="hidden" name="options" value="test_status=true,currency=USD">

…alternatively these parameters could have been implemented as individual fields:

<input type="hidden" name="test_status" value="true">
<input type="hidden" name="currency" value="USD>

A full list of all optional parameters follows.

9.1 backcallback

An extra callback that, if supplied, will give the customer the option of abandoning the card entry dialogue and returning to the merchant's site. You must enter the URL that will be loaded when the back button is clicked into the backcallback field.

You must also supply an additional show_back parameter when you use backcallback, else the back button on the payment form will act as a submit button. Both parameters are shown below:

Example

<input type="hidden" name="backcallback" value=http://www.example.com/backcallback.jsp>

<input type="hidden" name="show_back" value="submit">

Note: You must use both the above: not one or the other.

9.2 amex

This parameter is for use when testing and is used to indicate that you have an extra American Express merchant number and can accept these cards. Only applicable when testing, you must actually have an account when live. (Please contact both Amex and your acquiring bank to get this set up)
Example

amex=true

9.3 auth_code

If you receive a response code of 2 or 83, or a message saying 'REFERRAL', it may be possible to call the bank to obtain an authorisation code over the phone. If there is a telephone number after the 'REFERRAL' message, you call that number. If there is no number, you call your own merchant bank's authorisation number. Once you have an authorisation code, you will need to process the payment via the manual payment form within the extranet by entering the following in the 'Options (Advanced)' box:

auth_code=<*auth_code*>,password=<*remote_password*>

<*auth_code*> is the authorisation code you got from the bank, and <*remote_password*> is the remote password on your account.

Your remote Password can be configured from within the merchant extranet at http:s//www.paypoint.net/secnet/app. Click 'Account' on the top menu, then 'Remote passwords' on the left, then select 'Remote' from the dropdown list. Enter the password, then click 'Save'.

9.4 map_flds

This option allows you to map existing field names to their Pay360 by Capita equivalents, and is especially useful if you want to integrate an existing system without having to make too many changes to it. It is also useful for Shopping Cart vendors who wish to provide a Pay360 by Capita module for integration.
To illustrate this, the example below will make the following changes:

- Change the name of the 'amount' field to be 'money'.
- Change the name of the 'auth_code' field to be 'bank-auth'.

Example

<input type="hidden" name="map_flds" value="amount= money:auth_code=bath-auth">

or

<input type="hidden" name="options" value="maps_flds-amount= money:authcode=bank-auth">

Each of the changes to the field names you would like to make is separated by a colon. You can make your mapped field names appear in the callback to your server by including their names in the cb_flds options parameter:

<input type="hidden" name="cb_flds" value="monkey:bank-auth">

9.5 usage_type

The usage_type parameter is used to advise us what type of transaction you are processing. If you don't supply a usage_type, then the default setting is 'Electronic Commerce'. There are three possible values:

- usage_type=E eCommerce (Electronic Commerce) (***Default***)
- usage_type=M MOTO (Mail Order/Telephone Order)
- usage_type=R Recurring payments

The usage_type parameters can be sent in its own hidden field, or within the options parameters.

Example

<input type="hidden" name="usage_type" value="M">

or

<input type="hidden" name="options" value="usage_type=M">

9.6 allowed_ct

This option allows you to specify a list of card types that will be the only card types accepted. Note that this relies on the system being able to identify the card from its prefix. This is not an exact science because there are overlaps.

NOTE:

This will not alter the drop down list of card types on your template.

Example

<input type="hidden" name="allowed_ct" value="Maestro,Visa">

9.7 banned_ct

This option allowed you to specify a list of card types that will not be accepted by the system,. Note that this relies on the system being able to identify the card type from its prefix. This is not an exact science because there are overlaps.

NOTE: This will not alter the drop down list of card types on your template.

Example

<input type=hidden" Name="banned_ct" value=Maestro,Visa">

10 Emails

Pay360 by Capita will, by default, send the merchant and the customer an order notification email from trans.admin@paypoint.net.

NOTE: The only definitive order confirmation is via the reports on our website. Exclusive use of email to support business processes leaves your system open to misuse, as emails are easily spoofed or mislaid. It is not Pay360 by Capita's policy, not is there any standard procedure to resend emails which have failed to be collected for any reason. Emails are a courtesy function and should be treated as 'useful but not vital'.

Figure 3: Example email sent to the customer

```
Subject: SEC-Order : The Book Shop

Date                Name                    Amount Curr Type         Status
------------------- -------------------- ---------- ---- ------------ ------
15.08.2008 12:02:18 Fred Bloggs              10.00 GBP  Sale         Okay

Order : ABCD1234

Product                          Item Amnt Qty     Amount
-------------------------------- --------- --- ----------
funny_book                           18.50   1      18.50
sad_book                             16.50   2      33.00
Discount                                             5.00
Shipping                                             5.00
Tax                                                  7.00
-------------------------------- --------- --- ----------
                                                    10.00

Addresses
                    Shipping                       Billing
                    ------------------------------ ------------------------------
Name                : Fred Bloggs                  Fred Bloggs
Company             : Paypoint.Net                 Paypoint.Net
Address             : 1 The Road                   1 The Road
                      The Street                   The Street
Town/City           : Metropolis                   Metropolis
County/State        : Sunny State                  Sunny State
Country             : Atlantis                     Atlantis
Post Code           : AA12 3BB                     AA12 3BB
Telephone           : 01234 567891                 01234 567891
Fax                 : 01234 567892                 01234 567892
Email               :                              someone@somewhere.com
```

The email we send to merchants has more information about the customer at the bottom.

The email we send to merchants has more information about the customer at the bottom.

Figure 4: Extra Details sent in the email to the merchant.

```
Last 4 digits on card : 1111

IP Address : 12.34.56.789

CVV2 / AVS Message : ALL MATCH
```

You can configure your email settings as follows:

10.1 mail_attach_customer

Used to specify whether order confirmation details sent to the cardholder should be emailed inline or as an attachment.

Example

- mail_attach_customer=true (default – this means send it as an attachment)
- mail_attach_customer=false (this means send it inline)

10.2 mail_attach_merchant

Same as mail_attach_customer but relates to order confirmation emails sent to the Merchant.

10.3 mail_html

This option is used to specify whether to send the email as html. There are three possible values:

- true
- false
- full

This is 'false' by default, as it's not supported by all email readers. The 'true' setting uses the <PRE> tag in html with minimal html used, so it doesn't look too bad, even in a non-html mail reader, and will line up columns correctly when html does work; so it's a reasonable compromise.

The 'full' setting gives a complete html email, but will show all the tags in old email clients.

10.4 mail_customer

This option is used to specify whether or not to send an order confirmation email to the cardholder. It can also be used to specify that either the Billing Address email or Shipping Address email should be sent an order confirmation email.

Example

- mail_customer=true (default: send an email to the address specified by bill_email)
- mail_customer=false (do not send an email to the cardholder).
- mail_customer=bill (send an email to the address specified by bill_email).
- mail_customer=ship (send an email to the address specified by ship_email).

A sending rule can also be supplied as a prefix to the above to determine when email should/should not be sent.

Example

mail_customer=+L:true

The above example will send an email to the customer (as per the bill_email value) only when you released a deferred payment.

'-' (don't send one) or '+' (do send one)

The possible values are:

- R Refund
- D Deferred
- L Released
- P Repeated
- A Normal Authorisation
- F Referral
- C Communication problem
- S Refund Failure
- M Release Failure
- Q Repeat Failure
- N Normal Payment Failure.

The default rule is '+FARDLPQ'

10.5 mail_merchants

This option is used to send a colon (not semi-colon) delimited list of recipients for the order confirmation email. This will override whatever default is setup on your Customer Account at Pay360 by Capita.
To not send any emails to the Merchant, a single ':' will work. You can also use the sending rule prefixes as per section 13.4

Example

<input type="hidden" name="mail_message" value="Thanks for ordering from Website.com!">

NOTE: The value you send for mail_message must not contain any commas.

