



# REMOTE AUTH API INTEGRATION GUIDE

Version 2.5 – Sept 2019

CashFlows™

# TABLE OF CONTENTS

<b>ABOUT THIS GUIDE</b> .....	<b>3</b>
Copyright.....	3
<b>INTRODUCTION</b> .....	<b>4</b>
CASHFLOWS ACQUIRER API .....	4
Security Requirements.....	4
<b>SUBMITTING A PAYMENT REQUEST</b> .....	<b>5</b>
PAYMENT REQUEST PARAMETERS .....	5
Example of a Payment Request (with card number) .....	7
Example of a Payment Request (with a card token).....	8
<b>REMOTE ADMINISTRATION</b> .....	<b>9</b>
VOID REFUND REQUESTS .....	9
Void/Refund Request Parameters .....	9
Example of a Void Request.....	9
Example of a Refund Request.....	9
CREDIT TRANSFERS.....	10
Credit Transfer Request Parameters .....	10
Example of a Credit Transfer.....	10
BATCH RELEASE REQUEST .....	11
Batch Release Request Parameters.....	11
Example of a Batch Release Request.....	11
Batch Release Response .....	12
Examples of a Batch Release Request Response .....	12
Batch Release Query Request .....	12
Example of Batch Release Query Request.....	13
Batch Release Query Response.....	13
<b>RECURRING PAYMENTS</b> .....	<b>15</b>
Account Verification Request Parameters .....	15
Example of an Account Verification Request (with card number) .....	16
Example of an Account Verification Request (with card token).....	16
Example of Account Verification Response.....	16
Continuous Payments Request Parameters .....	17
Example of a Continuous Payment Request .....	18
<b>AUTHORISATION REQUEST RESPONSE</b> .....	<b>19</b>
CVV/AVS check values.....	19
<b>RETRY HANDLING</b> .....	<b>20</b>
Sending a Retry Request .....	20
<b>TESTING YOUR INTEGRATION</b> .....	<b>21</b>
<b>APPENDIX A: ACQUIRER SYSTEM RESPONSE CODE</b> .....	<b>22</b>
Revision History .....	24

## ABOUT THIS GUIDE

Welcome to the CashFlows Acquirer API Integration Guide. This document is designed to provide gateway providers details on how to integrate their gateway into the CashFlows acquirer network and optionally take advantage of our m-commerce voice signature technology.

This document assumes a working knowledge of HTML, HTTP(S) and some programming skills like, Java, PHP, ASP or .Net.

In addition to this guide we have a team of specialists providing technical support during your integration with CashFlows.

The latest version of this guide is always available from: <https://www.cashflows.com/user-and-integration-guides>

### Copyright

2019 © CashFlows Europe Group

While every effort has been made to ensure the accuracy of the information contained in this publication, the information is supplied without representation or warranty of any kind, is subject to change without notice and does not represent a commitment on the part of CashFlows Europe Group. CashFlows Europe Group, therefore, assumes no responsibility and shall have no liability, consequential or otherwise, of any kind arising from this material or any part thereof, or any supplementary materials subsequently issued by CashFlows Europe Group. CashFlows Europe Group has made every effort to ensure the accuracy of this material.

# INTRODUCTION

CashFlows delivers a range of business to business financial services that are provided from a single account, designed to help businesses manage and maximise their cash flow. The CashFlows account enables businesses to offer their customers a full range of payment channels including, online, mobile and mail & telephone orders.

## CASHFLOWS ACQUIRER API

The CashFlows Acquirer API is a mechanism that allows you to collect cardholder and transaction details within your gateway and to submit them directly to CashFlows acquirer network for processing.

### Security Requirements

Using the Acquirer API model to send payment data means that you will be capturing, transmitting, and possibly storing card data. The Card Schemes, Visa Europe and MasterCard International, have never permitted the storage of sensitive data (track data and/or CVV2) post-authorisation, and it is prohibited under 'Requirement 3' of the Payment Card Industry Data Security Standard (PCI DSS). Gateways who store Sensitive Authentication Data (SAD) are being fined by the Card Schemes.

Consequently, if you use the Acquirer API model you will need to demonstrate that your systems can handle this data securely and that you are taking full responsibility for your PCI compliance. One part of this is the need for us to see a clean Vulnerability scan being made on your systems. To view a list of Approved Scanning Vendors, please go to [https://www.pcisecuritystandards.org/qa\\_asv/find\\_one.shtml](https://www.pcisecuritystandards.org/qa_asv/find_one.shtml).

