

Sage Pay Form Integration and Protocol Guidelines 3.00

Published: 27/08/2015

Table of Contents

Document Details	4
Version History	4
Legal Notice	4
1.0 Introduction	6
2.0 Overview of Form Integration	7
3.0 Form Integration in Detail	9
Step 1: The customer orders from your site	9
Step 2: Your server builds a Confirmation Page	10
Step 3: Customer enters payment details on Sage Pay's server	12
Step 4: Sage Pay checks for 3D-Secure enrolment	13
Step 5: Sage Pay redirects your customer to their Issuer	14
Step 6: Issuing bank returns the customer to Sage Pay	15
Step 7: Sage Pay servers request card authorisation	16
Step 8: Sage Pay redirects the customer to your website	17
Step 9: Sage Pay sends Settlement Batch Files	18
4.0 Integrating with Sage Pay Form	20
5.0 Testing on the Test Server (Stage 1)	21
5.1 Registering a Payment	21
5.1.1 Test card numbers	24
5.2 Accessing MySagePay on Test	25
5.3 Refunding a transaction	28
6.0 Additional Transaction Types	29
6.1 DEFERRED transactions	29
6.2 REPEAT payments	30
6.3 AUTHENTICATE and AUTHORISE	30
6.4 REFUNDS and VOIDS	31
7.0 Applying Surcharges	32
8.0 Sage 50 Accounts Software Integration	33
9.0 Going Live (Stage 2)	34
10.0 Congratulations, you are live with Sage Pay Form	35
11.0 Character Sets and Encoding	36
Appendix A: Transaction Registration	37
A1. Form Fields	37
A1.1 The Crypt Field	38
A1.2 Example Crypt Field	38

A1.3	Request Crypt Fields	39
A1.4	SurchargeXML	46
A1.5	Basket	47
A1.6	BasketXML	48
A1.7	CustomerXML	54
Appendix B: Transaction Completion		55
B1.	Response Crypt Fields	56
12.0	URLs	61

Document Details

Version History

Date	Change	Page
27/08/2013	Document published.	
	Added Expiry Date as a returned field.	56
	Basket XML includes Discounts.	45
	Removed reference to repeats for PayPal.	26
	Allowed characters in BankAuthCode now Alphanumeric.	56
30/01/2014	New screenshots.	---
	References to Sage Pay website updated.	---
	Example Crypt field.	33
	Removed reference to Laser Cards.	---
	Surcharge XML clearer.	28
01/08/2014	Rebranded.	---
	Included additional fields for Financial Institutions (MCC 6012).	40
	Information on pre-authorisations.	25
	Sage Software.	29
	3D-Secure simulation.	21
	XML snippets moved to sagepay.com	---
	Updated Test Cards.	20
	Added European / PayPal indicators.	---
	Basket XML Amendments.	44
05/01/2015	Maestro LUHN exception.	10
	Corrected contents of crypt field	34
13/02/2015	Updated payment pages to responsive designs	19
27/08/2015	Remove validation from Basket XML	48

Legal Notice

This Protocol and Integration Guidelines document ("Manual") has been prepared to assist you with integrating your own (or your client's) service with Sage Pay's payment gateway. You are not permitted to use this Manual for any other purpose.

Whilst we have taken care in the preparation of this Manual, we make no representation or warranty (express or implied) and (to the fullest extent permitted by law) we accept no responsibility or liability as to the accuracy or completeness of the information contained within this Manual. Accordingly, we provide this Manual "as is" and so your use of the Manual is at your own risk.

In the unlikely event that you identify any errors, omissions or other inaccuracies within this Manual we would really appreciate it if you could please send details to us using the contact details on our website at www.sagepay.com.

We may update this Manual at any time without notice to you. Please ensure that you always use the latest version of the Manual, which we publish on our website at www.sagepay.com, when integrating with our payment gateway.

Copyright © Sage Pay Europe Limited 2015. All rights reserved.

1.0 Introduction

This guide contains all essential information for the user to implement Sage Pay using Form integration.

Sage Pay's Form integration provides a straightforward hosted payment interface for the customer and takes complete responsibility for the online transaction, including the collection and encrypted storage of payment data. Eliminating the security implications of storing such sensitive information on your own servers and removing the need for you to maintain highly secure encrypted databases or obtain digital certificates.

Form integration is designed for merchants who have less experience in server side scripting or use shared web servers that do not offer database services. With Form integration all transaction information is held at Sage Pay, including the full basket contents and customer details. Emails can be sent to both you and your customers to confirm the success or failure of a transaction.

This document explains how your website should communicate with Sage Pay, how to integrate with our test and live environments, and contains the complete Form protocol in the Appendix.



Indicates additional information specific to European Payment method transactions.



Indicates additional information specific to PayPal transactions.

2.0 Overview of Form Integration

The final 'Pay Now' button on your website is your link to the Sage Pay gateway. Once the customer has selected their purchases, entered billing and delivery details on your site, you present them with an order summary and the option to 'Pay Now'.

What the customer does not see is that whilst generating their order summary a simple piece of server-side scripting builds an encrypted hidden field that it places on the form. This field contains all the transaction information in a format that the Sage Pay gateway can understand. When the customer clicks 'Pay Now' the encrypted contents of that field are POSTed to the Sage Pay gateway and the customer redirected to the hosted Sage Pay payment pages.

The Sage Pay payment pages will present the customer with available payment methods and allow them to enter their payment details. The hosted payment pages can carry your logo and a `Description` of the goods that the customer is paying for, so they can remain confident they are buying from you. You can even customise the payment pages to carry the look and feel of your site at no additional cost. You can download our payment page templates from sagepay.com. Please note; the most recent responsive designs are not yet customisable but you can continue to customise and use our older design.

Once the customer has selected their payment method and supplied their details, they are shown a full summary of their order, including the basket contents (provided this was included in your post) and asked to confirm that they wish to process. If applicable, Sage Pay will request authentication from the 3D directory (Verified by Visa, MasterCard SecureCode and Amex SafeKey), provided the result passes the rules you have set in MySagePay, then we request authorisation from your acquiring bank. Once the bank has authorised the payment (and assuming the address and card security code results pass the rules you have set) your customer is redirected back to a success page on your site. If authorisation fails, the customer is redirected back to a failure page on your site. Both pages are sent encrypted information which you can decrypt to obtain and extract detailed information about the transaction.

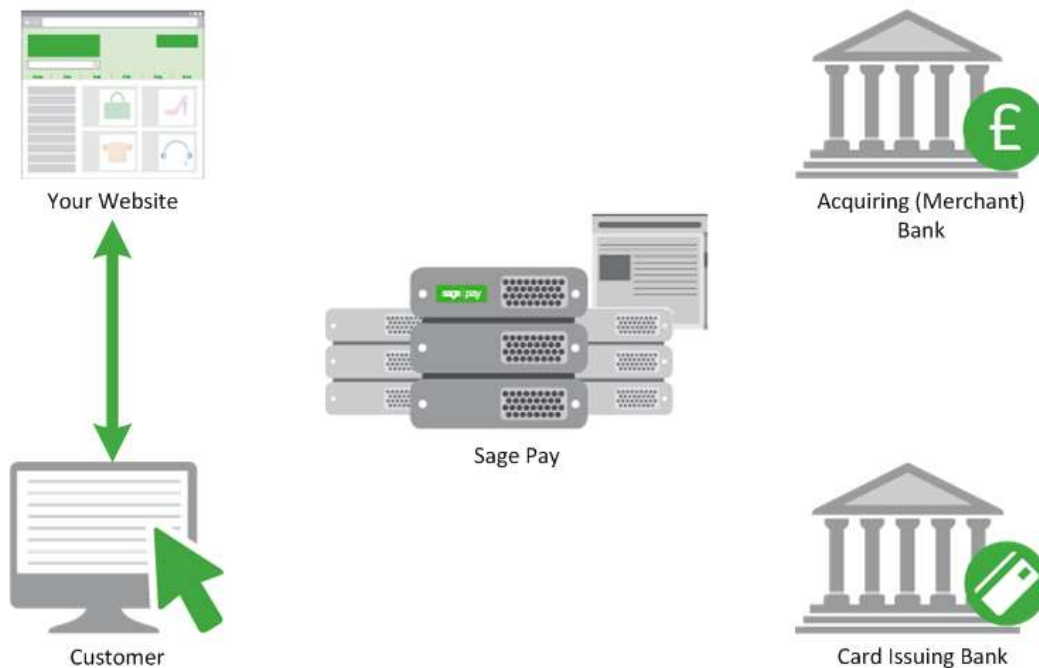
If you provide an email address for you and/or the customer in your post you have the flexibility to request that confirmation emails are sent in the event of a successful transaction. If the order fails, only you are emailed with a failure notification.

Sage Pay provides Integration Kits, which are simple worked examples in various different scripting languages that perform all the tasks described above. You simply customise these to work with your particular environment. These can be downloaded from sagepay.com.

The following sections explain the integration process in more detail. The protocol is attached in the Appendix providing a detailed breakdown of the contents of the encrypted fields sent between your servers and ours during a transaction.

3.0 Form Integration in Detail

Step 1: The customer orders from your site



A payment begins with the customer ordering goods or services from your website. This process can be as simple as selecting an item from a drop down list, or can involve a large shopping basket containing multiple items with discounts and delivery charges. Your interaction with your customer is entirely up to you and Form integration requires you to collect only a few compulsory pieces of information, which are detailed in the latter part of this guide.

It is generally a good idea to identify the customer by name, email address, delivery and billing address and telephone number. It is also helpful to have your server record the IP Address from which the user is accessing your system. You should store these details in your session alongside details of the customer's basket contents or other ordered goods.

You do not need to collect payment data, all your site needs to do is calculate the total cost of the order in whatever currency your site operates and present the user with a confirmation page, containing the transaction detail in an encrypted hidden field (see Step 2).

If you wish to apply a surcharge to a particular payment method/currency then this will be applied and shown on the subsequent payment pages.

Step 2: Your server builds a Confirmation Page

Your server-side script will build an order confirmation page, displaying the full details of the purchase to the customer, including their billing and delivery addresses, basket contents, total order value and contact details.

This script will also place an HTML FORM on that page with the action set to the Sage Pay Form registration page. That form will also contain four hidden fields:

VPSPProtocol – which lets our system know the version of our messages you are using (the current version is 3.00).

TxType – this lets us know which transaction type you wish to perform. In most cases this will be PAYMENT.

VendorName – your unique identifier, assigned to you by Sage Pay during sign up.

Crypt – a field containing encrypted and encoded details of the transaction. This prevents the customer from being able to tamper with the contents of the order before they are submitted to us. See Appendix A1.1 for details on how to construct the Crypt field.

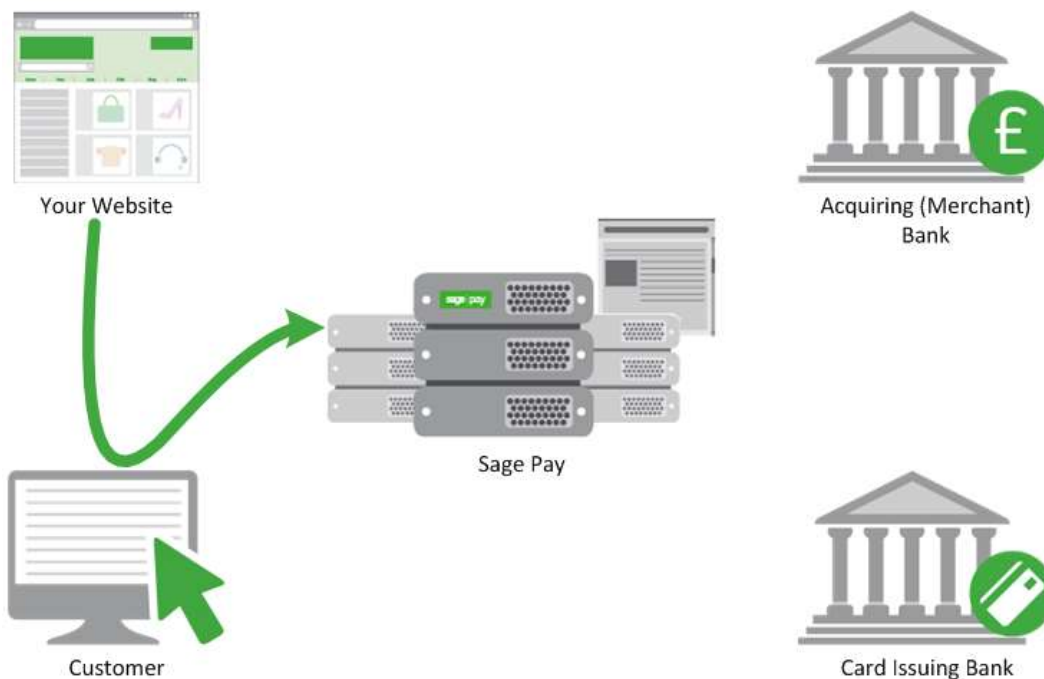
The contents of the crypt field are built by your script and include, amongst other things:

- A unique reference to this transaction that you generate (**VendorTxCode**).
- Total transaction value and currency.
- The URLs of the order Success and Failure pages.
- Customer email address for confirmation emails.
- Your email address for notification emails.
- Billing and Delivery addresses.
- Basket contents and a description of goods.

See Appendix A1.3 for the full protocol which lists all the fields you can send if you wish, and those which are compulsory for all transactions.

The integration kits we provide contain scripts in a variety of languages that illustrate how you compose and send this message from your server to ours. These can be downloaded from [sagepay.com](https://www.sagepay.com).

When the customer clicks the 'Pay Now' button on the form, the hidden fields are POSTed to Sage Pay and customer's browser is redirected to our hosted payment pages.



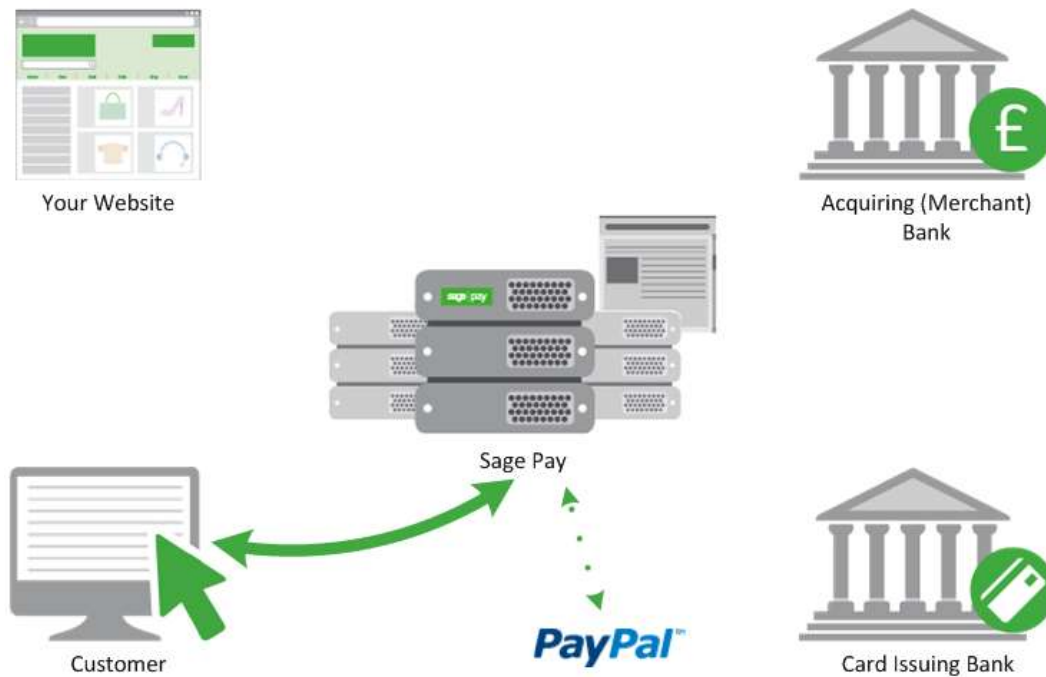
The Sage Pay system begins by validating the Crypt field contents. It first checks to ensure all the required fields are present, and that their format is correct. If any are not present or contain the wrong type of data, a validation error is sent to your failure page if possible, or displayed on screen.