10.6 mail_subject

Sets the subject line on order confirmation emails

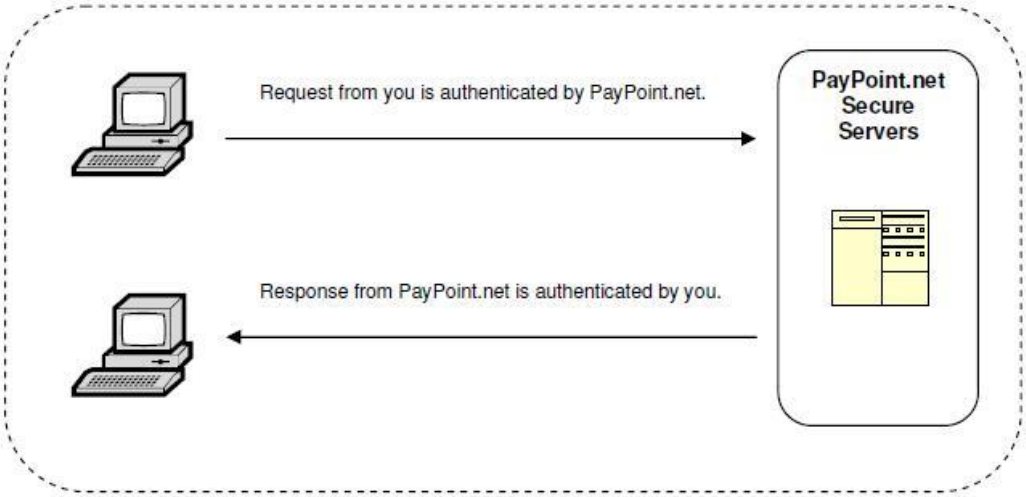
11 Security

There is a mandatory process which allows authentication to occur between your server and the Pay360 by Capita servers. This process involves the sharing of data encrypted using the MD5 algorithm, and a remote password or secret digest key known only to you and Pay360 by Capita.

Implementing this relatively simple security measure is mandatory and you will not be able to successfully submit transactions until it is complete. By doing this you are protected against one of the most common forms of fraud. This method of authentication ensures the authenticity and integrity of information shared between your server and the Pay360 by Capita servers.

Authentication can be implemented during the request from your server to the Pay360 by Capita servers and also during the callback from the Pay360 by Capita servers.

Figure 5: Request and Response Authentication



11.1 What is MD5?

MD5 is a one-way encryption algorithm that takes as input a message of arbitrary length and produces as output, a 128-bit 'fingerprint' or 'message digest' of the input. What is used as the 'input' depends upon the context in which the algorithm is being used, as you will see below. There are many implementations of MD5 available for many different programming languages.

NOTE: When calculating a hash using MD5, always ensure that you use UTF-8 encoding, and not Unicode.

11.2 Authentication from You to Pay360 by Capita (using the remote password)

In order for Pay360 by Capita to be able to be sure that a request to process a transaction actually came from your web application, you need to authenticate yourselves to us.

This is done by POSTing the digest optional parameter to Pay360 by Capita (along with the other mandatory parameters)

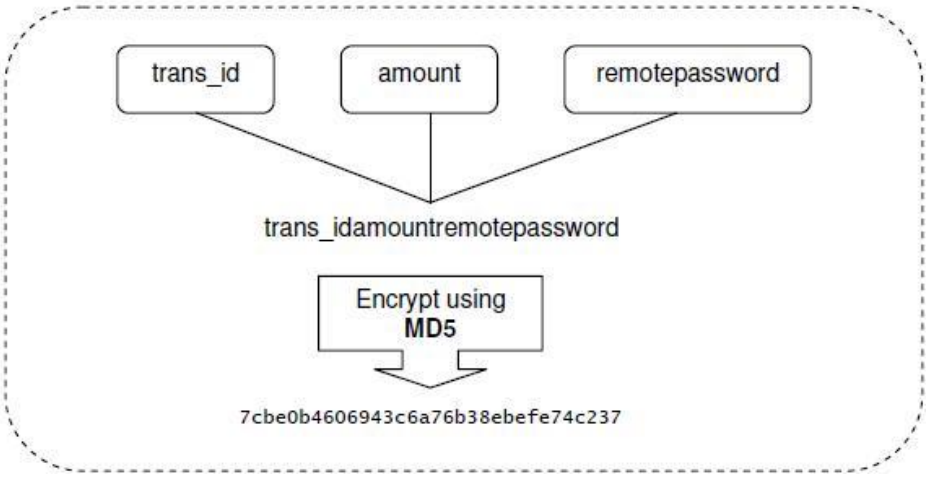
For example:

<input type="hidden" name="digest" value="7cbe0b4606943c6a76b38ebefe74c237">

11.2.1 Creating Your Encrypted Digest Parameter

The value of the digest request parameter shown in the example above is created by concatenating the trans_id and amount for a given transaction, along with your remote password, known only to you and Pay360 by Capita. The following process must take place on your server before posting the transaction details to Pay360 by Capita

Figure 6: Creating the Digest



Note: The remote password can be configured from within the Pay360 by Capita Merchant Extranet (Click on 'Change Remote Password" and select 'Remote' from the dropdown list).

When the POST request from your server is received by Pay360 by Capita, the same process takes place to build the MD5 encrypted string. If the string created by Pay360 by Capita matches the string sent by you in the value of the digest request parameter, then we know that the request came from you and that none of the data used to create the digest was altered in transit, therefore the transaction is permitted to proceed as normal.

11.2.2 Ensuring Pay360 by Capita checks for Authentication

In order to ensure that Pay360 by Capita knows to check the request from your application for authentication, you must have requested us to set this up on your account: This is not in place by default.
Once you have tested your integration to be sure that the digest key is being submitted correctly, please ask for this to be done by emailing 'gatewaysupport@Pay360 by Capita, quoting your Pay360 by Capita account ID and requesting that the 'req_digest=true' option is added to your account.

11.3 Authentication from Pay360 by Capita to You (using the digest key)

In a similar method to that used by Pay360 by Capita to authenticate a request from you, you may authenticate a response from Pay360 by Capita. Each time a callback is made to your server after a transaction has either been authorised or declined, the hash parameter is sent as one of the callback parameters.
The hash callback parameter is created on the Pay360 by Capita server before it is sent to your callback page. The callback to your server is made as a GET request by default. However, you may wish for this callback to be made as a POST request, and can configure this by including the optional parameter 'cb_post=true' in an options field that you may send to Pay360 by Capita
The method by which Pay360 by Capita (and consequently you) create the hash parameter before sending it to your callback page, depends on whether or not you are receiving a GET callback or a POST callback.

11.3.1 Creating a GET Request hash

When the callback to your server is made with a GET request (which is the default) the MD5 hash is generated using the request URL of the callback request.