For further information on PCI security standard, please visit <https://www.pcisecuritystandards.org>

# SUBMITTING A PAYMENT REQUEST

The payment request takes the form of a HTTPS **POST** request containing a description of the goods or services being purchased, the total cost, your CashFlows profile Id and the credit card and cardholder details of the consumer. The **POST** request must be UTF-8 encoded and submitted to:

[https://secure-int.cashflows.com/gateway/remote\\_auth](https://secure-int.cashflows.com/gateway/remote_auth) for the Integration environment.

[https://secure.cashflows.com/gateway/remote\\_auth](https://secure.cashflows.com/gateway/remote_auth) for the Production environment.

Please contact [techsupport@cashflows.com](mailto:techsupport@cashflows.com) if you require an integration account.

**Warning:** Our payment service does not have fixed IP addresses and are therefore subject to change. When sending payment requests, you should always point to the DNS record of [secure.cashflows.com](https://secure.cashflows.com) instead.

**Note:** Before you can send payment requests you will be required to provide our Implementations team with the IP address(es) of your payment server(s), so that we can correctly configure your profile.

## PAYMENT REQUEST PARAMETERS

The following table lists the parameters that can be passed to the Acquirer API to request a payment authorisation.

**Note:** All payment request parameters are mandatory unless specified.

PARAMETER	DESCRIPTION
<b>auth_id</b>	Your Profile Id
<b>auth_pass</b>	Authentication password
<b>card_num</b>	Customer's card number ( <i>Must be numeric only with no separators</i> ) ( <i>Conditional, not required where card_token is provided</i> )
<b>card_token</b>	Customer's card token (Max of 50 characters) ( <i>Conditional, not required where card_num is provided</i> )
<b>return_token</b>	If not Null the card_token will be included in the response only when we have processed a successful production payment
<b>card_cvv</b>	Card security code
<b>card_start</b>	Card start date, format is MMY (Optional)
<b>card_issue</b>	Card issue number (Optional)
<b>card_expiry</b>	Card expiry date, format is MMY
<b>cust_name</b>	Customer's name (Optional)
<b>cust_address</b>	Customer's address ( <i>Multiple lines can be separated using the new line break character (ASCII code 10)</i> ) (Optional)
<b>cust_postcode</b>	Customer's post/zip/area code (Optional)
<b>cust_country</b>	Customer's country ( <i>ISO3166 2-character code</i> ) (Optional)
<b>cust_ip</b>	Customer's IP address ( <i>IPV4 Format only</i> ) (Optional)
<b>cust_email</b>	Customer's email address (Optional)
<b>cust_tel</b>	Customer's telephone number (Optional)
<b>tran_ref</b>	Your transaction reference ( <i>e.g. cart ID</i> )
<b>tran_desc</b>	Your transaction description (Max of 99 characters) (Optional)
<b>tran_amount</b>	Transaction amount to 2 decimal places, e.g. 24.99 (The currency symbol must <b>not</b> be included)