This normally only happens in the development stage so your customers are unlikely to encounter this page.

If all fields are present and correct, the information in those fields is then validated. The `VendorName` is checked against our database and the `Currency` of the transaction is validated against those accepted by your merchant accounts. The `VendorTxCode` is checked to ensure it has not been used before, the `Basket/BasketXML` contents are validated to ensure they have been sent in the correct format and the `Amount` field is validated. Flag fields are checked... every field, in fact, is checked to ensure you have passed appropriate values.

If everything in the POST checks out, the transaction is registered with the Form system and a new transaction code is generated that is unique across ALL merchants using the Sage Pay gateway. This code, the `VPSTxId` (or Transaction ID), is our unique reference to the transaction, and is sent back to you at the transaction completion stage.

Step 3: Customer enters payment details on Sage Pay's server



The customer is presented with a page where they can select a payment type. If the customer selects a card type then their credit/debit card details are requested. If you are a certified PayPal Business account holder and you have activated PayPal on your Sage Pay account, the PayPal option will also be displayed to your shoppers on this page. Click [here](#) to view instructions on setting up PayPal.

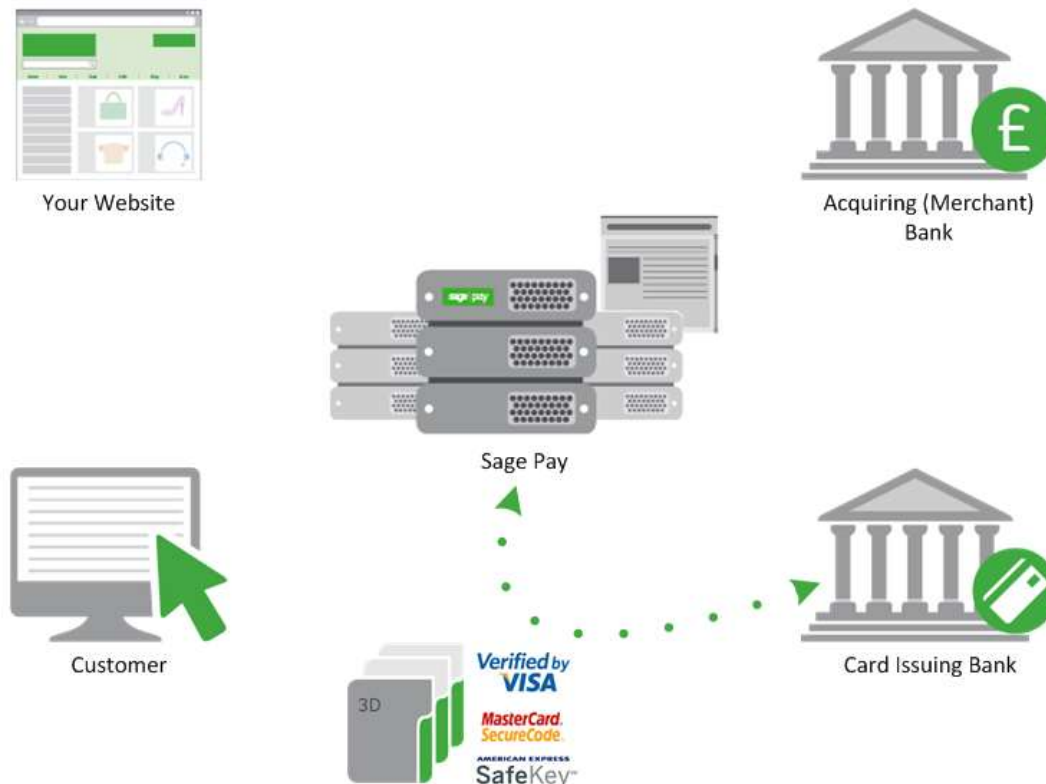
The payment type selection page will contain your company logo and the `Description` passed in Step 2. You can elect to customise these pages further by producing your own custom templates, the kit can be downloaded from [sagepay.com](https://www.sagepay.com). Please note; the most recent responsive designs are not yet customisable but you can continue to customise and use our older design.

Once the customer has entered their details, the Sage Pay Form system verifies that information prior to communicating with the bank. We ensure the card number is valid by performing a Luhn check (except for Maestro) and verification against our Issuer Identification Number database. We also check that the card type selected matches the card number and the expiry date is not in the past. If the customer selects PayPal on the card selection page, the customer is redirected to PayPal to select their payment method, before being returned to the Sage Pay order confirmation screen. Customers have the opportunity to supply an alternative address to the billing address details sent in your post. If you want to restrict this then you should apply our 'Address read-only' or 'No address' payment page templates via MySagePay.

If valid card details have been entered, the customer is presented with an order confirmation screen where they have one last chance to change their mind and cancel the transaction. If the customer decides to cancel, you will be sent a cancellation message to your `FailureURL` and the customer redirected there.

If your Sage Pay account is not set up with 3D-Secure or authentication is not active or applicable for this transaction the next step is for the system to obtain an authorisation (see Step 7).

Step 4: Sage Pay checks for 3D-Secure enrolment



The Sage Pay servers send the card details provided by your customer to the Sage Pay 3D-Secure Merchant Plug-In (MPI). This formats a verification request called a VEReq, which is sent to the 3D-Secure directory servers to query whether you, the merchant, and the card issuer are enrolled in the 3D-Secure scheme.

The 3D-Secure directory servers send a verification response called a VERes back to our MPI where it is decoded and the Sage Pay system is informed of the inclusion or exclusion of the card.

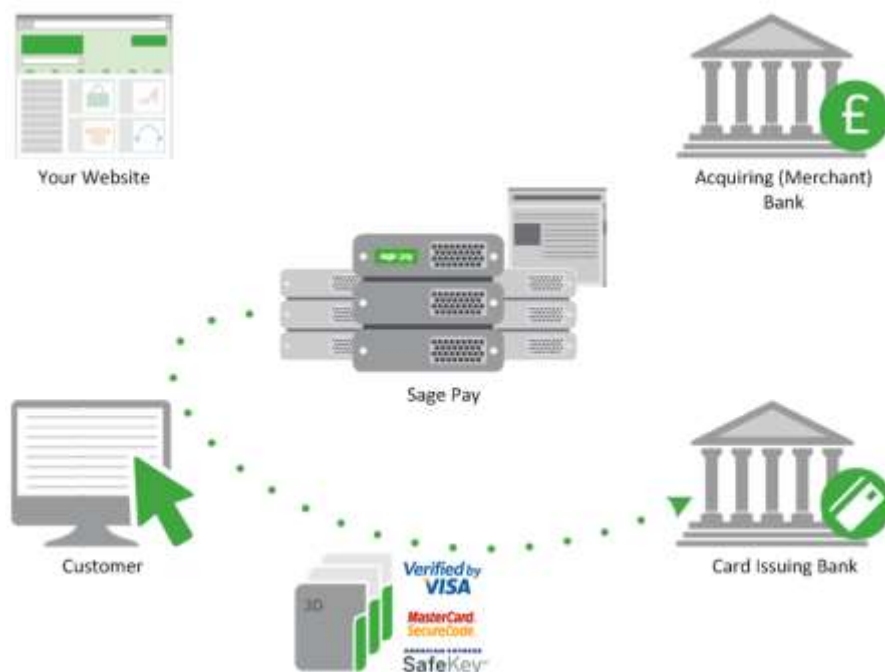
If the issuer is not part of the scheme, or if an MPI error occurs, our server will check your 3D-Secure rulebase to determine if authorisation should occur. By default, transactions that cannot be authenticated will be forwarded to your acquiring bank for authorisation.

If you do have a rulebase set up, our system check the rules you have in place to determine whether you wish the customer to proceed with authorisation, or you require them to select a different payment method. In such circumstances the shopper will be returned to the card selection page for another attempt. After the 3rd unsuccessful attempt, the customer will be redirected to your `FailureURL` with a `Status` of **REJECTED** and a `StatusDetail` indicating the reason for the failure. The `3DSecureStatus` field will contain the results of the authentication. **REJECTED** transactions will never be sent for settlement and the customer never charged, your failure page should explain why the transaction was aborted.

If your rulebase does allow authorisation to occur for cards not enrolled, the next step is for the system to obtain this from your acquirer (see Step 7).

In most cases 3D-Secure verification will be possible and the process continues in the next step.

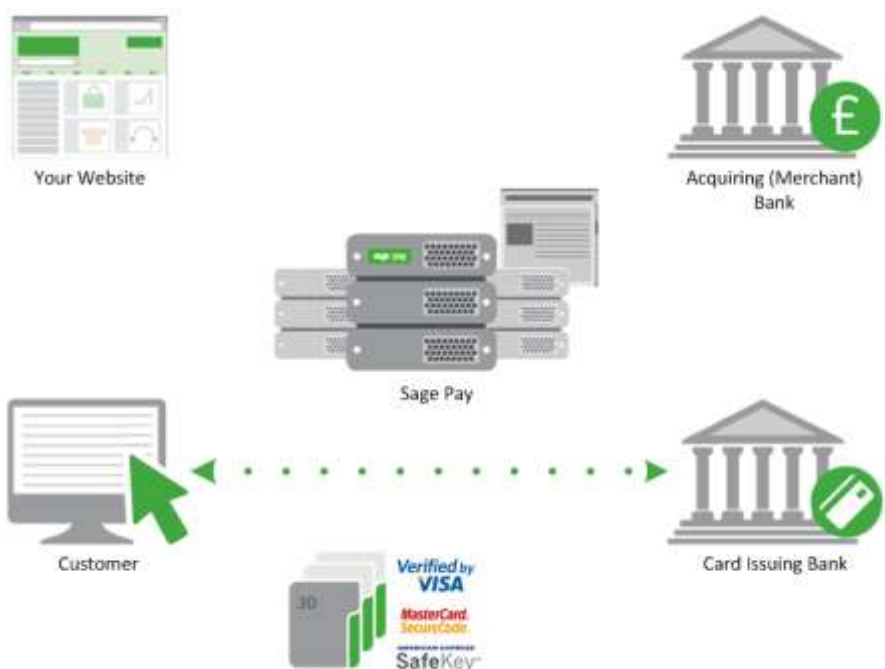
Step 5: Sage Pay redirects your customer to their Issuer



The customer's browser is redirected to their Card Issuing Bank's 3D-Secure authentication pages. These vary from bank to bank, but their purpose is to require the customer to authenticate themselves as the valid cardholder.

3D-Secure is much like an online version of Chip and Pin. The customer may be asked to answer questions at their card issuer's site (these might be a simple password,

characters from a password, or numbers generated via card devices, depending on the level of security employed by the bank) and in so doing, the bank is validating the customer's right to use the card for the transaction on your site. However, some issuers will automatically authenticate what they consider low risk transactions and not ask any questions.



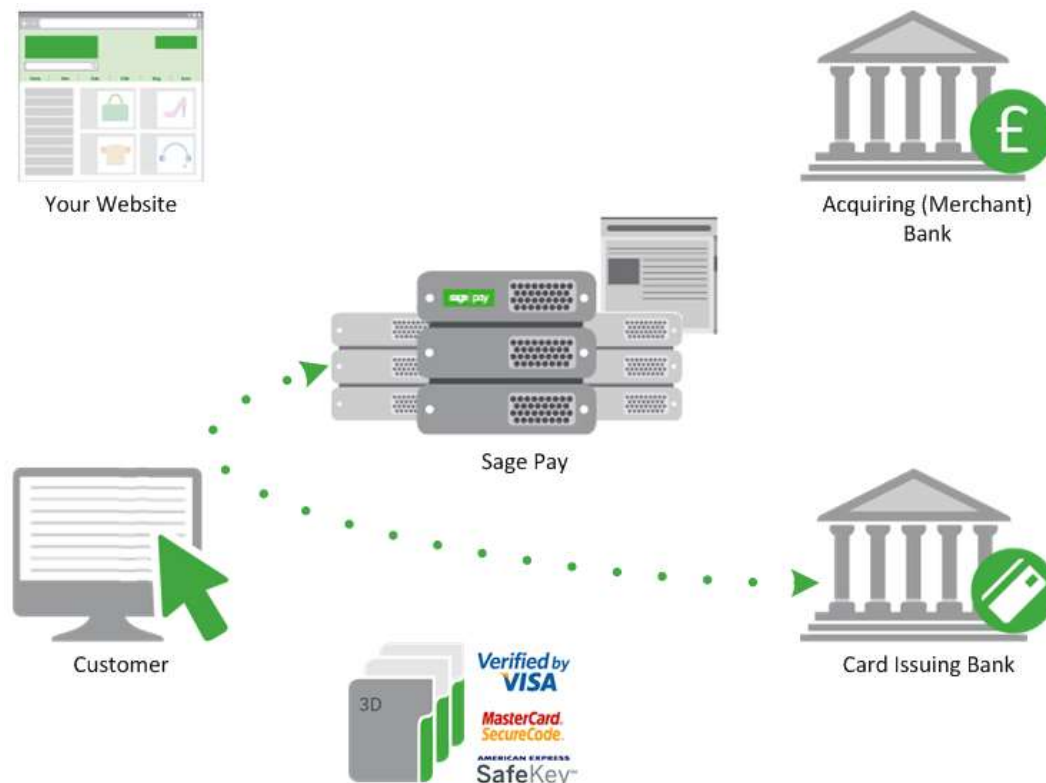
If they determine that the person attempting the transaction is the actual cardholder, they assume liability for the fraudulent use of that card during this transaction and you are protected from what are known as 'Chargebacks'.

Chargebacks occur when the cardholder subsequently challenges an authorisation with their issuing bank on the premise that it was obtained fraudulently. For

more information on chargebacks and the rules around liability shift, please contact your acquiring bank.

This level of protection for you is only afforded by 3D-Secure, which is why we recommend you enable this on your Sage Pay account. You can enable and specify rules for 3D-Secure in MySagePay.