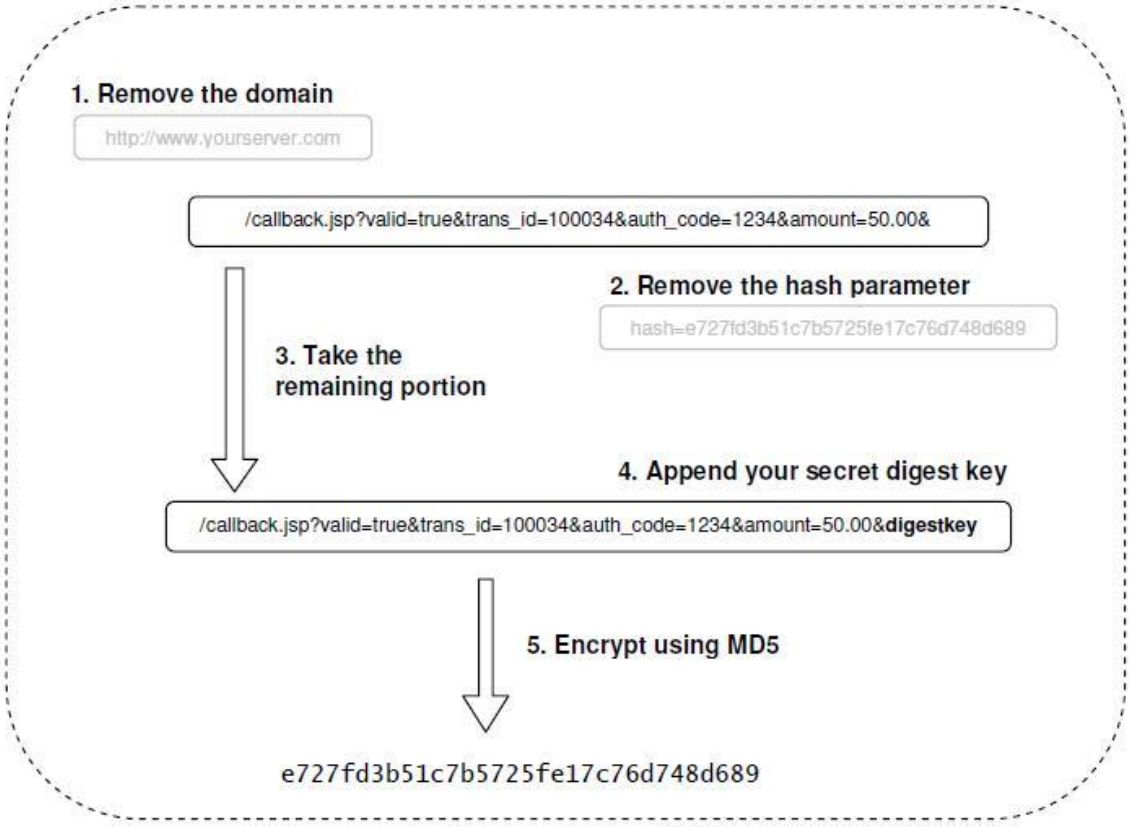
For example

If the callback made from Pay360 by Capita to your server was:

http://www.example.com/callback.jsp?valid=true&trans_id=1000034&auth_code=1234&amount=50.00&hash=e727fd3b51c7b5725fe17c76d748d689

You would be able to use the hash parameter to determine whether or not the request actually came from Pay360 by Capita and avoid processing 'spoofed' callbacks from fraudsters.

Figure 7: Creating a GET Request Hash.



Once you have created your own version of the callback hash, you compare it with the version sent by Pay360 by Capita. If they match – it is a genuine callback from Pay360 by Capita. If they do not match, you may wish to log the IP address from which the bogus request came, and display some kind of warning message.

11.3.2 Creating a POST Request hash

Since a request URL containing transaction specific information is not available in a POST request (as there is no query string), it cannot be used for the creation of an MD5 hash in a POST callback. Instead, the MD5 hash is created through the use of certain request parameters. Which parameters are used for this is determined by you through the use of the md_flds optional parameter.

Example

Take the snippet of an HTML form below. This is a portion of a form which sends the mandatory and several optional parameters to Pay360 by Capita:

```
<form name="my_form" method="POST" action="https://www.secpay.com/java-bin/ValCard">
        <input type="hidden" name="merchant" value="abcdef01">
        <input type="hidden" name="trans_id" value="100034">
        <input type="hidden" name="amount" value="50.00">
        <input type="hidden" name="callback" value="http://www.example.com/callback.jsp">
        <input type="hidden" name="md_flds" value="trans_id:amount:callback">
        <input type="hidden" name="options" value="cb_post=true">
```

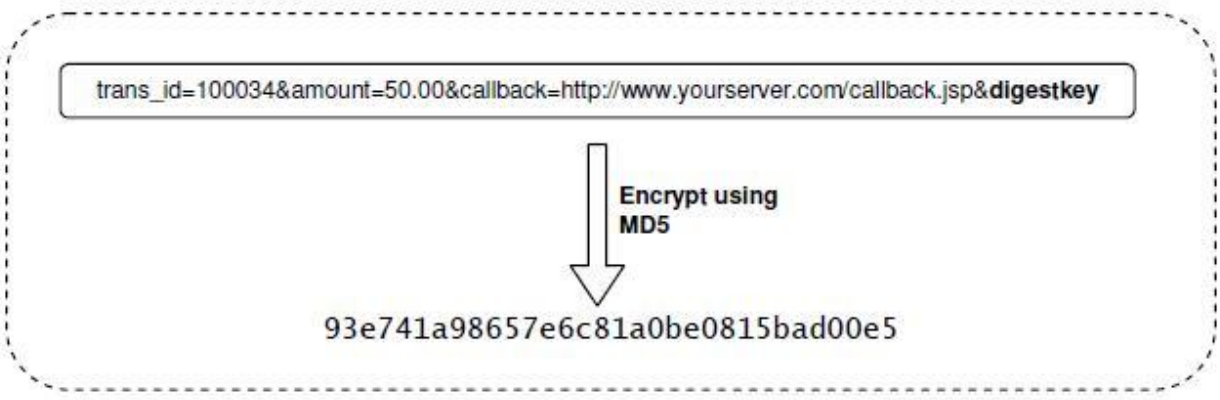
Note:

The md_flds parameter must be sent as its own hidden input field instead of being included in the options field.

All optional parameters can be submitted as separate hidden input fields.

In this instance, the md_flds parameter specifies that the trans_id, amount and callback fields should be used for the creation of the MD5 hash, though any callback parameters could have been chosen. Given the example above, the MD5 hash would be created as follows. In the example above, the optional parameter cb_post=true has been included in the options field telling Pay360 by Capita to send a POST request to your callback page instead of the default GET request.

Figure 8 Creating a POST Request Hash



Where 'digestkey' is the digest key that is set up from within the Pay360 by Capita merchant extranet. As you can see, the key/value pairs of each of your md_flds are concatenated in the same order they were specified and are separated by ampersands. Your secret Digest Key is appended after another ampersand, and the entire string is encrypted. If you hash matches that which is sent in the POST request, then the callback is from Pay360 by Capita.

11.3.3 Customising the md_flds

In order to get around potential limitations with the md_flds functionality, it is possible to customise it slightly. For example, what if the value of one of the parameters you wanted to use in your md_flds contained an ampersand? Or what if the value of one of the parameters you wanted to use in your md_flds contained an ampersand? Or you had already created

your own custom callback parameter named hash? In order to allow for some flexibility in these and other areas, it is possible to alter the following:

- The field name used to send the MD5 hash (The default name is “hash”).
- The md_flds parameter delimiter (The default is ‘&’).
- The attribute name/value delimiter (The default is ‘=’)

To illustrate this, the example below will make the following changes:

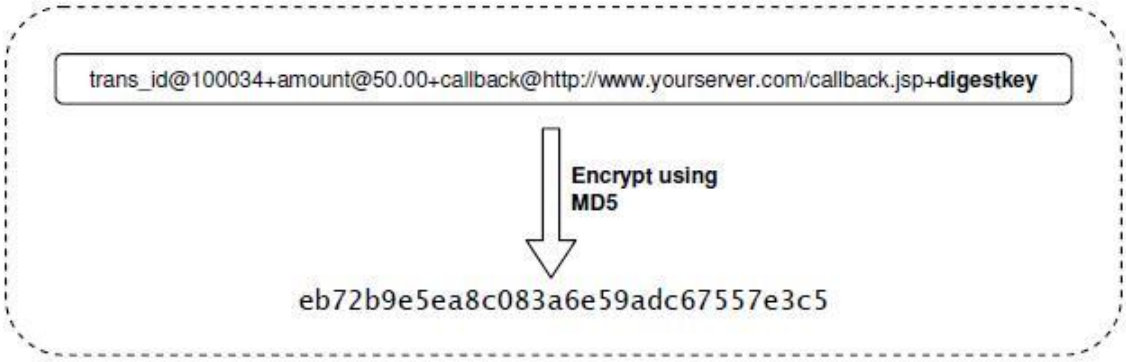
- The name of the request parameter containing the hash will be changed to “signature”.
- The md_flds parameter delimiter will be a ‘+’ instead of a ‘&’.
- The attribute name/value delimiter will be ‘@’ instead of a ‘=’.