<b>tran_currency</b>	Transaction currency, 3-character code
<b>tran_testmode</b>	Transaction test mode. 0
<b>tran_type</b>	Transaction type = sale
<b>tran_class</b>	Transaction class = ecom or moto
<b>tran_recurrence</b>	To be used to override default MID settings: Single transaction with no recurrence = <b>sing</b> Transaction in recurring subscription = <b>subs</b> Transaction in recurring instalment = <b>inst</b> Unscheduled transaction with a stored card = <b>unsc</b> Cardholder initiated transaction with a stored card = <b>card</b> (Optional)
<b>retry_number</b>	Indication of the number of <u>retries attempts</u> , 0 = initial attempt ( <i>For more information refer to <a href="#">Retry Handling</a></i> )
<b>return_acq_ref</b>	If not Null, the Acquirer's Authorisation <i>Request Response number (ARN)</i> will be included in the response only when we have processed a live payment. (Optional)
<b>return_issuer_response_code</b>	If not Null, the raw issuer response code will be included in the response when we have processed a live payment. (Optional)
<b>descriptor</b>	A soft descriptor that is added to your Company Name when displayed on the Cardholders statement (Max of 12 characters) ( <i>Optional</i> )
<b>Additional fields for 3d Secure Transactions</b>	
<b>Version 1:</b>	
<b>acs_3dsversion</b>	The version of 3DS being used for authentication (Optional, when not supplied version <b>1.0.2</b> is assumed)
<b>acs_eci</b>	The response from the Access Control Server, stating the 3DS method and result: VbyV - Full Authentication = 5, VbyV - Attempted Authentication = 6, VbyV - No Authentication = 7, MasterCard SecureCode - Full Authentication = 2 MasterCard SecureCode - Attempted Authentication = 1 MasterCard SecureCode - No Authentication = 0
<b>acs_cavv</b>	The Cardholder Authentication Verification Value from the Access Control Server, 28 characters
<b>acs_xid</b>	The unique Authentication Id for transaction from the Access Control Server, 28 characters
<b>Version 2:</b>	
<b>acs_3dsversion</b>	The version of 3DS being used for authentication (Optional, when not supplied version <b>1</b> is assumed) otherwise <b>2.1.0</b> should be set
<b>acs_eci</b>	The response from the 3DS server. VbyV - Full Authentication = 5, VbyV - Attempted Authentication = 6, VbyV - No Authentication = 7, MasterCard SecureCode - Full Authentication = 2 MasterCard SecureCode - Attempted Authentication = 1 MasterCard SecureCode - No Authentication = 0
<b>acs_cavv</b>	The Cardholder Authentication Verification Value from 3DS server, 28 characters
<b>acs_dstransid</b>	The Directory Server Transaction ID from the 3DS server, 36 characters
<b>Additional fields for merchants with the MCC 6012, 6051 or 7299. (Financial Institutions)</b>	
<b>primary_recipient_dob</b>	Customer's Date of Birth. Format is YYYYMMDD (8 numeric characters)
<b>primary_recipient_surname</b>	Customer's Surname or Last name (2-64-characters alpha characters, including -)
<b>primary_recipient_postcode</b>	Customer's Postcode (2 to 16-characters alpha characters, including spaces)
<b>primary_recipient_account_number</b>	Customer's Account Number (1 to 32 alpha numeric characters, including /-) For PAN Numbers: First 6 and Last 4

**Please note:** Either card\_num or card\_token must be provided.

### Example of a Payment Request (with card number)

You can submit a **POST** request using a range of different programming languages, below is an example of how to submit a payment request using PHP and CURL:

```
<?php
$PaymentUrl = "https://secure.cashflows.com/gateway/remote_auth";
$Post String =
"auth_id=1234&auth_pass=Password&card_num=4000000000000002&card_cvv=123&card_expiry=0121&cust_name=
Testing&cust_address=My%20house%0AMy%20street%0AMy%20Town&cust_postcode=CB22%205LD&cust_country=G
B&cust_ip=123.45.67.89&cust_email=test@test.com&tran_ref=abc123&tran_amount=9.99&tran_currency=GBP&tran_
testmode=0&tran_type=sale&tran_class=ecom&acs_eci=5&acs_cavv=5dbc4a6a39b6730a360e42c3b5f4&acs_xid=ef18
1c0031b5da142e2e8c49424c";

$ch = curl_init($PaymentUrl);
curl_setopt($ch, CURLOPT_POST, 1);
curl_setopt($ch, CURLOPT_POSTFIELDS, $PostString);
curl_setopt($ch, CURLOPT_FOLLOWLOCATION, 1);
curl_setopt($ch, CURLOPT_HEADER, 0);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
$result = curl_exec($ch);
curl_close($ch);
?>
```

The above example **POST** request will send the following payment details to the Acquirer API for authorisation.

```
auth_id=1234&auth_pass=Password&card_num=4000000000000002&card_cvv=123&card_expiry=0121&cust_name=T
esting&cust_address=My%20house%0AMy%20street%0AMy%20Town&cust_postcode=CB22%205LD&cust_country=GB
&cust_ip=123.45.67.89&cust_email=test@test.com&tran_ref=abc123&tran_amount=9.99&tran_currency=GBP&tran_t
estmode=0&tran_type=sale&tran_class=ecom&acs_eci=5&acs_cavv=5dbc4a6a39b6730a360e42c3b5f4&acs_xid=ef181
c0031b5da142e2e8c49424c
```

### Example of a Payment Request (with card token)

You can submit a **POST** request using a range of different programming languages, below is an example of how to submit a payment request using PHP and CURL:

```
<?php
$PaymentUrl = "https://secure.cashflows.com/gateway/remote_auth";
$PostString =
"auth_id=1234&auth_pass=Password&card_token=100000000030419&card_cvv=123&card_expiry=0121&cust_name=Testing&cust_address=My%20house%0AMy%20street%0AMy%20Town&cust_postcode=CB22%205LD&cust_country=GB&cust_ip=123.45.67.89&cust_email=test@test.com&tran_ref=abc123&tran_amount=9.99&tran_currency=GBP&tran_testmode=0&tran_type=sale&tran_class=ecom&acs_eci=5&acs_cavv=5dbc4a6a39b6730a360e42c3b5f4&acs_xid=ef181c0031b5da142e2e8c49424c";

$ch = curl_init($PaymentUrl);
curl_setopt($ch, CURLOPT_POST, 1);
curl_setopt($ch, CURLOPT_POSTFIELDS, $PostString);
curl_setopt($ch, CURLOPT_FOLLOWLOCATION, 1);
curl_setopt($ch, CURLOPT_HEADER, 0);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
$result = curl_exec($ch);
curl_close($ch);
?>
```