Step 6: Issuing bank returns the customer to Sage Pay

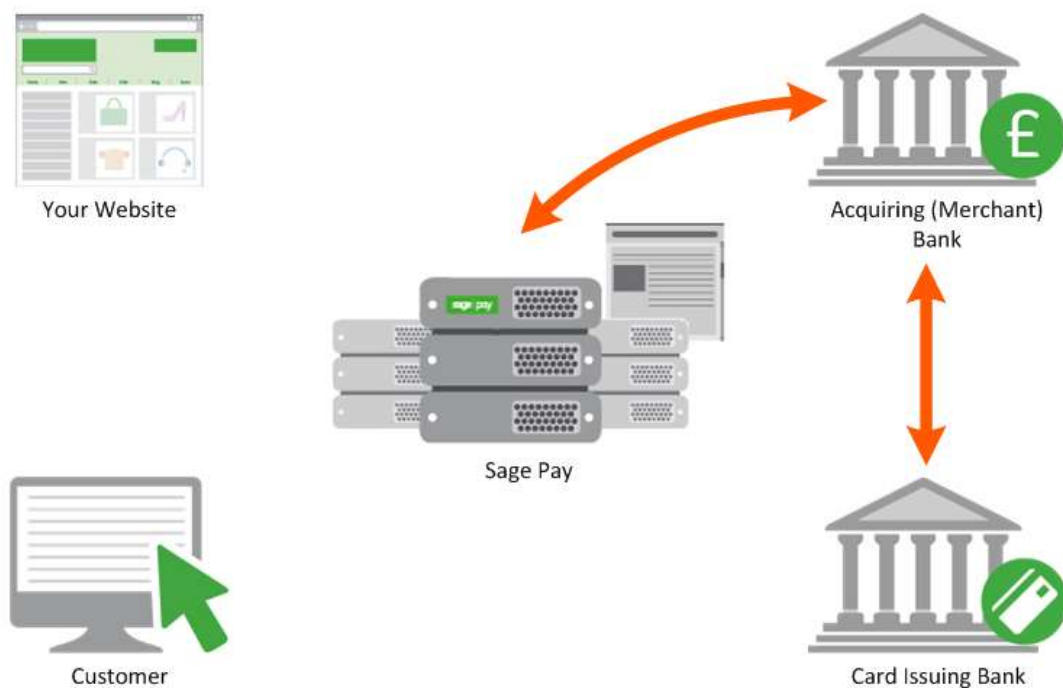


If the customer successfully completes authentication with their issuers, they are redirected to Sage Pay along with a unique authentication value (called CAVV for Visa, and UCAF for MasterCard). This is passed to your acquiring bank during authorisation to secure the liability shift for the transaction.

If the customer does not successfully authenticate with their issuing bank, they are passed back to the Sage Pay server without the CAVV/UCAF value. At this stage we consult your 3D-Secure rulebase to see if authorisation should be attempted. By default 3D-Authentication failures are not sent for authorisation. For more information on 3D-Secure and rulebases, please refer to our Fraud Prevention Guide available on [sagepay.com](https://www.sagepay.com).

If authorisation is not possible, your customer is returned to the card selection screen to choose an alternative payment method. After three failed attempts, the Sage Pay servers will redirect your customer to your `FailureURL` with a `Status` of **REJECTED** and a `StatusDetail` indicating the reason for the failure. Otherwise, authorisation will be gained from your acquiring bank.

Step 7: Sage Pay servers request card authorisation



The Sage Pay servers format a bank specific authorisation message (including any 3D-Secure authentication values where appropriate) and pass it to your merchant acquirer over the private banking network.

The request is normally answered within a second or so with either an authorisation code, or a declined message. This is obtained directly from the issuing bank by the acquiring bank in real time.

Whilst this communication is on-going, the customer is shown a page containing the text, “Please wait while your transaction is authorised with the bank”.

Sage Pay handles all authorisation failures by replying to your site with a **NOTAUTHED** message and a blank authorisation code after three failed attempts (the first two failures return the customer to the card selection screen to try another card).

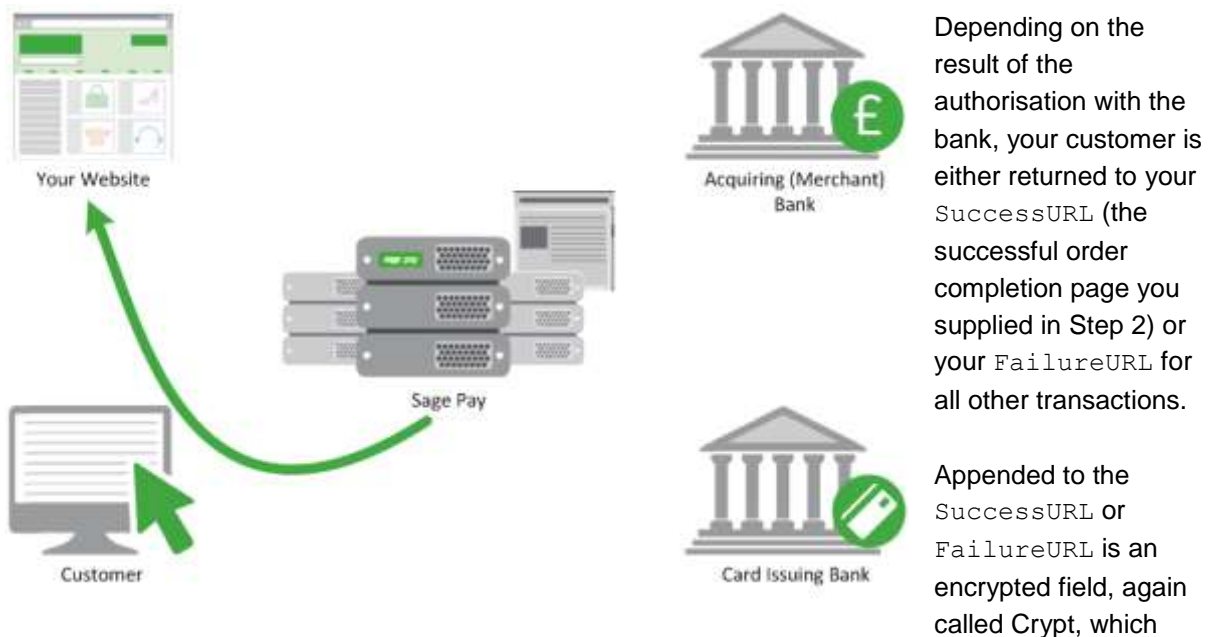
If the acquirer does return an authorisation code, Sage Pay prepares an **OK** response to send back to you in Step 8.

If AVS/CV2 fraud checks are being performed, the results are compared to any rulebases you have set up (refer to our Fraud Prevention Guide available on [sagepay.com](https://www.sagepay.com)). If the bank has authorised the transaction but the card has failed the fraud screening rules you have set, Sage Pay will immediately reverse the authorisation with the bank, requesting the shadow on the card for this transaction to be cleared, and prepares a **REJECTED** response.

Some card issuing banks may decline the reversal which can leave an authorisation shadow on the card for up to 10 working days. The transaction will never be settled by Sage Pay and will appear

as a failed transaction in MySagePay, however it may appear to the customer that the funds have been taken until their bank clears the shadow automatically after a period of time dictated by them.

Step 8: Sage Pay redirects the customer to your website



contains the `Status` of the transaction, the reference code for those transactions and the fraud checking results. The field is decoded in the same manner that your original script was encoded, using the same password (which is known only to you). The contents of the `Crypt` field are detailed in Appendix B1.

The `Status` field holds either:

- **OK** if the transaction was authorised at Step 7.
- **NOTAUTHED** if the authorisation was failed by the bank.
- **ABORT** if the user decided to click cancel whilst on the Sage Pay payment pages.
- **REJECTED** if authorisation occurred but your fraud screening rules were not met, or 3D-Authetnication failed three times.
- **ERROR** if an error has occurred at Sage Pay (these are very infrequent, but your site should handle them anyway. They normally indicate a problem with authorisation).

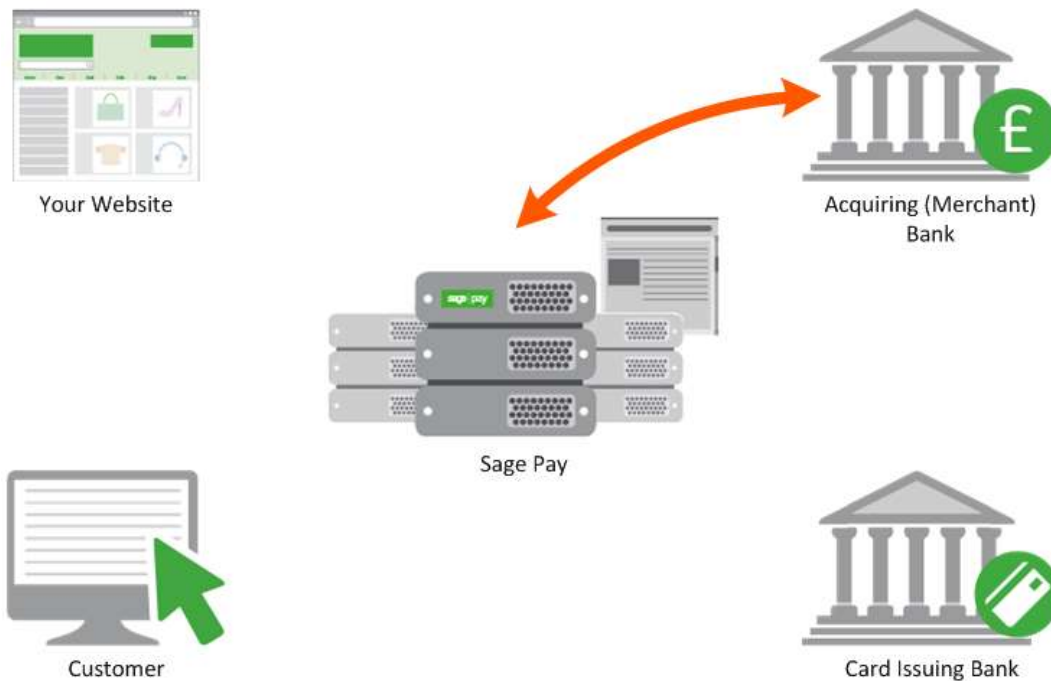
The `StatusDetail` field contains human readable description of the error message.

You may wish to display some of the information contained in the `Crypt` field to your customer, especially the reason for failure. You are not required to store any of the information sent to you in a database, but if you have access to one, you may wish to do so.

You will receive an email if you supplied a `VendorEmail` address with all these detail in, plus details of the order and the customer who placed it. Sage Pay cannot guarantee that the email will always arrive in a timely manner since we have no control over what happens once it leaves our servers. You should not rely solely on email confirmations, but regularly check MySagePay for new orders.

The real time processing of the transaction by Sage Pay is now complete. Later in the day the final stage of the process is carried out between us and the banks without you or your site needing to do anything.

Step 9: Sage Pay sends Settlement Batch Files



Once per day, from 12.01am, the Sage Pay system batches all authorised transactions for each acquirer and creates an acquirer specific settlement file.

Transactions for ALL merchants who use the same merchant acquirer are included in this file. Every transaction (excluding PayPal transactions*) that occurred from 00:00:00am until 11:59:59pm on the previous day, are included in the files.

They are uploaded directly to the acquiring banks on a private secure connection. This process requires no input from you or your site. The contents of these batches and confirmation of their delivery can be found in the Settlement section of MySagePay.

Sage Pay monitors these processes to ensure files are submitted successfully, and if not, the support department correct the problem to ensure the file is sent correctly that evening or as soon as reasonably possible. Ensuring funds are available to all vendors more expediently.

The acquirers send summary information back to Sage Pay to confirm receipt of the file, then later more detailed information about rejections or errors. If transactions are rejected, we will contact you to make you aware and where possible, resubmit them for settlement.

* Funds from your customers' PayPal payments are deposited into your PayPal Business account immediately, there is no settlement process. You can then withdraw or transfer the funds electronically into your specified bank account. Although PayPal transactions are included in the Settlement Reports displayed within MySagePay, as PayPal transactions are not settled by Sage Pay directly with the banks, we recommend you to log into your PayPal Admin area to obtain a report of your PayPal transactions.

4.0 Integrating with Sage Pay Form

Linking your website to Sage Pay using the Form integration method involves creating one script (or modifying the example provided in the integration kits), and two completion pages, one for successful transactions, the other for failures.

Stage 1

The first step of the integration will be to get your site talking to Sage Pay's Test server and process all possible outcomes. This is an exact copy of the live site but without the banks attached and with a simulated 3D-Secure environment. Authorisations on the test server are only simulated, but the user experience is identical to Live, MySagePay also runs here so you can familiarise yourself with the features available to you.

The MySagePay admin system for viewing your Test transactions is at:

<https://test.sagepay.com/mysagepay>

Transactions from your scripts should be sent to the Sage Pay Test Server at:

<https://test.sagepay.com/gateway/service/vspform-register.vsp>

Stage 2

Once you are happily processing end-to-end transactions on the test server and we can see test payments and refunds going through your account, you've completed the online Direct Debit signup and the MID has been confirmed by your Acquirer, your account on the Live Server is activated for you to start using. You will need to modify your scripts to send transactions to the live server, send through a Payment using your own credit or debit card, and then VOID it through the MySagePay Admin service so you don't charge yourself. If this works successfully, then you are ready to trade online.

The Live MySagePay admin system is at:

<https://live.sagepay.com/mysagepay>

Transactions from your scripts should be sent to the Sage Pay Live Server at:

<https://live.sagepay.com/gateway/service/vspform-register.vsp>

5.0 Testing on the Test Server (Stage 1)

The Test Server is an exact copy of the Live System but without the banks attached and with a simulated 3D-Secure environment. This means you get a true user experience but without the fear of any funds transferring during testing.

In order to test on the Test Server, you need a Test Server account to be set up for you by the Sage Pay Support team. Your test account can only be set up once you have submitted your Sage Pay application. You can apply online [here](#). Often when applying to trade online it takes a while for the Merchant Account to be assigned by your acquirer, so you may wish to ensure that you set those wheels in motion before you begin your integration with Sage Pay, to ensure things don't bottleneck at this stage.

The Support Team will set up an account for you on the Test Server within 48 hours of you submitting a completed application form. This will be under the same Sage Pay Vendor Name as your online application form. You will, however, be issued with different passwords for security purposes. The Support Team will let you know how to retrieve those passwords and from there how to use the MySagePay screens to look at your transactions.