<input type=“hidden” name=“md_flds” value=“signature,+,@/trans_id:amount:callback”>

This must be sent as a separate parameter. The values before the last slash (“/”) represent the hash parameter name, md_flds parameter delimiter and attribute name/value delimiter respectively. The values after the last slash represent the names of each of the request parameters being sent which should have their values used in the creation of the callback hash.

With the above md_flds field, the callback hash would be sent back in the request parameter named signature, and would be created as follows:

Figure 9: Creating a custom md_flds Hash



12 Iframes

One feature of a freedom account is the ability to embed the payment page in an Iframe. (“inline-frame”)
If you don’t want the customer to leave your website at all and be redirected to our secure payment page, it is possible to host the page using an Iframe on your payment page.

N.B. this is noted here as confirmation that it is possible. Pay360 by Capita will be unable to assist in the implementation of an Iframe on your site.

NOTE: In order to do this, you must ensure that your checkout page that will host the Iframe is covered by an SSL certificate (i.e. is an https:// link not an http://), otherwise you will not be deemed to be PCI compliant.

13 Using the Pay360 by Capita API

13.1 Protocols

13.1.1 XMLRPC

This is a technical standard specification and a set of implementations that allow software running on disparate operating systems, running in different environments to make procedure calls over the Internet.
It's a remote procedure, calling using HTTP as the transport and XML as the encoding. XML-RPC is designed to be as simple as possible, while allowing complex data structures to be transmitted, processed and returned – please see:

https://www.secpay.com/secxmlrpc/make_call

Details of the XML-RPC protocol can be found at www.xmlrpc.org

Example code donated by merchants/developers can be found here:

http://www.paypoint.net/support/gateway/soap-xmlrpc

13.1.2 SOAP

SOAP stands for "Simple Object Access Protocol", and is a lightweight XML-based communications protocol designed for the exchange of information in a platform-independent, distributed environment.
Pay360 by Capita leverages the RPC (Remote Procedure Call) capabilities of SOAP to provide a secure, remote interface to our transactional network from within your own code, as though you were making a call to a local method or function.
SOAP Services are defined using the WSDL (Web Services Definition Language) and are accessible via a URL which is known as a SOAP endpoint.

Endpoint: https://www.secpay.com/java-bin/services/SECCardService
WSDL: https://www.secpay.com/java-bin/services/SECCardService?wsdl

Example code donated by merchants/developers can be found here:
http://www.paypoint.net/support/gateway/soap-xmlrpc

Java Apache SOAP: http://ws.apache.org/soap/
N
Perl SOAP Lite: http://www.soaplite.com
PHP SOAP Toolkit: http://phpsoaptoolkit.sourceforge.net/phpsoap/

13.2 Real-time Transactions

13.2.1 Real-time Transaction Request Parameters

Method Name: SECVPN.validateCardFull.

PARAMETER	EXAMPLE	DEFINITION
mid	secpay	This is your Pay360 by Capita Gateway account name (usually six letters and two numbers). You can see this ID in the top right corner while logged into the Extranet.
vpn_pswd	secpay	Your VPN password can be set from within the Extranet (https://www.paypoint.net/secnet/app) – Click on "Account", then "remote Passwords" and select "VPN" from the drop-down list.
trans_id	TRAN00001	A unique transaction identifier created by yourself. This can be used to refer to a transaction at a later date (to refund it for example).
ip	127.0.0.01	The IP address that the card holder's machine is presenting to the internet.
name	Mr Cardholder	The Cardholder's name as it is on their card
card_number	4444333322221111	The card number (this should contain no spaces of hyphens). The example card number shown to the left is a test Visa card which can be used during development. Any valid expiry date which is the in future can be used with this card number. Please see the Appendix for different card type examples.

amount	50.00	The Amount for the transaction. This should contain no currency symbols or formatting (for example, do not send any amount with a comma in).
expiry_date	0119	The expiry date on the card. Should be formatted either as mm/yy or mmyy.
issue_number	1	The issue number on the card. This only applies to Maestro or Solo cards. If the card in use does not have an issue number, then an empty string should be passed in.
start_date	0109	The Start date on the card. If the card does not have a start date then an empty string should be used.
cv2	123	The CV2/security code on the card. 3 digits for most cards and 4 digits for American Express.
order	prod=funny_book,item_amount=25.00x1;prod=sad_book,item_amount=16.00x1	Used to submit order details relevant to this transaction.
shipping	name=Fred Bloggs,company=Online Shop Ltd,addr_1=Dotcom House,addr_2=London Road,city=Townville,state=Countyshire,post_code=AB1 C23,tel=01234 567890,fax=09876 543210,email=somebody@paypoint.net,url=http://www.somedomain.com	Used to submit shipping address details relevant to this transaction.
billing	name=Fred Bloggs,company=Online Shop Ltd,addr_1=Dotcom House,addr_2=London Road,city=Townville,state=Countyshire,post_code=AB1 C23,tel=01234 567890,fax=09876 543210,email=somebody@paypoint.net=http://www.somedomain.com	Used to submit billing address details relevant to this transaction.
options	test_status-true,dups=false,card_type=Visa,cv2=123, currency=EUR	Used to submit optional parameters which are used to later the behaviour of this transaction. It is possible to add CURRENCY as an optional parameter if required.

13.3 Real time Transaction Response Parameters

The parameters below are the commonly returned parameters for a simple test payment request.

PARAMETER	EXAMPLE	DEFINITION
valid	True	This is true or false and indicates the acceptance or not of the card details.
trans_id	TRAN0001	We send you back the same trans_id that you sent us with your mandatory parameters so that you can update your system appropriately.
code	A	A short code giving extensive details of failure states. This is the parameter that should be used to programmatically determine whether or not a particular transaction was authorised. See Figure 1 below for possible values.
auth_code	9999	This is the authorisation code obtained from the bank for this transaction. This is only returned if valid=true. For a transaction sent when test_status=true, the auth_code will always be 9999.
message	TEST AUTH	A message to give you more information. This should NOT be displayed to the cardholder.

amount	50.00	This is the amount actually authorised by the bank.
test_status	True	Returns the test flag you send in your request. Only returned if you send a test_status parameter.

Please see section 4.5.8 for all possible response codes

13.3.1 Additional Response Parameters

Below are some additional parameters you may receive in the response, depending on the contents of your request.

PARAMETER	EXAMPLE	DEFINITION
card_no	1111	Last 4 digits of the card number. Supplied if you sent “repeat=true” in the options field of your request.
card_type	Visa	Card Type. Supplied if you sent “repeat=true” in the options field of your request.
currency	EUR	The currency of the transaction. Only supplied when a currency other than the default (GBP) is used.
customer	Mr Cardholder	Cardholders name. Supplied if you sent “repeat-true” in the options field of your request.
cv2avs	ALL MATCH	The apacs approved text that is supplied as a result of the CV2 and AVS anti-fraud checks. There are five core values defined; these are: ALL MATCH All the data provided matches that which the card issuer has on record. SECURITY CODE MATCH ONLY Only the security code matched ADDRESS MATCH ONLY Only the address matched NO DATA MATCHES None of the data matched DATA NOT CHECKED The cv2avs system is unavailable or not supported by this card issuer. With all these core codes, and address is only understood to match if and only if, both the address and the postcode match at the same time. This is a little strict for some people so the following codes have been introduced too. PARTAIL ADDRESS MATCH / POSTCODE The postcode matched but the address did not. PARTIAL ADDRESS MATCH / ADDRESS The address matched but the postcode did not. SECURITY CODE MATCH / POSTCODE The security code and postcode matched but the address did not. SECURITY CODE MATCH / ADDRESS The security code and address matched but the postcode did not.
expiry	0119	Expiry date. Supplied if you send “repeat=true” in the options field of your original request.
resp_code	5	This parameter is only returned when a transaction is declined and code=N. This is the bank’s failure code for your information only. Do not show it to the customer