The above example **POST** request will send the following payment details to the Acquirer API for authorisation.

```
auth_id=1234&auth_pass=Password&card_token=100000000030419&card_cvv=123&card_expiry=0121&cust_name=Testing&cust_address=My%20house%0AMy%20street%0AMy%20Town&cust_postcode=CB22%205LD&cust_country=GB&cust_ip=123.45.67.89&cust_email=test@test.com&tran_ref=abc123&tran_amount=9.99&tran_currency=GBP&tran_testmode=0&tran_type=sale&tran_class=ecom&acs_eci=5&acs_cavv=5dbc4a6a39b6730a360e42c3b5f4&acs_xid=ef181c0031b5da142e2e8c49424c
```

# REMOTE ADMINISTRATION

To Void, Refund, or Release an 'On Hold' transaction you can either use the administration system or send a request to the Acquirer API.

A Void or Refund request takes the form of a HTTPS **POST** request containing the transaction details that you wish to void or refund. The **POST** request must be UTF-8 encoded and submitted to:

[https://secure.cashflows.com/gateway/remote\\_auth](https://secure.cashflows.com/gateway/remote_auth)

## VOID/REFUND REQUESTS

To void a transaction, you will need to enter all of the transaction details including the full amount of the original transaction and send the Void value in the *trans\_type*. Using the Acquirer API, you can also full or partial refund a transaction by entering the amount that you wish to refund into the *trans\_amount*.

**Note:** You cannot refund more than the original transaction value and are unable to complete a partial refund on the same day that the transaction was made.

### Void/Refund Request parameters

The following table lists the parameters that can be passed to the Acquirer API:

PARAMETER	DESCRIPTION
<b>auth_id</b>	Your Profile Id
<b>auth_pass</b>	Authentication password
<b>tran_amount</b>	Transaction amount to 2 decimal places, e.g. 24.99
<b>tran_currency</b>	Transaction currency, 3-character code, e.g. GBP
<b>tran_testmode</b>	Transaction test mode. 0
<b>tran_type</b>	Transaction type = refund OR void
<b>tran_class</b>	Transaction class = must match the original transaction class
<b>tran_orig_id</b>	Original transaction ID to be refunded or voided
<b>descriptor</b>	A soft descriptor that is added to your Company Name when displayed on the Cardholders statement (Max of 12 characters) (Optional)

### Example of a Void Request

Example of the **POST** string sent in the Void request to the Acquirer API for administration.

```
auth_id=1234&auth_pass=Password&tran_amount=9.99&tran_currency=GBP&tran_testmode=0&tran_type=void&tran_class=ecom&tran_orig_id=01S0001
```

### Example of a Refund Request

Example of the **POST** string sent in the Refund request to the Acquirer API for administration.

```
auth_id=1234&auth_pass=Password&tran_amount=2.99&tran_currency=GBP&tran_testmode=0&tran_type=refund&tran_class=ecom&tran_orig_id=01S0001
```

## CREDIT TRANSFERS

If you have an appropriate 7995 MCC and have Credit Transfers enabled, you can use the API to make a credit transfer request. The credit transfer request takes the form of a HTTPS **POST** containing the amount that you wish to credit a customer who made the original transaction. The **POST** request must be UTF-8 encoded and submitted to:

[https://secure.cashflows.com/gateway/remote\\_auth](https://secure.cashflows.com/gateway/remote_auth)

### Credit Transfer Request parameters

The following table lists the parameters that can be passed to the API:

PARAMETER	DESCRIPTION
<b>auth_id</b>	Your Profile Id
<b>auth_pass</b>	Authentication password
<b>tran_amount</b>	Credit Transfer amount to 2 decimal places, e.g. 24.99 (The currency symbol must not be included)
<b>tran_currency</b>	Transaction currency, 3-character code, e.g. GBP
<b>tran_ref</b>	Your transaction reference ( <i>e.g. cart ID</i> )
<b>tran_testmode</b>	Transaction test mode. 0
<b>tran_type</b>	Transaction type = credit
<b>tran_class</b>	Transaction class = cred
<b>tran_orig_id</b>	Original transaction Id to which the credit will be applied to.
<b>security_hash</b>	A security Hash value used to ensure that no-one has tampered with the credit transfer request
<b>descriptor</b>	A soft descriptor that is added to your Company Name when displayed on the Cardholders statement (Max of 12 characters) (Optional)