To link your site to the Test Server, you need only to change your transaction registration script to send the message to the Test Server URL for Sage Pay Form. In the kits this is done simply by changing the flag in the configuration scripts to TEST. If you've been developing your own scripts, then the Test Site URL for payment registration is:

<https://test.sagepay.com/gateway/service/vspform-register.vsp>

5.1 Registering a Payment

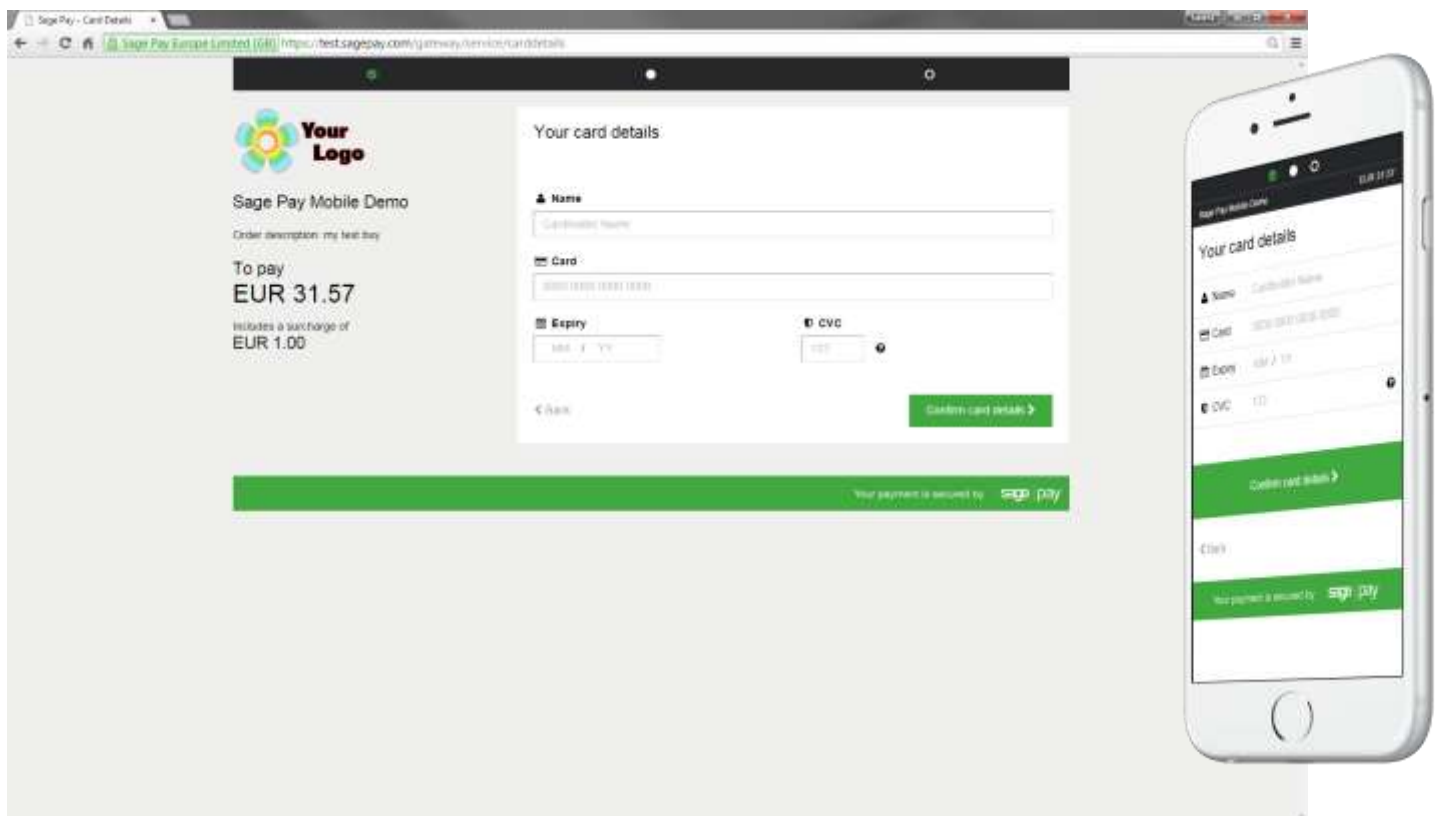
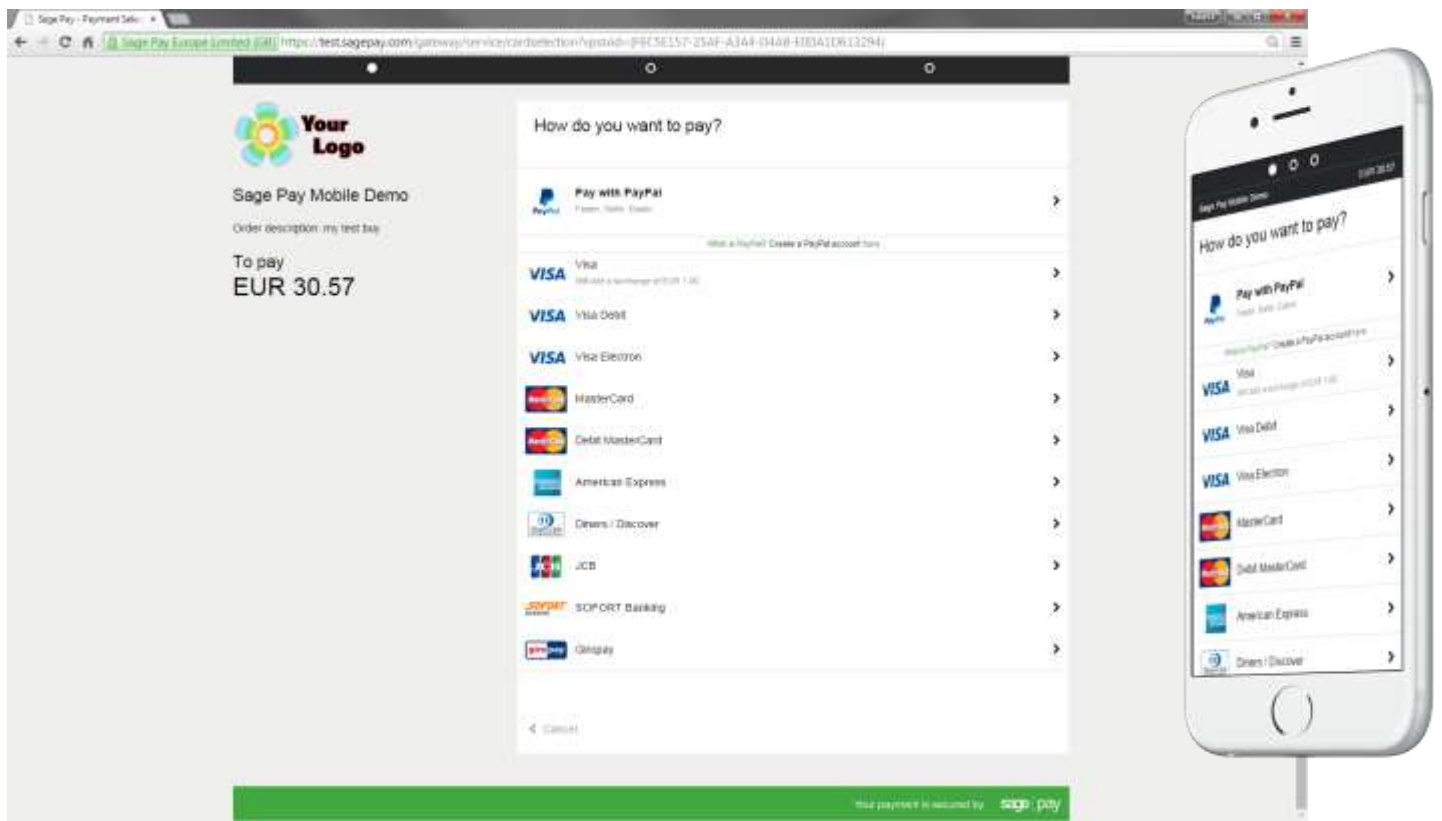
If you don't plan to implement the protocol entirely on your own, you should install the most appropriate integration kit or worked example for your platform. These can be downloaded from sagepay.com.

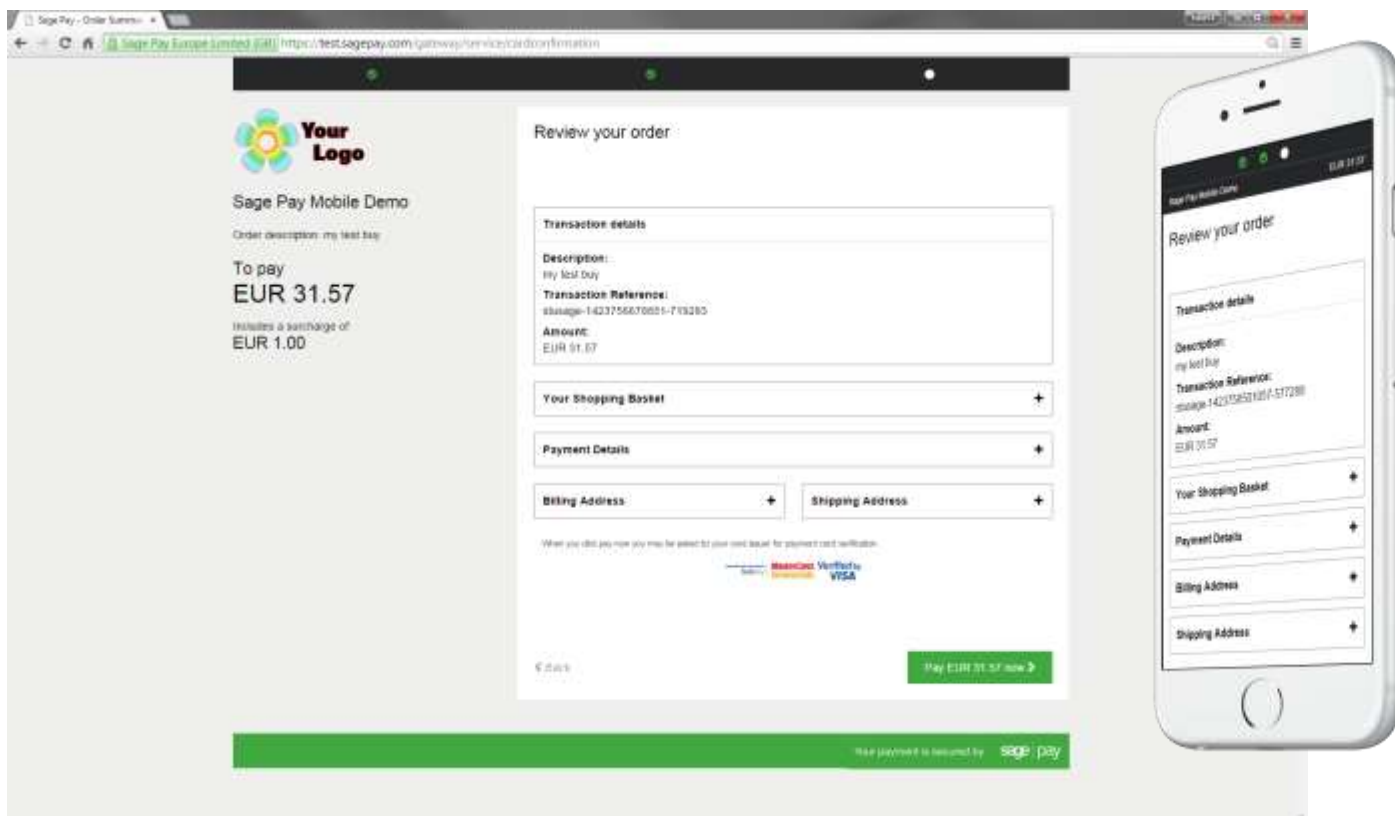
The kits will not quite run out of the box because you have to provide some specific details about your site in the configuration files before a transaction can occur, but they will provide end to end examples of registering the transactions and handling the success and failure redirects.

This script provides a worked example of how to construct the `Crypt` field that Sage Pay Form needs to initiate the payment process (see Appendix A1.1).

Check that this script is sending transactions to the Sage Pay test server, and then execute this page passing dummy values in applicable fields

You will first be presented with our responsive Card Selection page, where you select your payment method, then the Card Details and Order Summary.





5.1.1 Test card numbers

You will always receive an **OK** response and an Authorisation Code from the test server if you are using one of the test cards listed below. All other valid card numbers will be declined, allowing you to test your failure pages.

If you do not use the Address, Postcode and Security Code listed below, the transaction will still authorise, but you will receive NOTMATCHED messages in the AVS/CV2 checks, allowing you to test your rulebases and fraud specific code.

There are different cards for Visa and MasterCard to simulate the possible 3D-Secure responses.

Billing Address 1: 88

Billing Post Code: 412

Security Code: 123

Valid From: Any date in the past

Expiry Date: Any date in the future

Payment Method	Card Number	CardType Response	3D-Secure Response (VERes)
Visa	4929 0000 0000 6	VISA	Y
Visa	4929 0000 0555 9	VISA	N
Visa	4929 0000 0001 4	VISA	U
Visa	4929 0000 0002 2	VISA	E
Visa Corporate	4484 0000 0000 2	VISA	N
Visa Debit	4462 0000 0000 0003	DELTA	Y
Visa Electron	4917 3000 0000 0008	UKE	Y
MasterCard	5404 0000 0000 0001	MC	Y
MasterCard	5404 0000 0000 0043	MC	N
MasterCard	5404 0000 0000 0084	MC	U
MasterCard	5404 0000 0000 0068	MC	E
Debit MasterCard	5573 4700 0000 0001	MCDEBIT	Y
Maestro (UK Issued)	6759 0000 0000 5	MAESTRO	Y
Maestro (German Issued)	6705 0000 0000 8	MAESTRO	Y
Maestro (Irish Issued)	6777 0000 0000 7	MAESTRO	Y
Maestro (Spanish Issued)	6766 0000 0000 0	MAESTRO	Y
American Express	3742 0000 0000 004	AMEX	N/A
Diners Club / Discover	3600 0000 0000 08	DC	N/A
JCB	3569 9900 0000 0009	JCB	N/A
PayPal	Use your own PayPal Sandbox	PAYPAL	N/A

3D-Secure Response (VERes)

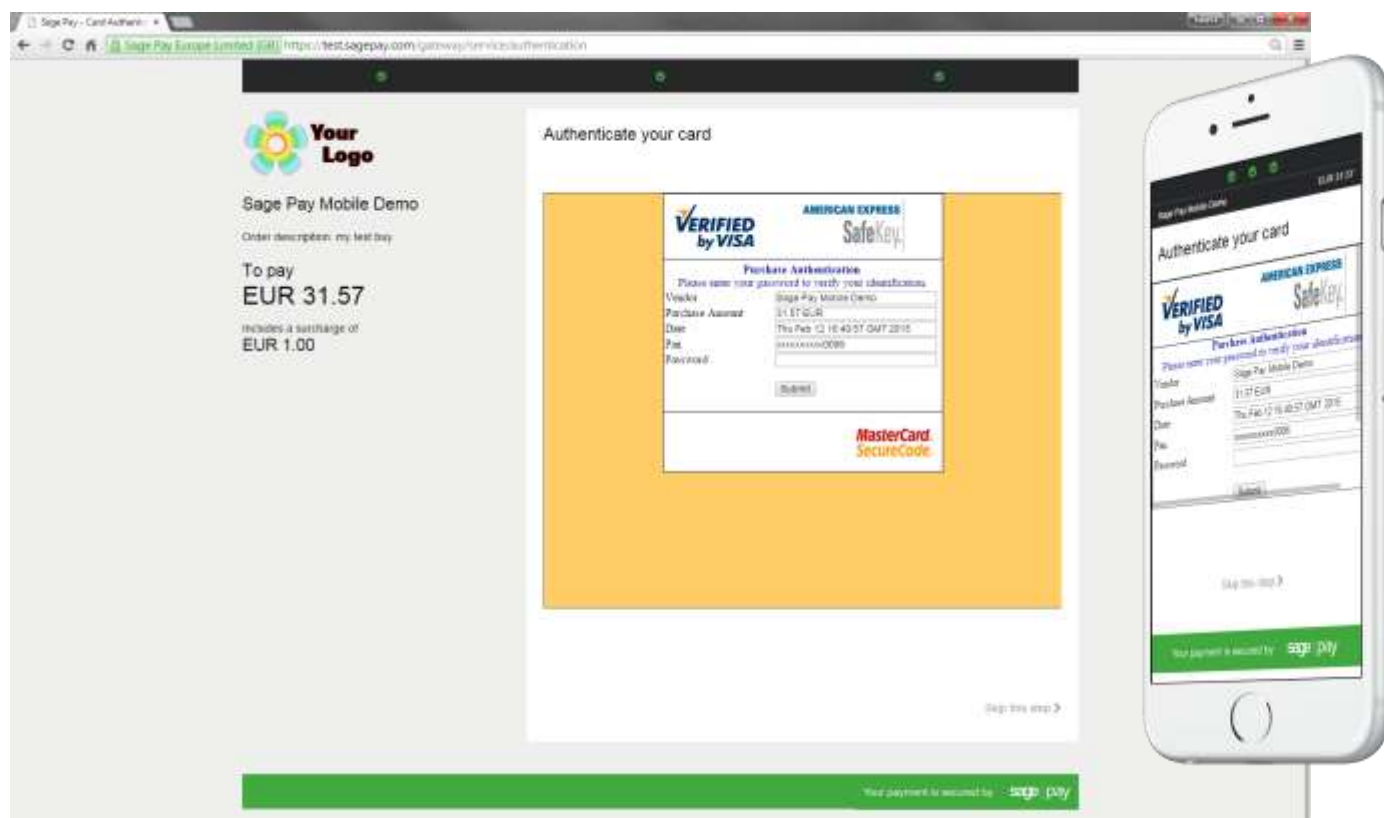
Y = Enrolled, will progress to PAReq (3D-Authentication)

N = Not Enrolled, will return the 3DSecureStatus **NOTAVAILABLE**

U = Unable to verify enrolment, will return the 3DSecureStatus **NOTAVAILABLE**

E = Error occurred during verification, will return the 3DSecureStatus **ERROR**

If you have 3D-Secure set up on your test account, you can use MySagePay to switch on the checks at this stage and simulate the Verification and Authentication process.