13.4 Example XMLRPC Request

```
<?xml version="1.0"?>
<methodCall>
<methodName>SECVPN.validateCardFull</methodName>
<params>
<param>
<value><string>secpay</string></value>
</param>
<param>
<value><string>secpay</string></value>
</param>
<param>
<value><string>TRAN0001</string></value>
</param>
<param>
<value><string>127.0.0.1</string></value>
</param>
<param>
<value><string>Mr Cardholder </string></value>
</param>
<param>
<value><string>4444333322221111</string></value>
</param>
<param>
<value><string>50.00</string></value>
</param>
<param>
<value><string>0119</string></value>
</param>
<param>
<value><string>1</string></value>
</param>
<param>
<value><string>0109</string></value>
</param>
<param>
<value><string>prod=funny_book,item_amount=25.00x1;prod=sad_book,item_amount=12.50x2</string></value>
</param>
<param>
<value><string></string></value>
</param>
<param>
<value><string></string></value>
</param>
<param>
<value><string>test_status=true,dups=false,card_type=Visa,cv2=123</string></value>
</param>
</params>
</methodCall>
```

13.5 Example XMLRPC Response

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<methodResponse>
<params>
<param>
<value>
?valid=true&trans_id=trans_id&code=A&auth_code=9999&message=TEST
AUTH&amount=10.51&test_status=true
</value>
</param>
</params>
</methodResponse>
```

13.6 Example SOAP Request

<soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:sec="http://secvpn.secpay.com">
<soapenv:Header/>
<soapenv:Body>
<sec:validateCardFull soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
<mid xsi:type="xsd:string">secpay</mid>
<vpn_pswd xsi:type="xsd:string">secpay</vpn_pswd>
<trans_id xsi:type="xsd:string">TRAN0001</trans_id>
<ip xsi:type="xsd:string">127.0.0.1</ip>
<name xsi:type="xsd:string">Mr Cardholder</name>
<card_number xsi:type="xsd:string">4444333322221111</card_number>
<amount xsi:type="xsd:string">50.00</amount>
<expiry_date xsi:type="xsd:string">0119</expiry_date>
<issue_number xsi:type="xsd:string">1</issue_number>
<start_date xsi:type="xsd:string">0109</start_date>
<order
xsi:type="xsd:string">prod=funny_book,item_amount=25.00x1;prod=sad_book,item_amount=12.50x2</order>
<shipping xsi:type="xsd:string"></shipping>
<billing xsi:type="xsd:string"></billing>
<options xsi:type="xsd:string">test_status=true,dups=false,card_type=Visa,cv2=123</options>
</sec:validateCardFull>
</soapenv:Body>
</soapenv:Envelope>

Example SOAP Response

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soapenv:Body>
<ns1:validateCardFullResponse soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:ns1="http://secvpn.secpay.com">
<validateCardFullReturn xsi:type="soapenc:string"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">?valid=true&trans_id=TRAN0001&code=A&auth_code=999
9&message=TEST AUTH&amount=50.0&test_status=true</validateCardFullReturn>
</ns1:validateCardFullResponse>
</soapenv:Body>
</soapenv:Envelope>

14 Refund Transactions

14.1 Refund Transaction Request Parameters

Method Name: SECVPN.refundCardFull

PARAMETER	EXAMPLE	DEFINITION
mid	secpay	This is your Pay360 by Capita Gateway account name (usually six letters and two numbers). You can see this ID in the top right corner whilst logged in to the Extranet.
vpn_pswd	secpay	Your VPN password can be set from within the Extranet. https://www.paypoint.net/secnet/app (Click on 'Account', then 'Remote Password' then select 'VPN' from the dropdown list).
trans_id	TRAN0001	A unique transaction identifier created by yourself. This should be the trans_id of the transaction you would like to refund. If you reference a non-unique trans_id, it is the most recent transaction with this trans_id that will be refunded. Warning: If you deferred and then released your original transaction, you should refund the sales transaction, not the original defer/released.
amount	50.00	The amount to refund. This should contain no currency symbols or formatting (for example do not send an amount with a comma in).
remote_pswd	secpay	Your Remote password can be set from within the Extranet:

		https://www.paypoint.net/secnet/app (Click on “Account” then “Remote Passwords” and select ‘Remote from the drop down list. This differs from the VPN

14.2 Refund Transaction Response Parameters

PARAMETER	EXAMPLE	DEFINTION
valid	true	This is true or false and indicates the acceptance or not of the request.
trans_id	TRAN0001_refund	We send you back the same new trans_id that you sent us with your mandatory parameters so that you can update your system appropriately.
code	A	A short code giving extensive details of failure states. This is the parameter that should be used to programmatically determine whether or not a particular transaction was authorised. See Section 3.2 figure 1 for possible values.
auth_code	9999	This is the authorisation code obtained from the bank for this transaction . This is only returned if valid=true. For a transaction sent when test_status=true, the auth_code will always be 9999.

14.2.1 Example XMLRPC Refund Request

```
<?xml version=“1.0”?>
<methodCall>
<methodName>SECVPN.refundCardFull</methodName>
<params>
<param>
<value><string>secpay</string></value>
</param>
<param>
<value><string>secpay</string></value>
</param>
<param>
<value><string>TRAN0001</string></value>
</param>
<param>
<value><string>50.00</string></value>
</param>
<param>
<value><string>secpay</string></value>
</param>
<param>
<value><string>TRAN0001_refund</string></value>
</param>
</params>
</methodCall>
```

14.2.2 Example XMLRPC Refund Response