To successfully authenticate the transaction, enter “**password**” (without the quotes) into the password field. Enter the values below (without the quotes) into the password field to simulate all other possible 3D-Secure responses:

“**A:D:06**” = Cardholder not enrolled, will return the 3DSecureStatus **ATTEMPTONLY**

“**U:N:06**” = Authentication not available, will return the 3DSecureStatus **INCOMPLETE**

“**E:N:06**” = Error occurred during Authentication, will return the 3DSecureStatus **ERROR**

Any other phrase will fail the authentication, allowing you to test your rules and 3D-Secure response handling.

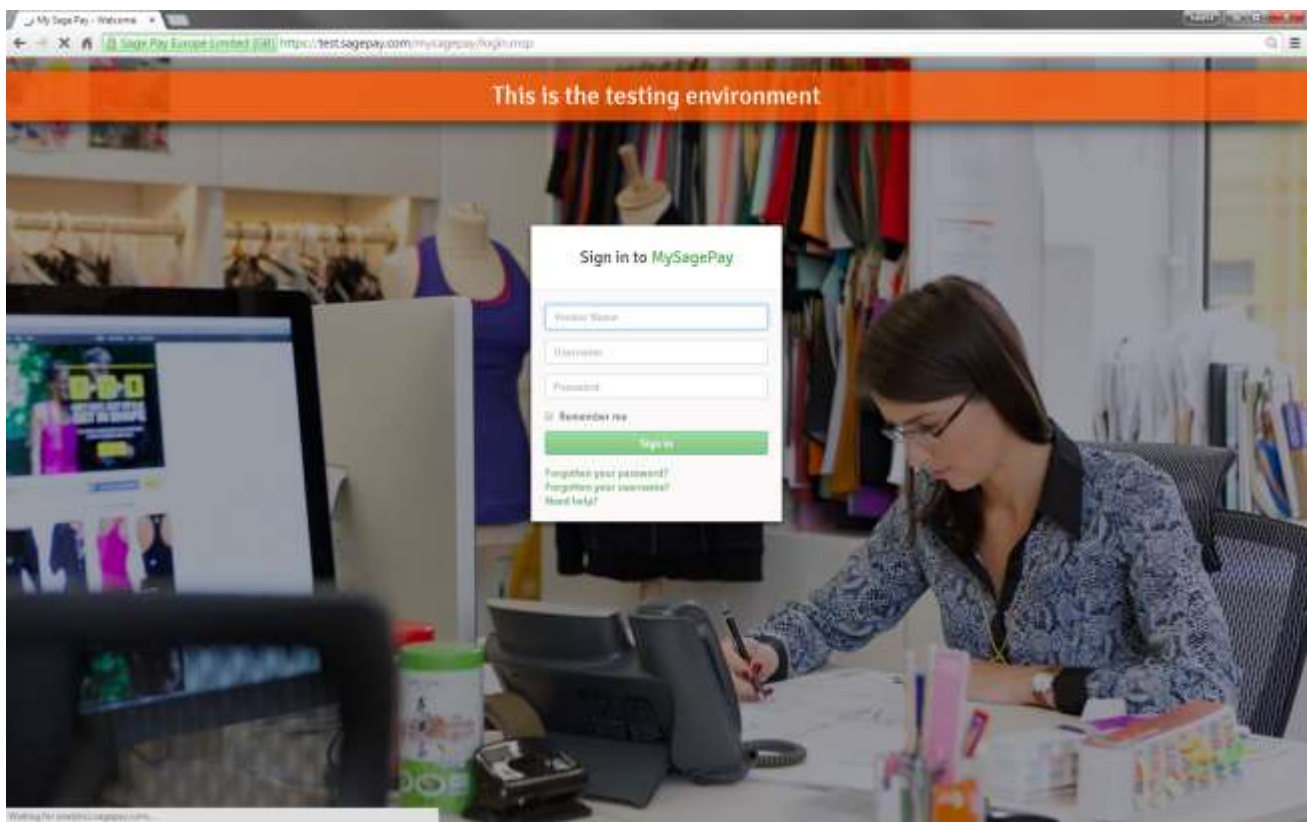
The process will then continue as per the Live Servers. Only the authorisation stage is simulated.

Once you’ve checked you can process a transaction then you are almost ready to go live. Before doing so, you must log in to MySagePay on the test server, create a user and refund a transaction.

5.2 Accessing MySagePay on Test

A Test Server version of MySagePay is available to you whilst using your test account to view your transactions, refund payments, release deferred payments, void transactions etc. You should familiarise yourself with this system on the Test Server before you go live so you know how to use the system on the Live Servers. The user guide for MySagePay can be found [here](#).

The Test Server MySagePay can be found at: <https://test.sagepay.com/mysagepay>



When you log in to MySagePay screens you will be asked for a Vendor Name, a Username and a Password. The first time you log in you will need to do so as your system Administrator:

- In the Vendor Name field, enter your Vendor Name, set during the application process used throughout the development as your unique Sage Pay identifier.
- In the Username field, enter the Vendor Name again.
- In the Password field, enter the MySagePay Admin password as supplied to you by Sage Pay when your test account was set up.

The administrator can ONLY access the settings Tab. You cannot, whilst logged in as administrator, view your transactions or take MO/TO payments through the online terminal.

To use those functions, and to protect the administrator account, you need to create new users for yourself and others by clicking on the 'Users' tab then the 'New User' button. You will be presented the following screen where you set the log in credentials and account privileges.

Add new user

Username:

*

x

First name:

Last name:

Email address:

Confirm email address:

Receive updates and communications:

☐

Enter password:

*

Confirm password:

*

Password Strength:

The minimum password length required is 8 characters
To improve security on your account we recommend a strong password that contains at least one uppercase letter (A-Z), one lowercase letter (a-z), one number (0-9) and one special character (^\$.?+:%-_=~!@#;).

Account Privileges

☐ View All transactions

☐ REFUND transactions

☐ RELEASE and AUTHORISE transactions

☐ ABORT and CANCEL transactions

☐ VOID transactions

☐ REPEAT or REPEATDEFERRED transactions

☐ MANUAL transactions via the Terminal screens

My Sage Pay Access

☒ Search

☐ Transactions

☐ Settings (Admin settings)

☐ Terminal

Default Landing Page

☒ Search

☐ Transactions

☐ Settings

☐ Terminal

Add User

Once you have created a new user, click the Sign Out button and sign back in, this time entering:

- Your Vendor name in the Vendor Name field.
- The Username of the account you just created in the Username field.
- The password for the account you just created in the Password field.

You are now logged in using your own account and can view your test transactions and use all additional functions. If you lock yourself out of your own account, you can use the Administrator account to unlock yourself or use the lost password link on the Sign In screen.

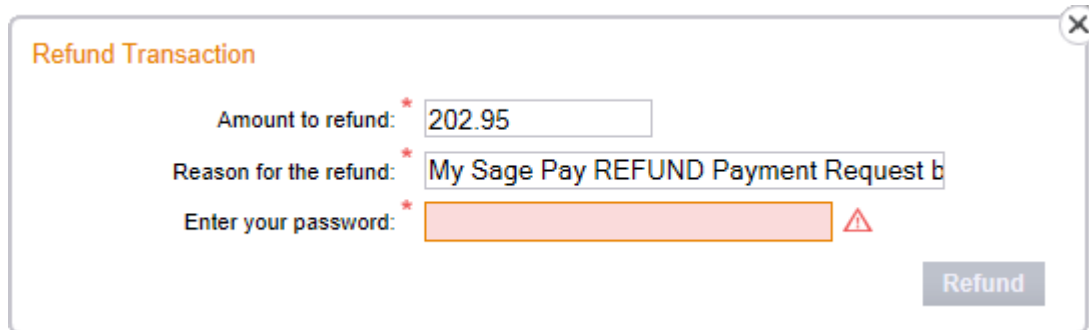
If you happen to lock out the Administrator account, you will need to contact Sage Pay to unlock it for you. Send an email to unlock@sagepay.com stating the Vendor Name and Merchant Number of the account. If you need reminding of your unique account passwords, send an email to the above and request a password retrieval link, stating the Vendor Name and Merchant Number of the account.

Detailed information on using MySagePay can be found [here](#). Play with the system until you are comfortable with it. You cannot inadvertently charge anyone or damage anything whilst on the test server.

5.3 Refunding a transaction

Before we can set your account live, you will need to refund one of the test transactions you have already performed. Whilst signed in to MySagePay as a user which has privileges to refund a transaction, select the Transactions tab.

Click a successful transaction and then the 'Refund' button.

A screenshot of a 'Refund Transaction' form. The form has a title 'Refund Transaction' in orange at the top left. It contains three input fields: 'Amount to refund:' with a red asterisk and the value '202.95'; 'Reason for the refund:' with a red asterisk and the text 'My Sage Pay REFUND Payment Request b'; and 'Enter your password:' with a red asterisk and a red rectangular box. To the right of the password box is a red warning triangle icon. A 'Refund' button is located at the bottom right of the form. The form has a close button (X) in the top right corner.

You will be prompted with a screen to enter your password. You also have the opportunity to set a description for the refund and modify the amount. You cannot refund for more than the original amount.

MySagePay is also available on mobile devices.

The following features are currently available:

- List of transactions (including status)
- Transaction details
- Account activity monitoring
- Sage Pay news and alert notifications
- Sage Pay support access



6.0 Additional Transaction Types

Sage Pay supports a number of additional methods of registering a transaction and completing the payment.

6.1 DEFERRED transactions

By default a **PAYMENT** transaction type is used to gain an authorisation from the bank, and then settle that transaction early the following morning, committing the funds to be taken from your customer's card.

In some cases you may not wish to take the funds from the card immediately, merely place a 'shadow' on the customer's card to ensure they cannot subsequently spend those funds elsewhere. Then take the money when you are ready to ship the goods. This type of transaction is called a **DEFERRED** transaction and is registered in exactly the same way as a **PAYMENT**. You simply need to change your script to send a TxType of **DEFERRED** when you register the transaction instead of **PAYMENT**.

DEFERRED transactions are not sent to the bank for completion the following morning. In fact, they are not sent at all until you **RELEASE** them through MySagePay. You can release only once and only for an amount up to and including the amount of the original **DEFERRED** transaction.

If you are unable to fulfil the order, you can also **ABORT** deferred transactions in a similar manner and the customer will never be charged.

DEFERRED transactions work well in situations where it is only a matter of days between the customer ordering and you being ready to ship. Ideally all **DEFERRED** transaction should be released within 6 days. After that the shadow may disappear from the cardholders account before you settle the transaction, and you will have no guarantee that you'll receive the funds if the customer has spent all available funds in the meantime.

If you regularly require longer than 6 days to fulfil orders, you should consider using Authenticate and Authorise instead of **DEFERRED** payments.

DEFERRED transactions remain available for **RELEASE** for up to 30 days. After that time they are automatically **ABORTed** by the Sage Pay system.

As settlement is not guaranteed to occur within 4 days for this transaction type, you may be charged a higher fee by your acquirer for ALL Deferred transactions. You should contact your Merchant Bank for more information on Pre-Authorisations.



Unlike a normal Sage Pay **DEFERRED** transaction, no shadow is placed on the customer's account for a PayPal **DEFERRED** transaction. An order is simply registered with the PayPal account and a successful authorisation for a **DEFERRED** transaction only confirms the availability of funds and does not place any funds on hold.

When you **RELEASE** a **DEFERRED** PayPal transaction, PayPal applies best efforts to capture funds at that time, but there is a possibility that funds will not be available. We recommend that you do not ship goods until obtaining a successful release.

6.2 REPEAT payments

If you have already successfully authorised a **PAYMENT**, a released **DEFERRED** or an **AUTHORISE** you can charge an additional amount to that card using the **REPEAT** transaction type, without the need to store the card details yourself.

If you wish to regularly **REPEAT** payments, for example for monthly subscriptions, you should ensure you have a merchant number from your bank that supports this recurring functionality (sometimes called Continuous Authority). **REPEAT** payments cannot be 3D-Secured nor have CV2 checks performed on them unless you supply this value again, as Sage Pay are not authorised to store CV2 numbers. It may be better to make use of Authenticate and Authorise if you need to vary the transaction amount on a regular basis.

The Sage Pay gateway archives all transactions that are older than 2 years old; this prevents any subsequent authorisations from being made. We therefore recommend that you repeat against the last successful authorised transaction.



At the moment it is not possible to **REPEAT** a PayPal transaction when using Form integration.

6.3 AUTHENTICATE and AUTHORISE

The **AUTHENTICATE** and **AUTHORISE** methods are specifically for use by merchants who are either:

- Unable to fulfil the majority of orders in less than 6 days or sometimes fulfil them after 30 days.
- Do not know the exact amount of the transaction at the time the order is placed, for example; items shipped priced by weight or items affected by foreign exchange rates.

Unlike normal **PAYMENT** or **DEFERRED** transactions, **AUTHENTICATE** transactions do not obtain an authorisation at the time the order is placed. Instead the card and cardholder are validated using the 3D-Secure mechanism provided by the card-schemes and card issuing banks, with a view to later authorise.

Your site will register the transaction with a TxType of **AUTHENTICATE**, and the customer will be redirected to the Sage Pay payment pages to enter their payment details. Sage Pay will verify the card number and contact the 3D-Secure directories to check if the card is part of the scheme. If it is not, the card details are simply held safely at Sage Pay and your customer returned to your SuccessURL with a Status of **REGISTERED**. This also happens if you do not have 3D-Secure active on your account or have used the Apply3DSecure flag to turn it off.

If they have not passed authentication, your rule base is consulted to check if they can proceed for authorisation anyway. If not, your customer is returned to your FailureURL with a Status of **REJECTED**. If they failed authentication but can proceed, your customer is returned to your SuccessURL with a Status of **REGISTERED**. If the user passed authentication with their bank and a CAVV/UCAF value is returned, the customer is returned to your SuccessURL a Status of **AUTHENTICATED**.

In all cases, the customer's card is never authorised. There are no shadows placed on their account and your acquiring bank is not contacted. The customer's card details and their associated authentication status are simply held at Sage Pay for up to 90 days (a limit set by the card schemes, 30 days for International Maestro cards) awaiting you to **AUTHORISE** or **CANCEL** via MySagePay.

To charge the customer when you are ready to fulfil the order, you will need to **AUTHORISE** the transaction through MySagePay. You can authorise for any amount up to 115% of the value of the original Authentication, and use any number of Authorise requests against an original Authentication. As long as the total value of those authorisations does not exceed the 115% limit and the requests are inside the 90 days limit the transactions will be processed by Sage Pay. This is the stage at which your acquiring bank is contacted for an authorisation code. AVS/CV2 checks are performed at this stage and rules applied as normal. This allows you greater flexibility for partial shipments or variable purchase values. If the **AUTHENTICATE** transaction was **AUTHENTICATED** (as opposed to simply **REGISTERED**) all authorisations will be fully 3D-Secured.

When you have completed all your Authorisations, or if you do not wish to take any, you can **CANCEL** the **AUTHENTICATE** in MySagePay, prevent any further Authorisations being made against the card. This happens automatically after 90 days.



You can use the Authenticate and Authorise transaction type but the transaction will only ever be **REGISTERED** (because the transaction will never be 3D-Secured).

6.4 REFUNDS and VOIDS

Once a **PAYMENT**, **AUTHORISE** or **REPEAT** transaction has been **AUTHORISED**, or a **DEFERRED** transaction has been **RELEASED**, it will be settled with the acquiring bank early the next morning and the funds will be moved from the customer's card account to your merchant account. The bank will charge you for this process, the exact amount depending on the type of card and the details of your merchant agreement.

If you wish to cancel that payment before it is settled with the bank the following morning, you can **VOID** a transaction in MySagePay to prevent it from ever being settled, thus saving you your transaction charges and the customer from ever being charged. **VOIDed** transactions can NEVER be reactivated, so use this functionality carefully.

Once a transaction has been settled you can no longer **VOID** it. If you wish to return funds to the customer you need to perform a **REFUND** in MySagePay.

You can **REFUND** any amount up to the value of the original transaction. You can even send multiple refunds for the same transaction so long as the total value of those refunds does not exceed the value of the original transaction.

The Sage Pay gateway archives all transactions that are older than 2 years old; we therefore recommend that you check the date of the original transaction which you wish to refund before processing.



You cannot **VOID** a PayPal transaction, but you are able to **REFUND** a PayPal transaction.

7.0 Applying Surcharges

The ability to apply surcharges based on the currency and payment type selected will provide a financial benefit to you by transferring the cost of these transactions to the customer.

You will have the ability to pass surcharge values (fixed amount or percentage) for all transactions except PayPal. For example, credit card = fixed fee of £2.00 or 2%.

Different surcharges can be set for each payment type/currency combination you accept.

Please note it is your responsibility to ensure that any surcharges set up comply with laws within your country.

How does it work

- You set up default surcharges for the payment types/currencies you wish to apply them to in MySagePay.
- Customers select the goods they wish to purchase from your website.
- They then select the payment type to complete the transaction.
- Alternatively you can use the `SurchargeXML` (see Appendix A1.4) to send through surcharge values that override the defaults. If the payment type selected is not sent through in the `SurchargeXML` then the default in MySagePay will be applied.

For more information, please contact our support team on support@sagepay.com

8.0 Sage 50 Accounts Software Integration

It is possible to integrate your Sage Pay account with Sage Accounting products to ensure you can reconcile the transactions on your account within your financial software.

To learn more about the integration options available and which version of Sage Accounts integrate with Sage Pay please visit sagepay.com, or email tellmemore@sagepay.com.

If you wish to link a transaction to a specific product record this can be done through the `Basket` field in the transaction registration post.

Please note the following integration is not currently available when using `BasketXML` fields.

In order for the download of transactions to affect a product record the first entry in a basket line needs to be the product code of the item within square brackets.

Example;

```
4:[PR001]Pioneer NSDV99 DVD-Surround Sound System:1:424.68:74.32:499.00:
499.00:[PR002]Donnie Darko Director's Cut:3:11.91:2.08:13.99:41.97:[PR003]Finding
Nemo:2:11.05:1.94:12.99:25.98: Delivery:000:000:000:000:4.99
```

When a transaction with the `Basket` field containing the items above is imported into Sage 50 Accounts an invoice is created and product codes PR001, PR002 and PR003 are updated with the relevant activity and stock levels reduced accordingly.

For further information on the `Basket` field please see Appendix A1.5.

9.0 Going Live (Stage 2)

Once Sage Pay receives your application your account will be created and details will be sent to the bank for confirmation. The bank will be expected to confirm your merchant details within 3 to 5 working days. Once both the Direct Debit (filled out during application) and the confirmation of your merchant details reach Sage Pay, your account will become Live automatically and you will start to be billed for using our gateway.

This does not mean you will immediately be able to use your live account

You must ensure you have completed Stage 1 Testing on the Test Server, before you are granted access to your live account. Further information on testing can be found on sagepay.com.

NB – Without confirmation from the bank and without a Direct Debit submission, Sage Pay will not be able to set your account live. You will only be charged by Sage Pay when your account has valid Direct Debit details and confirmation of your merchant details from the bank.

Once your live account is active, you should point your website transaction registration scripts to the following URL:

<https://live.sagepay.com/gateway/service/vspform-register.vsp>

You should then run an end-to-end transaction through your site, ordering something relatively inexpensive from your site and paying using a valid credit or debit card. If you receive an authorisation code, then everything is working correctly.

You should then log into MySagePay on the live server <https://live.sagepay.com/mysagepay>. It is worth noting here that none of the users you set up on the MySagePay system on the test server are migrated across to live. This is because many companies use third party web designers to help design the site and create users for them during testing that they would not necessarily like them to have in a live environment. You will need to recreate any valid users on the live system when you first log in as described in 5.2.

Once logged in, locate your test transaction and **VOID** it so you are not charged. At this stage the process is complete.

10.0 Congratulations, you are live with Sage Pay Form

Well done. Hopefully the process of getting here was as painless and hassle free as possible. You should contact us with any transaction queries that arise or for any help you need with MySagePay.

Here are the best ways to reach us and the best department to contact:

- If you require any information on additional services, email tellmemore@sagepay.com
- If you have a query regarding a Sage Pay invoice, email finance@sagepay.com
- If you have a question about a transaction, have issues with your settlement files, are having problems with your payment pages or MySagePay screens, or have a general question about online payments or fraud, email support@sagepay.com with your Sage Pay Vendor Name included in the mail.
- If you have any suggestions for future enhancements to the system, or additional functionality you'd like to see added, please email feedback@sagepay.com with your comments. We do take all comments on board when designing upgrades, although we may not be able to answer every mail we get.
- You can call on 0845 111 44 55, for any type of enquiry.

Your email address will be added to our group mail list used to alert you to upgrades and other pending events.

You can also always check our system availability and current issues on the Sage Pay Monitor page at www.sagepay.com/support/system-monitor.

Thanks again for choosing Sage Pay, and we wish you every success in your e-commerce venture.

11.0 Character Sets and Encoding

All transactions are simple synchronous HTTPS POSTs sent from a script on your servers to the Sage Pay gateway, with the same script reading the Response component of that POST to determine success or failure. These POSTs can be sent using any HTTPS compatible objects (such as cURL in PHP, HttpRequest in .NET and Apache HttpComponents in Java).

The data should be sent as URL Encoded Name=Value pairs separated with & characters and sent to the Sage Pay Server URL with a Service name set to the message type in question.

The following sections detail the contents of the POSTs and responses, between your server and ours. The format and size of each field is given, along with accepted values and characters. The legend below explains the symbols:

Aa	Letters (A-Z and a-z)	^	Caret	+	Plus
0-9	Numbers	[]	Square brackets	()	Parentheses
á	Accented characters	*	Asterisk	;	Semi-colon
&	Ampersand	'	Apostrophe (single quote)	 	Pipe
@	At sign	/\	Slash and Backslash	!	Exclamation Mark
:	Colon	-	Hyphen	 	Space
,	Comma	_	Underscore	~	Tilde
{}	Curly brackets	.	Full stop / Period	=	Equals
"	Quotes	\$	Dollar	US	Valid 2-letter US States
#	Hash	?	Question Mark	DATE	Date in the format YYYY-MM-DD
ISO639	ISO 639-2 (2-letter language codes)	BASE64	Valid Base64 characters (A-Z,a-z,0-9,+ and /)	BOOLEAN	True or False
ISO3166	ISO 3166-1 (2-letter country codes)	CR / LF	New line (Carriage Return and Line Feed)	RFC532N	RFC 5321/5322 (see also RFC 3696) compliant email addresses Valid HTML with no active content.
ISO4217	ISO 4217 (3-letter currency codes)	RFC1738	RFC 1738 compliant HTTP(S) URL All non-compliant characters, including spaces should be URL encoded	<HTML>	Script will be filtered. Includes all valid letters, numbers, punctuation and accented characters

Appendix A: Transaction Registration

A1. Form Fields

The final confirmation page on your website should contain an HTML FORM with the Action set to the Sage Pay Form submission URL and the following 4 hidden fields as part of that Form.

Request format

Name	Mandatory	Format	Max Length	Allowed Values	Description
VPSProtocol	Yes	0-9 -	4 chars	3.00	This is the version of the protocol you are integrating with. Default or incorrect value is taken to be 3.00 .
TxType	Yes	Aa	15 chars	PAYMENT DEFERRED AUTHENTICATE	The value should be in UPPERCASE.
Vendor	Yes	Aa 0-9	15 chars		Used to authenticate your site. This should contain the Sage Pay Vendor Name supplied by Sage Pay when your account was created.
Crypt	Yes	BASE64 @	40 chars		Your site builds the Crypt field in real time for each order. The contents of the field are described below. All other transaction information is encrypted then encoded. See below.

A1.1 The Crypt Field

1. The Crypt field should contain all the other transaction information in plain text as Name=Value fields separated by '&' characters. Ensure that all mandatory fields are present and that there are no spaces after the '&' character.
2. This string should then be encrypted using AES(block size 128-bit) in CBC mode with PKCS#5 padding using the provided password as both the key and initialisation vector and encode the result in hex (making sure the letters are in upper case).
3. Prepend the '@' sign to the beginning of the encoded result.

To decrypt use the same procedure in decryption mode, making sure you remove the '@' sign before doing so.

A1.2 Example Crypt Field

Using the key 55a51621a6648525

To encrypt the following request we should get the encrypted result below it

VendorTxCode=TxCode-1310917599-223087284&Amount=36.95&Currency=GBP&Description=description&CustomerName=Fname
Surname&CustomerEMail=customer@example.com&BillingSurname=Surname&BillingFirstnames=Fname&BillingAddress1=BillAddress Line
1&BillingCity=BillCity&BillingPostCode=W1A
1BL&BillingCountry=GB&BillingPhone=447933000000&DeliveryFirstnames=Fname&DeliverySurname=Surname&DeliveryAddress1=BillAddress
Line 1&DeliveryCity=BillCity&DeliveryPostCode=W1A
1BL&DeliveryCountry=GB&DeliveryPhone=447933000000&SuccessURL=https://example.com/success&FailureURL=https://example.com/failur
e

@2DCD27338114D4C39A14A855702FBAB2EF40BCAC2D76A3ABC0F660A07E9C1C921C2C755BA9B59C39F882FBF6DFED114F23141D94E50A01A665B1E31A86C07CA1CD1BB8EF5B6CF2C23D495CD
79F9C0F678D61773E7A1AA30AA5B23D56503FC0B52AC0694A8C341263D2C5FE1BAD93BDB94726761E155E900448F644AF1F67BE1AC77E852B9D90809A44F258EE9478B6D8C1C4ED58759263E7DBF
8871C6592287C0358F36F4EEC326CEDDD440DA2FED8AB35F1B630A5C6FA671E4D78CC8CACECF9DFDC31D6C5EC8270FB21E297E2C2E14F99A04223EFFD4F00062D440E78A3D2C7140EC8F123D24
7B75E7482AE98858DA34D37EDE6D7C69AA74391F559305CF675ADB3615244A107ABBB6AF26E29A2FFA059B12688D90FE09E0DE069325BFF3587A695F5DA36E4B809B69CC9A37034F166B63B5A62
B986F4DA34E9AC9516AFDE70642EC7DAD1AEB9A3A1F347D6AC7046E967DCBFE7ACFCEE5DAFC0B29F1765032B3060EBE565CBD57D092075D15CF12725199C6881605B2E0F105698CE3ADD04361C
A9D620C187B90E3F9849445B5C3C0FDF1768BFFD61F97E51316826F4F10E0E3E668F0A9F5ED9CCDA6F2C7C957F12DB48F9041482E3D035E7A91852C404BFA325FED947E71F57B871DFAC6AF4FF2
9F4513A4A80B2D7ECC9D19D47ED04FA99CD9C881DFA771E1EA4F3F9B2C5AC673EF3DA2699A309CC8522993A63CB8D45D3CDF09B1DFDC573CD19679B250AD6721450B5042F201670B464505DCAE
F59E2C67ABACC9AE2EEE793CE191FEBF66B8FAF4204EFFB359246B9C99FB52805C46375FF35140F74707FBC73C7731A28A2C883A

A1.3 Request Crypt Fields

Name	Mandatory	Format	Max Length	Allowed Values	Description
VendorTxCode	Yes	Aa 0-9 { } - _	40 chars		This should be your own reference code to the transaction. Your site should provide a completely unique <code>VendorTxCode</code> for each transaction.
Amount	Yes	0-9 - ,		0.01 to 100,000.00	Amount for the transaction containing minor digits formatted to 2 decimal places where appropriate. e.g. 5.10 or 3.29. Values such as 3.235 will be rejected. Minimum for no minor unit currencies like JPY is 1. Amounts must be in the UK currency format. The period must be used to indicate the decimal place. The comma must only be used to separate groups of thousands.
Currency	Yes	ISO4217	3 chars	ISO 4217 Examples: GBP , EUR and USD	The currency the transaction is performed in. This must be supported by one of your Sage Pay merchant accounts or the transaction will be rejected.
Description	Yes	<HTML>	100 chars		Free text description of goods or services being purchased. This will be displayed on the Sage Pay payment page as the customer enters their card details.
SuccessURL	Yes	RFC1738	2000 chars		This should be the fully qualified URL (including http:// or https:// header). It is the URL of the page/script to which the user is redirected if the transaction is successful. You may attach parameters if you wish. Sage Pay Form will also send an encrypted field containing important information appended to this URL (see below).
FailureURL	Yes	RFC1738	2000 chars		This should be the fully qualified URL (including http:// or https:// header). It is the URL of the page/script to which the user is redirected if the transaction is not successful, aborted or an error occurs. You may attach parameters if you wish. Sage Pay Form will also send an encrypted field containing important information appended to this URL (see below).