```
<?xml version=“1.0” encoding=“ISO-8859-1”?>
<methodResponse>
<params>
<param>
<value>
?valid=true&trans_id=TRAN0001_refund&code=A&auth_code=9999
</value>
</param>
</params>
</methodResponse>
```

14.2.3 Example SOAP Refund Request

<soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:sec="http://secvpn.secpay.com">
<soapenv:Header/>
<soapenv:Body>
<sec:refundCardFull soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
<mid xsi:type="xsd:string">secpay</mid>
<vpn_pswd xsi:type="xsd:string">secpay</vpn_pswd>
<trans_id xsi:type="xsd:string">TRAN0001</trans_id>
<amount xsi:type="xsd:string">50.00</amount>
<remote_pswd xsi:type="xsd:string">secpay</remote_pswd>
<new_trans_id xsi:type="xsd:string">TRAN0001_refund</new_trans_id>
</sec:refundCardFull>
</soapenv:Body>
</soapenv:Envelope>

14.2.4 Example SOAP Refund Response

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soapenv:Body>
<ns1:refundCardFullResponse soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:ns1="http://secvpn.secpay.com">
<refundCardFullReturn xsi:type="soapenc:string"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">?valid=true&trans_id=TRAN0001_refund&code=
A&auth_code=9999</refundCardFullReturn>
</ns1:refundCardFullResponse>
</soapenv:Body>
</soapenv:Envelope>

15 Placing Deferred Transactions

Deferred transactions are transactions that take place in two stages. The first time a deferred transaction is sent to Pay360 by Capita an authorisation takes place and funds in the cardholders account are frozen pending release by the money will not be debited. In order for these frozen funds to be released, you need to follow up with a release request. The frozen funds will timeout and become un-frozen if the deferred transaction is never released.

Deferred transactions are created by submitting the deferred parameter inside the options parameter of a standard real-time transaction request. See Section 3. There are two possible values for the deferred parameter; true or reuse:

deferred=true

If deferred=true is used then only one unit of currency is authorised against (and subsequently frozen pending release) in the initial deferred transaction. This is useful for avoiding typing up funds in your customers account is all you want to do is "get their details into the system".

Important Note: When a deferred=true transaction is released, a new authorisation takes place. Therefore it is entirely possible that an authorised deferred=true transaction may be declined on the subsequent attempt to release it. It is for this reason that we would remind merchants not to ship any goods unless they have successfully released a deferred transaction.

deferred=reuse

If deferred=reuse is used, the initial deferred transaction will authorise for the full amount but on release will use the original auth code (that which was obtained by the initial deferred transaction) if it is still valid. If not still valid, releasing will cause a new authorisation to take place (as is the case with deferred=true).

What do we mean by "still valid"…?

The current validity of an authorisation code that was obtained using deferred=reuse is determined by:

- Whether or not the card used is a credit or debit card.
- The amount of time that has passed since the initial deferred transaction occurred.

By default it will reuse the same authorisation code in the same day for debit cards and within the last 7 days for credit cards. The full syntax is 'reuse:<credit period>:<debit period>'. So the default is implicitly 'reuse:7:1'. If you wish you may alter these default settings when sending the initial deferred transaction.

Examples

Use default settings:

deferred=reuse

Set credit card auth codes to be valid for 5 days and debit card auth codes to be valid for 3 days:

deferred=reuse:5:3

Set debit card auth codes to be valid for 2 days but leave credit card auth code validity period at the default value (7 days):

deferred=reuse::2

Set credit card auth codes to be valid for 6 days but leave debit card auth code validity period at the default value (1 day):

deferred=reuse:6

Important Note: It cannot be 100% guaranteed that a chargeback will not occur just because the original authorisation code was reused. The longer the period of time an authorisation code is considered to be valid, the greater the risk – please bear in mind that you alter the default time periods at your own risk.

16 Releasing Deferred Transactions

16.1 Release Request Parameters

Method Name: SECVPN.releaseCardFull

PARAMETER EXAMPLE DEFINITION

PARAMETER	EXAMPLE	DEFINITION
mid	secpay	This is your Pay360 by Capita Gateway account name (usually six letters and two numbers). You can see this ID in the top right corner while logged into the Extranet.
vpn_pswd	secpay	Your VPN password can be set from within the Extranet: https://www.paypoint.net/secnet/app (Click on “Account” then “Remote Passwords” and select VPN from the drop-down list).
trans_id	TRAN0001	A unique transaction identifier created by yourself. This should be the trans_id of the transaction you would like to release. If you reference a non-unique trans_id, it is the most recent transaction with this trans_if that will be released.
amount	50.00	The amount to release. This should contain no currency symbols or formatting (for example do not send an amount with a comma in)
remote_pswd	secpay	Your Remote password can be set from within the Extranet: https://www.paypoint.net/secnet/app (click on “account” then “Remote Passwords” and select ‘Remote’ from the drop-down list).
new_trans_id	TRAN0001_release	A unique transaction identifier created by yourself. This will be the trans_id of the news transaction you are creating by performing this release.

17 Cancelling Deferred Transactions

To mark a deferred transaction as cancelled so it can't be released, you need to send the release request above with the amount set to -1 and send the new_trans_id as an empty string.

Note: This purely marks the transaction as cancelled on the Pay360 by Capita system and doesn't send a request to
the bank to remove the authorisation.

18 Performing Repeat Transactions

18.1 Repeat Request Parameters

Repeat requests can be used for one off transactions or you can programme your system to call us on a regular basis to take payments.

There are two methods available for repeat requests.

Method Name: SECVPN.repeatCardFull or SECVPN.repeatCardFullAddr

SECVPN.repeatCardFull parameters are shown below. If you use SECVPN.repeatCardFullAddr you must add **shipping**, **billing** and **options** parameters on the end as in a standard transaction request in **Section 3**. You would use SECVPN.repeatCardFullAddr for instance if you wanted to defer the repeat, as you can include the deferred parameter in the options string.

PARAMETER EXAMPLE DEFINITION

Method Name: SECVPN.repeatCardFull

PARAMETER	EXAMPLE	DEFINITION
mid	secpay	This is your Pay360 by Capita Gateway account name (usually six letters and two numbers). You can see this ID in the top right corner while logged into the Extranet.
vpn_pswd	secpay	Your VPN password can be set from within the Extranet: https://www.paypoint.net/secnet/app (Click on "Account" then "Remote Passwords" and select VPN from the drop-down list).
trans_id	TRAN0001	A unique transaction identifier created by yourself. This should be the trans_id of the original transaction that has card details associated with it that you would like to use for this repeat transaction. **Warning:** If you reference a non-unique trans_id, it is the card details associated with the most recent transaction that will be used for this repeat transaction.
amount	50.00	The amount to release. This should contain no currency symbols or formatting (for example do not send an amount with a comma in)
remote_pswd	secpay	Your Remote password can be set from within the Extranet: https://www.paypoint.net/secnet/app (click on "account" then "Remote Passwords" and select 'Remote' from the drop-down list).
new_trans_id	TRAN0001_release	A unique transaction identifier created by yourself. This will be the trans_id of the new transaction you are creating by performing this repeat.
expiry_date	0119	If the expiry date associated with the original transaction which you are referencing in order to create this repeat

		transaction has expired, you may provide a new expiry date using this parameter.
options	test_status=true,dups=false,card_type=Visa,cv2=123	Used to submit optional parameters which are used to alter the behaviour of this transaction.

Note: The amount bears no relation to the amount of the original transaction you are repeating. The request is purely re-using the card details from the original request.

PARAMETER EXAMPLE DEFINITION

Method Name: SECVPN.repeatCardFullAddr

PARAMETER	EXAMPLE	DEFINITION
mid	secpay	This is your Pay360 by Capita Gateway account name (usually six letters and two numbers). You can see this ID in the top right-hand corner while logged into the Extranet.
vpn_pswd	Secpay	Your VPN password can be set from within the Extranet. https://www.paypoint.net/secnet/app (Click on “Account” then “Remote Passwords” and select ‘VPN’ from the drop-down list).
trans_id	TRAN0001	A unique transaction identifier created by yourself. This should be the trans_id of the original transaction that has card details associated with it that you would like to use for this repeat transaction. Warning: If you reference a non-unique trans_id, it is the card details associated with the most recent transaction that will be used for this repeat transaction.
amount	50.00	The amount for the transaction. This should contain no currency symbols or formatting (for example, do not send an amount with a comma in).
remote_pswd	secpay	Your Remote password can be set from within the Extranet: https://www.paypoint.net/secnet/app (click on “account” then “Remote Passwords” and select ‘Remote’ from the drop-down list).