CustomerName	No	Aa á / \ & - ' , 0-9	100 chars		If provided the customer's name will be included in the confirmation emails and stored in MySagePay.
CustomerEMail	No	RFC532N	255 chars		If provided, the customer will be emailed on completion of a successful transaction (but not an unsuccessful one). If you wish to use multiple email addresses, you should add them using the : (colon) character as a separator. e.g. me@mail1.com:me@mail2.com
VendorEMail	No	RFC532N	255 chars		If provided, an email will be sent to this address when each transaction completes (successfully or otherwise). If you wish to use multiple email addresses, you should add them using the : (colon) character as a separator. e.g. me@mail1.com:me@mail2.com
SendEMail	No	0-9	Flag	0 1 2	0 = Do not send either customer or vendor emails 1 = Send customer and vendor emails if addresses are provided 2 = Send vendor email but NOT the customer email If you do not supply this field, 1 is assumed and emails are sent if addresses are provided.
EmailMessage	No	<HTML>	7500 chars		A message to the customer which is inserted into the successful transaction emails only. If provided this message is included toward the top of the customer confirmation emails.
BillingSurname	Yes	Aa á / \ & - ' , 0-9	20 chars		Customer billing details. All mandatory fields must contain a value, apart from the BillingPostcode. The BillingPostcode can be blank for countries that do not have postcodes (e.g. Ireland) but is required in all countries that do have them. Providing a blank field when information is required will cause an error.
BillingFirstnames	Yes	Aa á / \ & - ' , 0-9	20 chars		
BillingAddress1	Yes	Aa á / \ & - ' , 0-9 : + () CR / LF	100 chars		

BillingAddress2	No	Aa á / \ & - ' , 0-9 : + () CR / LF	100 chars		The BillingState becomes mandatory when the BillingCountry is set to US .
BillingCity	Yes	Aa á / \ & - ' , 0-9 : + () CR / LF	40 chars		
BillingPostCode	Yes	Aa - 0-9	10 chars		
BillingCountry	Yes	ISO3166	2 chars	ISO 3166 Examples: GB , IE and DE	
BillingState	No	US	2 chars	Examples: AL , MS and NY	
BillingPhone	No	0-9 - Aa + ()	20 chars		
DeliverySurname	Yes	Aa á / \ & - ' , 0-9	20 chars		Customer delivery details. All mandatory fields must contain a value, apart from the DeliveryPostcode. The DeliveryPostcode can be blank for countries that do not have postcodes (e.g. Ireland) but is required in all countries that do have them. Providing a blank field when information is required will cause an error. The DeliveryState becomes mandatory when the DeliveryCountry is set to US .
DeliveryFirstnames	Yes	Aa á / \ & - ' , 0-9	20 chars		
DeliveryAddress1	Yes	Aa á / \ & - ' , 0-9 : + () CR / LF	100 chars		
DeliveryAddress2	No	Aa á / \ & - ' , 0-9 : + () CR / LF	100 chars		
DeliveryCity	Yes	Aa á / \ & - ' , 0-9 : + () CR / LF	40 chars		
DeliveryPostCode	Yes	Aa - 0-9	10 chars		
DeliveryCountry	Yes	ISO3166	2 chars	ISO 3166 Examples: GB , IE and DE	
DeliveryState	No	US	2 chars	Examples: AL , MS and NY	
DeliveryPhone	No	0-9 - Aa + ()	20 chars		

Basket	No	<HTML>	7500 chars	See A1.4	<p>You can use this field to supply details of the customer's order. This information will be displayed to you in MySagePay.</p> <p>If this field is supplied then the <code>BasketXML</code> field should not be supplied.</p>
AllowGiftAid	No	BOOLEAN	Flag	0 (default) 1	<p>This flag allows the gift aid acceptance box to appear for this transaction on the payment page. This only appears if your vendor account is Gift Aid enabled.</p> <p>0 = No Gift Aid box displayed (default)</p> <p>1 = Display Gift Aid box on payment page.</p>
ApplyAVSCV2	No	0-9	Flag	0 (default) 1 2 3	<p>Using this flag you can fine tune the AVS/CV2 checks and rule set you've defined at a transaction level. This is useful in circumstances where direct and trusted customer contact has been established and you wish to override the default security checks.</p> <p>0 = If AVS/CV2 enabled then check them. If rules apply, use rules (default)</p> <p>1 = Force AVS/CV2 checks even if not enabled for the account. If rules apply, use rules.</p> <p>2 = Force NO AVS/CV2 checks even if enabled on account.</p> <p>3 = Force AVS/CV2 checks even if not enabled for the account but DON'T apply any rules.</p> <p>This field is ignored for PAYPAL transactions.</p>

Apply3DSecure	No	0-9	Flag	0 (default) 1 2 3	<p>Using this flag you can fine tune the 3D-Secure checks and rule set you've defined at a transaction level. This is useful in circumstances where direct and trusted customer contact has been established and you wish to override the default security checks.</p> <p>0 = If 3D-Secure checks are possible and rules allow, perform the checks and apply the authorisation rules. (default)</p> <p>1 = Force 3D-Secure checks for this transaction if possible and apply rules for authorisation.</p> <p>2 = Do not perform 3D-Secure checks for this transaction and always authorise.</p> <p>3 = Force 3D-Secure checks for this transaction if possible but ALWAYS obtain an auth code, irrespective of rule base.</p> <p>This field is ignored for PAYPAL transactions.</p>
BillingAgreement	No	BOOLEAN	Flag	0 1	<p>This field is for future use. It is not currently possible to perform Repeats for PayPal Transactions using the FORM Integration method.</p>
BasketXML	No		20000 chars	See A1.5	<p>A more flexible version of the current basket field which can be used instead of the basket field.</p> <p>If this field is supplied then the Basket field should not be supplied.</p>
CustomerXML	No		2000 chars	See A1.6	<p>This can be used to supply information on the customer for purposes such as fraud screening.</p>
SurchargeXML	No		800 chars	See A1.7	<p>Use this field to override current surcharge settings in "My Sage Pay" for the current transaction. Percentage and fixed amount surcharges can be set for different payment types.</p>

VendorData	No	Aa 0-9	200 chars		Use this field to pass any data you wish to be displayed against the transaction in MySagePay.
ReferrerID	No	Aa a / \ & - ' , 0-9 : + () CR / LF	40 char		This can be used to send the unique reference for the Partner that referred the Vendor to Sage Pay.
Language	No	ISO639	2 chars	ISO 639-2 Examples: EN , DE and FR	The language the customer sees the payment pages in is determined by the code sent here. If this is not supplied then the language default of the shopper's browser will be used. If the language is not supported then the language supported in the templates will be used. Currently supported languages in the Default templates are: French, German, Spanish, Portuguese, Dutch and English.
Website	No	Aa a / \ & - ' , 0-9 : + () CR / LF	100 chars		Reference to the website this transaction came from. This field is useful if transactions can originate from more than one website. Supplying this information will enable reporting to be performed by website.
FIRecipientAcctNumber	No	Aa 0-9	10 chars		This should either be the first 6 and the last 4 characters of the primary recipient PAN (no spaces). Where the primary recipient account is not a card this will contain up to 10 characters of the account number (alphanumeric), unless the account number is less than 10 characters long in which case the account number will be present in its entirety. This field is only required for UK merchants who have a merchant category code of 6012 (Financial Institutions)
FIRecipientSurname	No	Aa	20 chars		This is the surname of the primary recipient. No special characters such as apostrophes or hyphens are permitted. This field is only required for UK merchants who have a merchant category code of 6012 (Financial Institutions)

FIRecipientPostcode	No	Aa 0-9			<p>This is the postcode of the primary recipient.</p> <p>This field is only required for UK merchants who have a merchant category code of 6012 (Financial Institutions)</p>
FIRecipientDoB	No	0-9			<p>This is the date of birth of the primary recipient in the format YYYYMMDD</p> <p>This field is only required for UK merchants who have a merchant category code of 6012 (Financial Institutions)</p>

A1.4 SurchargeXML

Use this field to override the default surcharge in MySagePay for the current transaction. You can set a different surcharge value for each payment type (except PayPal). The value can either be a percentage or fixed amount.

If a surcharge amount for the payment type selected is NOT included in the Surcharge XML, then the default value for that payment type will be used from MySagePay. If you wish to remove the surcharge value currently set in MySagePay for a payment type then you should send through the payment type with a surcharge value of 0 in the Surcharge XML. The XML tags should follow the order stated in the table.

Surcharge XML elements

Node/Element	Mandatory	Format	Max Length	Allowed Values	Description
<surcharges>	No	Node			The root element for all other surcharge elements.
L<surcharge>	Yes	XML container element			At least one must occur in the xml file. There can be multiple <surcharge> elements but each must have a unique <paymentType>.
L<paymentType>	Yes	Aa	15 chars	VISA MC MCDEBIT DELTA MAESTRO UKE AMEX DC JCB	VISA is Visa MC is MasterCard MCDEBIT is Debit MasterCard DELTA is Visa Debit MAESTRO is Domestic and International issued Maestro UKE is Visa Electron AMEX is American Express DC is Diners Club International and Discover JCB is Japan Credit Bureau The value should be in UPPERCASE.
L<percentage>	Yes unless a <fixed> element supplied	0-9 , -	Maximum 3 digits to 2 decimal places		The percentage of the transaction amount to be included as a surcharge for the transaction for the payment type of this element.
L<fixed>	Yes unless a <fixed> element supplied	0-9 , -			Amount of the surcharge containing minor digits formatted to 2 decimal places where appropriate. e.g. 5.10 or 3.29. Values such as 3.235 will be rejected. Minimum for no minor unit currencies like JPY is 1. Amounts must be in the UK currency format. The period must be used to indicate the decimal place. The comma must only be used to separate groups of thousands.

View example Surcharge XML snippets on sagepay.com

A1.5 Basket

The shopping basket contents can be passed in a single, colon-delimited field, in the following format:

```
Number of lines of detail in the basket field:
Item 1 Description:
Quantity of item 1:
Unit cost item 1 without tax:
Tax applied to item 1:
Cost of Item 1 including tax:
Total cost of item 1 (Quantity x cost including tax):
Item 2 Description:
Quantity of item 2:
....
Cost of Item including tax:
Total cost of item
```

- The line breaks above are included for readability only. No line breaks are needed; the only separators should be the colons.
- The first value “The number of lines of detail in the basket” is **NOT** the total number of items ordered, but the total number of rows of basket information. In the example below there are 6 items ordered, (1 DVD player and 5 DVDs) but the number of lines of detail is 4 (the DVD player, two lines of DVDs and one line for delivery).

Example:

Items	Quantity	Item value	Item Tax	Item Total	Line Total
Pioneer NSDV99 DVD-Surround Sound System	1	424.68	74.32	499.00	499.00
Donnie Darko Director's Cut	3	11.91	2.08	13.99	41.97
Finding Nemo	2	11.05	1.94	12.99	25.98
Delivery	---	---	---	---	4.99

```
4:Pioneer NSDV99 DVD-Surround Sound System:1:424.68:74.32:499.00: 499.00:Donnie Darko Director's Cut:3:11.91:2.08:13.99:41.97:
Finding Nemo:2:11.05:1.94:12.99:25.98: Delivery:---:---:---:---:4.99
```

If you wish to leave a field empty, you must still include the colon. E.g. 1:DVD Player:1:199.99:::199.9

A1.6 BasketXML

The basket can be passed as an XML document with extra information that can be used for:

1. Displaying to the customer when they are paying using PayPal.
2. Displaying in MySagePay to give you more detail about the transaction.
3. Displaying on the payment page. It is possible to send through a delivery charge and one or more discounts. The discount is at the order level rather than item level and is a fixed amount discount. You can however add multiple discounts to the order.
4. More accurate fraud screening through ReD. Extra information for fraud screening that can be supplied includes; details of the items ordered, and also the shipping details and the recipient details. Any information supplied will be sent to ReD to enable them to perform more accurate fraud screening.
5. The supplying of TRIPs information. However this information will only be of use to you if your acquiring bank is Elavon. TRIPs information which can be supplied includes details of airlines, tours, cruises, hotels and car rental. If your acquiring bank is Elavon this information will be sent in the daily settlement file.

NB : Please note if your customer is buying more than one service from you (i.e. more than one of following ; airlines, tours, cruises, hotels and car rental) you will need to send the information through as separate transactions.

No validation is performed on the totals of the basket, it is your responsibility to ensure that the amounts are correct and that the total of the basket matches the transaction amount sent in the Registration

Both the `Basket` field and the `BasketXML` field are optional. If basket information is to be supplied, you cannot pass both the `Basket` and the `BasketXML` field, only one of them needs to be passed.

The XML tags should follow the order stated in the table.

Basket XML elements

Node/Element	Mandatory	Format	Max Length	Allowed Values	Description
<basket>	No	Node			The root element for all other basket elements.
L<agentId>	No	Aa 0-9 +	16 chars		The ID of the seller if using a phone payment.

L<item>		XML container element			There can be as many Items are you like in the BasketXML, each holding a different item and recipient. The sum of all <TotalGrossAmount> in all item elements and the <deliveryGrossAmount> amount must match the Amount field sent with the transaction
L<description>	Yes	Aa á / \ - ' , 0-9 + ()	100 chars		Description of the item
L<productSku>	No	Aa - 0-9 +	12 chars		Item SKU. This is your unique product identifier code.
L<productCode>	No	Aa - 0-9 +	12 chars		Item product code.
L<quantity>	Yes	0-9 -	12 chars		Quantity of the item ordered
L<unitNetAmount>	Yes	0-9 -	14 chars		Cost of the item before tax containing minor digits formatted to 2 decimal places where appropriate. e.g. 5.10 or 3.29. Values such as 3.235 will be rejected. Minimum for no minor unit currencies like JPY is 1. Amounts must be in the UK currency format. The period must be used to indicate the decimal place. The comma must only be used to separate groups of thousands.
L<unitTaxAmount>	Yes	0-9 -	14 chars		Amount of tax on the item containing minor digits formatted to 2 decimal places where appropriate. e.g. 5.10 or 3.29. Values such as 3.235 will be rejected. Minimum for no minor unit currencies like JPY is 1. Amounts must be in the UK currency format. The period must be used to indicate the decimal place. The comma must only be used to separate groups of thousands.
L<unitGrossAmount>	Yes	0-9 -	14 chars		<unitNetAmount> + <unitTaxAmount>
L<totalGrossAmount>	Yes	0-9 -	14 chars		<unitGrossAmount> x <quantity>
L<recipientFName>	No	Aa / \ - - ' + ()	20 chars		The first name of the recipient of this item.
L<recipientLName>	No	Aa / \ - - ' + ()	20 chars		The last name of the recipient of this item.

L<recipientMName>	No	Aa	1 char		The middle initial of the recipient of this item.
L<recipientSal>	No	Aa	4 chars		The salutation of the recipient of this item.
L<recipientEmail>	No	RFC532N	45 chars		The email of the recipient of this item.
L<recipientPhone>	No	0-9 - Aa + ()	20 chars		The phone number of the recipient of this item.
L<recipientAdd1>	No	Aa / \ - - ' , 0-9 : + () CR / LF	100 chars		The first address line of the recipient of this item.
L<recipientAdd2>	No	Aa / \ - - ' , 0-9 : + () CR / LF CR / LF	100 chars		The second address line of the recipient of this item.
L<recipientCity>	No	Aa / \ - - ' , 0-9 : + () CR / LF CR / LF	40 chars		The city of the recipient of this item.
L<recipientState>	No	US	2 chars		If in the US, the 2 letter code for the state of the recipient of this item.
L<recipientCountry>	No	ISO3166	2 chars		The 2 letter country code (ISO 3166) of the recipient of this item.
L<recipientPostCode>	No	Aa - 0-9	9 chars		The postcode of the recipient of this item.
L<itemShipNo>	No	Aa 0-9 + -	19 chars		The shipping item number.
L<itemGiftMsg>	No	Aa 0-9 +	160 chars		Gift message associated with this item.
L<deliveryNetAmount>	No	0-9 -	14 chars		Cost of delivery before tax containing minor digits formatted to 2 decimal places where appropriate. e.g. 5.10 or 3.29. Values such as 3.235 will be rejected. Minimum for no minor unit currencies like JPY is 1. Amounts must be in the UK currency format. The period must be used to indicate the decimal place. The comma must only be used to separate groups of thousands.

L<deliveryTaxAmount>	No	0-9 -	14 chars		Amount of tax on delivery containing minor digits formatted to 2 decimal places where appropriate. e.g. 5.10 or 3.29. Values such as 3.235 will be rejected. Minimum for no minor unit currencies like JPY is 1. Amounts must be in the UK currency format. The period must be used to indicate the decimal place. The comma must only be used to separate groups of thousands.
L<deliveryGrossAmount>	No	0-9 -	14 chars		<deliveryNetAmount> + <deliveryTaxAmount>
L<discounts>	No				The root element for all other discount elements.
L<discount>	Yes				There can be multiple discount elements.
L<fixed>	Yes	0-9 -	14 chars	Zero or greater	This is the amount of the discount. This is the monetary value of the discount. The value sent will be subtracted from the overall total
L<description>	No	Aa á / \ - - ' , 0-9 : + () @ { } ; - ^ " ~ [] ¢ \$ = ! # ?	100 chars		This is the description of the discount. This will appear on the payment pages, MySagePay and the PayPal checkout pages if appropriate.
L<shipId>	No	Aa + 0-9	16 chars		The ship customer ID.
L<shippingMethod>	No	Aa	1 char	C- Low Cost D – Designated by customer I – International M – Military N – Next day/overnight O – Other P – Store pickup T – 2 day service W – 3 day service	The shipping method used.
L<shippingFaxNo>	No	0-9 - Aa + ()	20 chars		The Fax Number
L<hotel>	No				Used to provide hotel information for settlement. There can be only one hotel element.

L<checkIn>	Yes	DATE			Check in date for hotel.
L<checkOut>	Yes	DATE			Check out date for hotel.
L<numberInparty>	Yes	0-9	3 chars		Number of people in the hotel booking.
L<folioRefNumber>	No	Aa 0-9 +	10 chars		Folio reference number for hotel.
L<confirmedReservation>	No	Aa		Y N	Flag to indicate whether a guest has confirmed their reservation Y= Confirmed Reservation N = Unconfirmed Reservation
L<dailyRoomRate>	Yes	0-9 - Aa	15 chars		Daily room rate for the hotel.
L<guestName>	Yes	Aa 0-9 +	20 chars		Name of guest
L<cruise>	No				Used to provide cruise information for settlement. There can be only one cruise element.
L<checkIn>	Yes	DATE			Start date for cruise.
L<checkOut>	Yes	DATE			End date for cruise.
L<cardRental>	No				Used to provide car rental information for settlement. There can be only one car rental element.
L<checkIn>	Yes	DATE			Check in date for car rental.
L<checkOut>	Yes	DATE			Check out date for car rental.
L<tourOperator>	No				Used to provide tour operator information for settlement. There can be only one tour operator element.
L<checkIn>	Yes	DATE			Check in date for tour operator.
L<checkOut>	Yes	DATE			Check out date for tour operator.
L<airline>	No				Used to provide airline information for settlement. There can be only one airline element
L<ticketNumber>	Yes	Aa 0-9	11 chars		The airline ticket number
L<airlineCode>	Yes	0-9	3 chars		IATA airline code
L<agentCode>	Yes	0-9	8 chars		IATA agent code
L<agentName>	Yes	Aa 0-9	26 chars		Agency name
L<flightNumber>	No	Aa 0-9	6 chars		Flight number
L<restrictedTicket>	Yes	BOOLEAN			Can be 0, 1, true or false.

L<passengerName>	Yes	Aa 0-9	29 chars		Name of passenger
L<originatingAirport>	Yes	Aa	3 chars		IATA airport code
L<segment>	Yes				Contains other elements detailing the segment At least one segment element must be supplied under the airline element, but can supply up to 4 segments.
L<carrierCode>	Yes	Aa	3 chars		IATA carrier code
L<class>	Yes	Aa 0-9	3 chars		Class of service
L<stopover>	Yes	BOOLEAN			Can be 0,1, true or false to indicate a stopover
L<legDepartureDate>	Yes	DATE			Departure date of the segment.
L<destination>	Yes	Aa	3 chars		IATA airport code of destination
L<fareBasis>	No	Aa 0-9	6 chars		Fare basis code
L<customerCode>	No	Aa 0-9	20 chars		Airline customer code
L<invoiceNumber>	No	Aa 0-9	15 chars		Airline Invoice Number
L<dinerCustomerRef>	No	Aa 0-9	15 chars		Diners customer reference Can include up to 5 elements

View example Basket XML snippets on sagepay.com

A1.7 CustomerXML

The extra fields detailed below can be passed as an xml document for more accurate fraud screening. The XML tags should follow the order stated in the table.

Customer XML elements

Node/Element	Mandatory	Format	Max Length	Allowed Values	Description
<customer>	No	Node			The root element for all other customer elements.
L<customerMiddleInitial>	No	Aa	1 char		The middle initial of the customer.
L<customerBirth>	No	DATE	19 chars		The date of birth of the customer.
L<customerWorkPhone>	No	0-9 - Aa + ()	19 chars		The work phone number of the customer.
L<customerMobilePhone>	No	0-9 - Aa + ()			The mobile number of the customer.
L<previousCust>	No	BOOLEAN			Whether the customer is a previous customer or new.
L<timeOnFile>	No	0-9 + -	16 chars	Min Value 0	The number of days since the card was first seen.
L<customerId>	No	Aa 0-9	1 char		The ID of the customer

View example Customer XML snippets on sagepay.com

Appendix B: Transaction Completion

For Sage Pay Form transactions, Sage Pay cannot guarantee to return the customer to your website. If the customer closes their browser mid-way through a transaction, or if something goes wrong at any redirect stages, it will be up to you to check the status of the transactions on the MySagePay reporting screens.

In normal circumstances, however, where the customer does not close their browser and there are no redirection problems, Sage Pay Form will return them to your site, either to the `SuccessURL` (in the event the transaction was successful), or the `FailureURL` (in all other circumstances).

The system will append to the `SuccessURL` or `FailureURL` a field called `Crypt`, in the manner:

[ResponseURL]?crypt=[encrypted_information]

Or if the URL already has your own fields attached, it will be appended thus:

[ResponseURL]?vendor1=test&vendor2=test2&crypt=[encrypted_information]

The `SuccessURL` and `FailureURL` fields should point to scripts on your server that extract the information in the `crypt` field and use it to update your database (if you have one) and/or format an appropriate response page for the customer. This is not compulsory, however, and you may choose to simply direct customers to a static HTML page that ignores the contents of the `crypt` field. In such cases, you will need to manually check the MySagePay report pages to determine if a transaction succeeded or failed. In fact, we recommend you always check the MySagePay pages before sending any goods just to confirm the status of each transaction.

The `Crypt` field contains the plain text shown overleaf. For details of the Encryption used see Appendix A1.1

Remember to remove the '@' sign before decrypting.

B1. Response Crypt Fields

Name	Mandatory	Format	Max Length	Allowed Values	Description
Status	Yes	Aa	15 chars	OK NOTAUTHED MALFORMED INVALID ABORT REJECTED AUTHENTICATED REGISTERED ERROR	<p>If the Status is not OK, the StatusDetail field will give more information about the problem.</p> <p>OK = Process executed without error.</p> <p>NOTAUTHED = The Sage Pay gateway could not authorise the transaction because the details provided by the customer were incorrect, or insufficient funds were available. However the transaction has completed.</p> <p>MALFORMED = Input message was missing fields or badly formatted – normally will only occur during development.</p> <p>INVALID = Transaction was not registered because although the POST format was valid, some information supplied was invalid. E.g. incorrect Vendor or Currency.</p> <p>ABORT = The Transaction could not be completed because the user clicked the CANCEL button on the payment pages, or went inactive for 15 minutes or longer.</p> <p>REJECTED = The Sage Pay System rejected the transaction because of the fraud screening rules you have set on your account. Note: The bank may have authorised the transaction but your own rule bases for AVS/CV2 or 3D-Secure caused the transaction to be rejected.</p> <p>AUTHENTICATED = The 3D-Secure checks were performed successfully and the card details secured at Sage Pay. Only returned if TxType is AUTHENTICATE.</p>

					<p>REGISTERED = 3D-Secure checks failed or were not performed, but the card details are still secured at Sage Pay. Only returned if TxType is AUTHENTICATE.</p> <p>ERROR = A problem occurred at Sage Pay which prevented transaction registration. Please notify Sage Pay if a Status of ERROR is seen, together with your Vendor, VendorTxCode and the StatusDetail.</p>
StatusDetail	Yes	Aa 0-9 - () , :	255 chars		<p>Human-readable text providing extra detail for the Status message.</p> <p>Always check StatusDetail if the Status is not OK</p>
VendorTxCode	Yes	Aa 0-9 {} - _	40 chars		Same as supplied in A1.3
VPSTxId	No	Aa 0-9 -	38 chars		<p>The Sage Pay ID to uniquely identify the transaction on our system.</p> <p>Only present if Status not INVALID, MALFORED or ERROR.</p>
TxAuthNo	No	0-9	10 chars		<p>Sage Pay unique Authorisation Code for a successfully authorised transaction.</p> <p>Only present if Status is OK.</p>
Amount	Yes	0-9 - ,		0.01 to 100,000.00	<p>Amount for the transaction containing minor digits formatted to 2 decimal places where appropriate.</p> <p>e.g. 5.10 or 3.29. Values such as 3.235 will be rejected.</p> <p>Minimum for no minor unit currencies like JPY is 1.</p> <p>Amounts must be in the UK currency format. The period must be used to indicate the decimal place. The comma must only be used to separate groups of thousands.</p>
AVSCV2	Yes	Aa	50 chars	<p>ALLMATCH SECURITY CODE MATCH ONLY ADDRESS MATCH ONLY NO DATA MATCHES DATA NOT CHECKED</p>	<p>This is the response from AVS and CV2 checks. Provided for Vendor info and backward compatibility with the banks. Rules set up in MySagePay will accept or reject the transaction based on these values.</p> <p>More detailed results are split out in the next three fields. Not present if the Status is AUTHENTICATED or REGISTERED.</p>

AddressResult	Yes	Aa	20 chars	NOTPROVIDED NOTCHECKED MATCHED NOTMATCHED	The specific result of the checks on the cardholder's address numeric from the AVS/CV2 checks. Not present if the Status is AUTHENTICATED or REGISTERED
PostCodeResult	Yes	Aa	20 chars	NOTPROVIDED NOTCHECKED MATCHED NOTMATCHED	The specific result of the checks on the cardholder's Postcode from the AVS/CV2 checks. Not present if the Status is AUTHENTICATED or REGISTERED
CV2Result	Yes	Aa	20 chars	NOTPROVIDED NOTCHECKED MATCHED NOTMATCHED	The specific result of the checks on the cardholder's CV2 code from the AVS/CV2 checks. Not present if the Status is AUTHENTICATED or REGISTERED
GiftAid	Yes	BOOLEAN			This field is always present even if GiftAid is not active on your account. 0 = The Gift Aid box was not checked this transaction. 1 = The customer checked the Gift Aid box on the payment page

3DSecureStatus	Yes	Aa	50 chars	OK NOTCHECKED NOTAVAILABLE NOTAUTHED INCOMPLETE ATTEMPTONLY ERROR	<p>This field details the results of the 3D-Secure checks (where appropriate)</p> <p>OK - 3D-Secure checks carried out and user authenticated correctly.</p> <p>NOTCHECKED – 3D-Secure checks were not performed. This indicates that 3D-Secure was either switched off at an account level, or disabled at transaction registration.</p> <p>NOTAVAILABLE – The card used was either not part of the 3D-Secure Scheme, or the authorisation was not possible.</p> <p>NOTAUTHED – 3D-Secure authentication checked, but the user failed the authentication.</p> <p>INCOMPLETE – 3D-Secure authentication was unable to complete. No authentication occurred.</p> <p>ATTEMPTONLY– 3D-Secure attempted but cardholder was not enrolled.</p> <p>ERROR - Authentication could not be attempted due to data errors or service unavailability in one of the parties involved in the check.</p>
CAVV	No	Aa 0-9	32 chars		<p>The encoded result code from the 3D-Secure checks (CAVV or UCAF).</p> <p>Only present if the 3DSecureStatus field is OK</p>
AddressStatus	Yes	Aa	20 chars	NONE CONFIRMED UNCONFIRMED	<p>PayPal Transactions Only.</p> <p>If AddressStatus is CONFIRMED and PayerStatus is VERIFIED, the transaction may be eligible for PayPal Seller Protection. To learn more about PayPal Seller Protection, please contact PayPal directly or visit paypal.com</p>
PayerStatus	Yes	Aa	20 chars	VERIFIED UNVERIFIED	

CardType	Yes	Aa	15 chars	VISA MC MCDEBIT DELTA MAESTRO UKE AMEX DC JCB PAYPAL	VISA is Visa MC is MasterCard MCDEBIT is Debit MasterCard DELTA is Visa Debit MAESTRO is Domestic and International issued Maestro UKE is Visa Electron AMEX is American Express DC is Diners Club International and Discover JCB is Japan Credit Bureau PAYPAL
Last4Digits	Yes	0-9	4 chars		The last 4 digits of the card number used in this transaction. PayPal transactions have 0000 This field is supplied to allow merchants using wallet systems to identify the card to their customers
FraudResponse	No	Aa	10 chars	ACCEPT CHALLENGE DENY NOTCHECKED	ACCEPT means ReD recommends that the transaction is accepted DENY means ReD recommends that the transaction is rejected CHALLENGE means ReD recommends that the transaction is reviewed. You have elected to have these transactions either automatically accepted or automatically denied at a vendor level. Please contact Sage Pay if you wish to change the behaviour you require for these transactions NOTCHECKED means ReD did not perform any fraud checking for this particular transaction
Surcharge	No	0-9 - ,		0.01 to 100,000.00	Returns the surcharge amount charged and is only present if a surcharge was applied to the transaction.
ExpiryDate	Yes	0-9	4 chars		Expiry date of the card used, in the format MMYYY.
BankAuthCode	No	Aa 0-9	6 chars		The authorisation code returned from the bank. e.g T99777
DeclineCode	No	0-9	2 chars		The decline code from the bank. These codes are specific to the bank. Please contact them for a description of each code. e.g. 00

12.0URLs

The table below shows the complete set of web addresses (URLs) to which you send the transaction registration post.

Environment	URL
TEST	https://test.sagepay.com/gateway/service/vspform-register.vsp
LIVE	https://live.sagepay.com/gateway/service/vspform-register.vsp