expiry_date	0119	If the expiry date associated with the original transaction which you are referencing in order to create this repeat transaction has expired, you may provide a new expiry date using this parameter.
shipping	name=Fred Bloggs,company=Online Shop Ltd,addr_1=Dotcom House,addr_2=London Road,city=Townville,state=Countyshire,post_code=AB1 C23,tel=01234 567890,fax=09876 543210,email=somebody@paypoint.net,url=http://www.somedomain.com	Used to submit shipping address details relevant to this transaction.
billing	name=Fred Bloggs,company=Online Shop Ltd,addr_1=Dotcom House,addr_2=London Road,city=Townville,state=Countyshire,post_code=AB1 C23,tel=01234 567890,fax=09876 543210,email=somebody@paypoint.net,url=http://www.somedomain.com	Used to submit billing address details relevant to this transaction.
options	test_status=true,dups=false,card_type=Visa,cv2=123	Used to submit optional parameters which are used to alter the behaviour of this transaction.

Note: The amount bears no relation to the amount of the original transaction you are repeating. The request is purely re-using the card details from the original request.

19 Optional Parameters

Below is a list of optional parameters which can be included in the options string.

Remember: Not every method name has an options string!

PARAMETER	EXAMPLE	DEFINTION
card_type	Visa	Possible values for card_type are: • American Express • Delta • Diners Card • JCB • Master Card • Solo • Maestro • Visa • Laser You should check with your merchant bank which card types you can accept. Delta is used for Visa Debit Connect and Electron Cards.
dups	false	See section 23.2
repeat	true	See section 18. As you can see, there are two uses for this option!
repeat	20120901/monthly/12:50	See section 18. As you can see, there are two uses for this option!
test_status	true	See section 23
usage_type	E	The usage_type parameter is used to advise us what type of transaction you are processing. If you don't supply a usage_type then the default setting is Electronic Commerce. (E) There are three possible values: • Usage_type=M (MOTO - Mail Order/Telephone Order) • Usage_type=E (Electronic Commerce) • Usage_type=R (Recurring Payments)

20 Transaction Reports

20.1 Transaction Report Parameters

Method Name: SECVPN.getReport

PARAMETER	EXAMPLE	DEFINITION
mid	secpay	This is your Pay360 by Capita Gateway account name (usually six letters and two numbers). You can see this ID in the top right corner whilst logged into the extranet.
vpn_pswd	secpay	Your VPN password can be set from within the Extranet. https://www.paypoint.net/secnet/app (Click on "Account" then "Remote Passwords"and select VPN from the drop down list).
remote_pswd	secpay	Your Remote password can be set from within the Extranet. https://www.paypoint.net/secnet/app (Click on "Account" then "Remote Passwords" and select Remote from the drop down list).
report_type	xml-report	Possible report types are: • CSV • CSV-All • CSV-Summary • CSV-Detail • CSV-Five • Summary • Statement • Origin-Statement • XML-Report Each report type returns different data in different formats so you will need to find the one that suits your requirements.
cond_type	date	Possible condition types are: • Date • Batch • TransId
condition	GBP	Represents the actual value for the condition type chosen. e.g. By Date 2000 : Transactions for 2000 200004-200005 : Transactions for April and May 2000 200004- : Transactions from April 2000 to present By Transaction 12345678 : Trans Id 12345678 only 1234% : Transactions starting 1234 1234%5678 : Transactions starting 1234 and ending 5678 12345678~12349999 : Transactions between 12345678 and 12349999 By Batch No. 100 : Batch 100 only 100-110 : Batches 100 through 110 **Note: There must be at least four characters preceding the '%'**
currency	GBP	The 3 character currency code specifying the currency of the transactions you would like included in the report.
predicate		Arbitrary Condition - advanced users only, otherwise use an empty string.
html	false	Boolean value specifying whether to return report as html or not.
showErrs	false	Boolean value specifying whether or not unauthorised transactions should be included in the report results.

21 Testing

You can test your integration using your own account (for example: abcdef01) through the use of the test_status optional parameter. See **Section 13.2** test_status

Test Card Numbers

Below are some test card numbers. These can be used with any name and address details, any 3 digit CV2 security code and any expiry date which is in the future.

VISA	**Maestro**
4444333322221111	6799999999593

MasterCard Credit

5555555555554444
5105105105105100

You can of course make an attempt to use an expiry date which is set in the past in order to see what happens in that event!

21.1 test_status

The test_status is used to stimulate either an authorised or declined response from Pay360 by Capita so that you can confirm
that your web application behaves appropriately in each instance. There are three possible values:

- **test_status=true** Simulate an authorised callback without contacting the bank
- **test_status=false** Simulate a declined callback without contacting the bank
- **test_status=live** Send the transaction to the bank for authorisation

The test_status parameter is sent in the options string.

Note:

If you are using 3D secure, the expiry month affects the response from the 3D secure servers while in test mode.

21.2 dups

By default Pay360 by Capita will stop the same trans_id from the same user being used twice in the same hour (this is to stop
duplicates).

Note: We believe this feature reduces customer bills by approximately 10% and also reduces the administration overhead, it is a 'best efforts' feature however and can never be 100%. You must accept that occasionally you will get duplicates.

However, sometimes during the testing phase of your integration, it is useful to not have to bother creating a new trans_id for each transaction you send to Pay360 by Capita.

If you wish to send test transactions with the same trans_id but want Pay360 by Capita to not perform duplicate checks, i.e.:
you want Pay360 by Capita to allow these duplicate transactions into the system without blocking them as duplicates then you
must send the dups parameter with a value of false within the options string. This will tell Pay360 by Capita to “turn off” duplicate checking for this transaction. This should be removed before going live.

21.3 default_cv2avs

When you submit a payment with the test_status parameter, the details obviously don’t go to the bank for checking, so you can use the default_cv2avs parameter in the options string to simulate the cv2avs response. See **Section 3.3** for the

possible values.

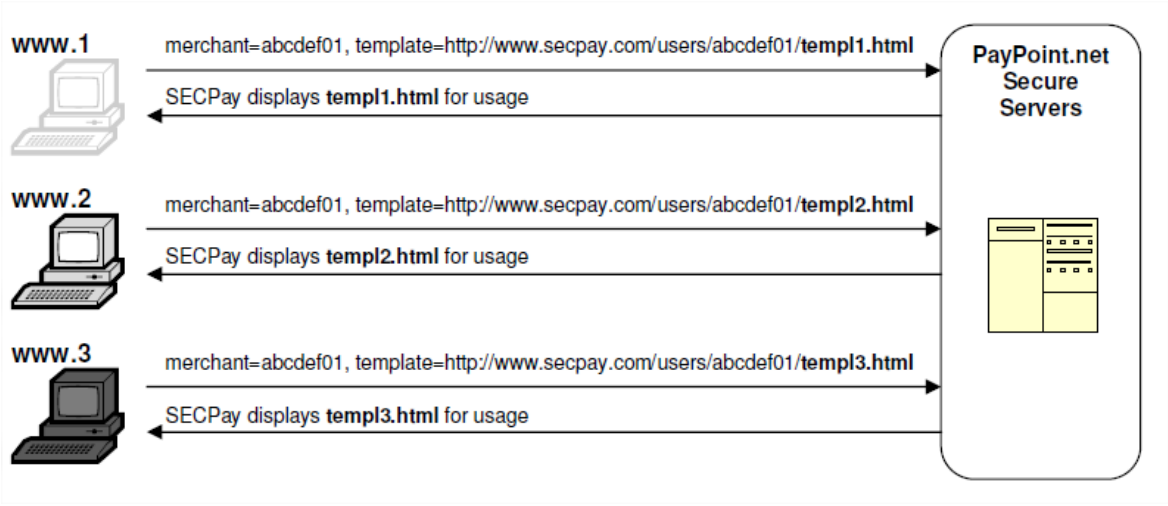
Example

default_cv2avs=ALL MATCH

Note: This is especially useful if you have set the CV2-AVS Options within the Extranet and don't want to have to remove them for testing.

22 Pay360 by Capita Custom Hosted Payment Pages

If you are going to use the Hosted Payment Pages, the page into which cardholders enter their credit card details is stored on the Pay360 by Capita servers and "linked to" from your website. You have the ability to customise this template so that its
look and feel blends in with the rest of your site. Unlike many other Payment Service Providers, it is possible for you to have several different websites sending transactions through one Pay360 by Capita account (which has associated with it one
Internet Merchant Account). You are therefore free to have several different templates, each of which fits in with your various websites.



22.1 Customising Your Template

The default payment page template can be found at the following address:

http://www.secpay.com/users/template.html

Please visit the above page and save the HTML source code to your computer.

The first thing you will notice when looking at this page is that it contains many server side variables such as ${bgcolor} These variables are replaced at runtime by software running on the Pay360 by Capita servers and should therefore be left intact. These variables exist in order to allow you to dynamically set their values without altering your template. The value of each variable can easily be set by sending a request parameter with exactly the same name. For example, you will notice that the standard template has a variable named ${bgcolor}. If you include the following hidden input field in the form which you POST to Pay360 by Capita:

<input type="hidden" name="bgcolor" value="hotpink">
..then the ${bgcolor} variable in your template would be replaced with the value "hotpink" and your template would be a nice shade of pink. This works for all of the variables present in the template and can be used for altering text, colours etc.

Tip: sending a parameter with the same name as a variable in your template will dynamically replace that variable with the value of your parameter!

Pay360 by Capita do not impose any restrictions on how you customise the look and feel of your template. You may add your
own graphics and stylesheets and restructure the layout of the form fields in any way you choose. We do not specify that you must include our logo – the look of this page is 100% up to you.

Warning: avoid using Frontpage, Dreamweaver or any other WYSIWYG design tools with this template. More often than not, the use of one of these tools will "break" the template either by incorrectly manipulating the server side variables or by moving the </form> tag around. We therefore recommend that this template is customised by hand.

Any images you add to your template must be uploaded to the Pay360 by Capita secure servers and referenced absolutely
using an HTTPS URL.

Example

<img src=https://www.secpay.com/users/abcdef01/logo.gif>

22.2 Uploading Your Template

Once you have customised your template you may upload it to the Pay360 by Capita servers from within the Extranet (https://www.paypoint.net/secnet/app).

In the Extranet, choose Account then File Manager on the left hand side menu. This should display a form and allow you to upload your template and associated files through the browser.

22.3 Referencing Your Template

In order to utilise your own custom template you must add the request parameter template to the list of parameters you post to the Pay360 by Capita servers.

The template parameter is not mandatory and therefore if you do not use it, the default card payment page would be displayed to the cardholder and your template will not be used.

After you have uploaded your template and other files to the Pay360 by Capita server, these files will be located in a directory
which has the same name as your Pay360 by Capita username.

Example

http://www.secpay.com/users/abcdef01/mytemplate.html

…and you would specify that it should be used by including the following request parameter amongst those that you POST to Pay360 by Capita:

<input type="hidden" name="template" value="http://www.secpay.com/users/abcdef01/mytemplate.html">

Note: The location of your template file should be specified using http and not https.

22.4 Customising the err_template

This is the HTML template we use for displaying error messages, if you wish you can supply your own by using this parameter. Depending on the error, our system calls one of two templates.
This one is called if there is a problem with your callback page:

https://www.secpay.com/errFile.html

This one for other errors:

http://secpay.com/vc_blank.html

If you want to customise the error pages, you will need to combine the code in the above two pages into one page. This must be uploaded to the Pay360 by Capita servers from within the Extranet (https://www.paypoint.net/secnet/app) then referenced to by using the err_template parameter in your form POST.

Example

<input type="hidden" name="err_template" value="http://www.secpay.com/users/abcdef01/errortemplate.html">

Note: If this URL is wrong then there will be no template for displaying the error message to that effect, you have been warned...

22.5 Customising Payment Page Error Messages

A set of parameters you can POST in your form that allows you to tailor the text shown in the event of an error. This is useful for tailoring error message text and foreign language errors.

Parameter	Default Value
card_no_Error	Card No
card_type_Error	Card Type
customer_name_Error	Customer Name
start_date_Error	Start Date
expiry_date_Error	Expiry Date
issue_Error	Issue No

Warning: Please remember – all parameters are case sensitive!

23 Preventing Fraud

Pay360 by Capita offers a number of optional additions to your installation to help reduce the chance of fraudulent transactions being processed.

23.1 3D Secure

3D Secure is run by MasterCard and Visa ('MasterCard Secure Code' and 'Verified by Visa'); it requires the customer to enter an additional password to complete the checkout process and works with all enabled MasterCard and Visa cards.

23.2 SecGuard

SecGuard is Pay360 by Capita's own offering and allows you to configure rules that affect certain transactions. For example, it is possible to set a rule up that marks any transaction over £500 as 'deferred' to allow you the chance to assess whether or not you believe it to be fraudulent.

Both of these options can be discussed further and set-up by talking to your Account Manager.

NOTE:

3D Secure involves a one off cost of £5
SECGuard involves a monthly fee of £10.

24 Going Live

Once testing is complete, it is time to go live. There are a few pitfalls that regularly catch merchants and developers out during the go-live stage of integration. Below is a checklist which should help you to avoid the more common of these.

24.1 Have you gone live with the bank?

When registering with Pay360 by Capita, you (or your client if you are a developer) will need to provide your Internet Merchant Account (IMA) number that you have with your acquiring bank. This account number represents the bank account through which your transaction will be authorised over the internet in real time.

‘Going live with the bank’ is the process of having your acquiring bank allocate you a Terminal ID (TID) through which Pay360 by Capita can send transaction for authorisation. A ‘Terminal ID’ can be thought of as a ‘virtual cashier’ that processes transactions sent from Pay360 by Capita. Until you have ‘gone live’ with the bank, there will be no TID allocated and therefore the bank will be unable to process your transaction.

In order to go live and get your TID allocated, you must email ‘gatewaysupport@Pay360 by Capita’ quoting your relevant Pay360 by Capita username (e.g. abcdef01) and state explicitly that you would like to go live with the bank. You do not need to contact your bank as this will be done on your behalf by Pay360 by Capita.

Note:

It takes the bank a minimum of two working days to allocate you a TID. Please ensure that this is built into the timescales on your project plans.

24.2 Are you still using a test account?

Please ensure that if you have previously been using a test account, that you have changed to using your own account. You can check this by viewing the hidden ‘merchant’ field in the form hosted on your server which contains your mandatory parameters:

<input type=“hidden” name=“merchant” value=“abcdef01”>

24.3 Are you still in test mode?

Please check that you are not sending the test_status parameter with a value of either ‘true’ or ‘false’. If you want to be live, you should send test_status=live, or better still, do not include the ‘test_status’ parameter at all!

24.4 Are you still using “dups=false”?

During testing, you may have added the optional parameter ‘dups=false’ to prevent Pay360 by Capita from blocking transaction which are sent through with the same trans_id (see section 15.3). Please ensure that the ‘dups=false’ parameter is NOT included in your options.

24.5 Have you enabled 3D Secure?

If required.

This requires a request to us (to support@paypoint.net) and you should allow up to 10 working days for confirmation that it has been activated.

This is also available in a tick-list (Appendix: B Go Live Tick-list)

25 Questions

For any integration questions please:

Email:

support@paypoint.net

Phone:

0333 313 7161

26 Appendix A: Trouble-Shooting

Use this section to help determine what could be causing the problem you're experiencing.

Error	**Unknown Acquirer: You do not have a merchant account for this acquirer.**
Cause(s)	• Attempting a transaction with a card type you can't accept. • Attempting a transaction in a currency you can't accept. • Supplying an incorrect account ID.
Resolution	• Try a different card type or currency you can accept. • Ask your merchant bank how you can start to accept the card type/currency you are trying • Double Check that you site is sending us the correct account and, and it is lower case.
Error	**?valid=false&code=P:M&message=Unknown_customer+:+for+testing+only+use+'secpay'&correct=false**
Cause(s)	Not supplying a value for 'merchant'
Resolution	You should be sending your Pay360 by Capita account ID in the merchant parameter e.g. abcdef01 If you think you are – it's worth checking with your </form> tag is located
Error	**Java.io.IOException: Server returned HTTP response: 500 for URL:**
Cause(s)	Your callback page is either unavailable or there is a problem with the code on it.
Resolution	Check for server issues with your host, and double check the callback path specified.
Error	**Java.io.FileNotFoundException**
Cause(s)	Your callback page could not be found.
Resolution	Ensure the callback URL you specified is correct and that the page is there.
Error	**Java.net.UnknownHostException**
Cause(s)	We could not resolve your host name
Resolution	Check that the host name in the callback URL you specified is correct.
Error	**Java.net.SocketTimoutException**
Cause(s)	We could not load all the content on your callback page within a set time period
Resolution	Check for web server issues and the availability of third party content providers (e.g. stats tracking).

Error	**Digests do not match : possible fraud attempt**
Cause(s)	The req_digest=true option is on your account, but the digest value is either blank or invalid.
Resolution	See Section: 14 Security on how to create the digest.

Error	Cannot find tid for this merchant number
Cause(s)	• Attempting a live transaction on your account when we haven't set a Terminal ID up for you. • Attempting an eCommerce transaction on an account that is only meant for MOTO payments.
Resolution	• Send an email to support@paypoint.net and ask for your account to be set live. • Send an email to support@paypoint.net and ask if your account is set to accept eCommerce transactions.
Error	**This is a test transaction. (Obviously only an error if you were expecting a live transaction!)**
Cause(s)	Either your account has test mode on, or your site is sending a 'test_status='true' or 'false' option.
Resolution	• Login to the extranet, select 'Account' on the top menu, then 'account options' and ensure that 'Test Mode' is set to 'OFF'. • Check in the website code that you are not sending 'test_status=true' or 'false' (It should be 'live') • If you have a shopping cart (e.g. Zen Cart) there will probably be a setting within the payment module that you need to alter.

27 Appendix B – Go Live Check List

- ☐ Have you tested all relevant cards?
- ☐ Have you 'gone live' with the bank?
- ☐ Have you informed support@paypoint.net that you'd like to go live?
- ☐ Have you reverted 'dups=false' to 'dups=true'?
- ☐ Have you turned test mode off? (check both the code and the account options)
- ☐ Have you changed all test values to the real merchant values?

OPTIONAL

- ☐ Have you enabled 3D secure? (If required)
- ☐ Have you configured SECGuard rules? (if required)