### Example of a Credit Transfer

Example of the **POST** string sent in the Credit Transfer request to the API for administration.

```
auth_id=1234&auth_pass=Password&tran_amount=2.99&tran_currency=GBP&tran_ref=Payout1234tran_testmode=0
&tran_type=credit&tran_class=cred&tran_orig_id=01S0001234&security_hash=
e5446ea59340d867af9fed6ba92f267e17d0119c7d972d7d84c0ab31ee4b1708
```

To protect the Credit Transfer from being tampered with whilst being transferred you **must** include a cryptographic hash digital signature.

The digital signature or 'message digest' needs to be created by your own server side scripting using the SHA256 algorithm method and contain the following values:

```
tran_type:tran_amount:tran_currency:tran_orig_id:tran_ref:[secret key]
```

Each section of data is separated using a ':' (colon) character, and the data must be organised in the exact sequence shown.

The 'message digest' is then be included into your credit transfer request using the *security\_hash* parameter.

CashFlows compares the ‘message digest’ against its own ‘message digest’ created from your credit transfer details supplied. As only you and the CashFlows know the secret key element of the ‘message digest’, the credit transfer will only be processed if the two ‘message digest’ match.

**Warning:** At no time should the actual pre-set secret key be included in any FORM or web page that is held on your server.

**Note:** For non 7995 Merchants that have a requirement to use this service, they may make a Credit Transfer request using an initial Account Verification as their original transaction id. To make an Account Verification request, please refer to Recurring Payment section of this guide.

## BATCH RELEASE REQUEST

Prior to the funds of a transaction being requested from the bank it is possible to place them on Hold on the day that they have been authorised and for up to 7 days.

To release a single or multiple transactions that have been placed ‘On Hold’ you will need send a MIME multipart **POST** request to the Acquirer API. The **POST** request must be UTF-8 encoded and submitted to:

[https://secure.cashflows.com/gateway/remote\\_batch](https://secure.cashflows.com/gateway/remote_batch)

The multipart **POST** request contains two parts, the first includes the following parameters to instruct the system of your batch request, and the second includes the attachment that has the transaction references that you wish to release from being on Hold.

**Warning:** If the transaction is not released within the 7 days it will expire and will require to be authorised again.

### Batch Release Request parameters

The following table lists the parameters used to request a batch release to the Acquirer API:

PARAMETER	DESCRIPTION
profile_id	Your Profile Id
profile_pass	Authentication password
batch_op	Type of Batch operation. For a batch release request the value must be ‘onhold-release-submit’
attached_type	Defines the format of the attachment. This must be set to ‘onhold_v0’

The second part of the **POST** header contains the batch file containing all of the transaction references that you wish to release. The batch file must be in either a CSV, or TXT file formats as specified in the request’s Content-Disposition filename.

### Example of a Batch Release Request

Example of the **POST** header sent in the batch release request to the Acquirer API for administration.

```
POST /gateway/remote_batch HTTP/1.0
Content-Type: multipart/x-vcg-remote-api; boundary=_partBoundary_
Content-Length: 323
```

The following part of the **POST** contains the instructions of the batch release request:

```
--_partBoundary_
Content-Type: application/x-www-form-urlencoded
profile_id=73&profile_pass=password1234&attached_type=onhold_v0&batch_op=onhold-release-submit
```

The last part of the **POST** contains the details of the attachment that includes the transaction reference that you wish to release:

```
--_partBoundary_
Content-Type: text/csv
Content-Disposition: attachment;filename="releaseRefs.csv"
01S00001724
01S00001725
01S00001726
--_partBoundary_--
```

### Batch Release response

After you have sent a batch release request to the Acquirer API, the response of the batch upload will contain one of following results:

RESULTS	DESCRIPTION
invalid_request	Error – Request cannot be parsed correctly
invalid_credentials	Error – Cannot verify the Profile Id or Authentication password
request_toobig	Error – The batch request is larger than 64k for the Content-Length
invalid_filename	Error – The attachment filename is not valid
internal_failure	Error – There has been an internal error, please try again
release_report	Success – The batch request has been successfully uploaded and a batch Id has been created

### Examples of a Batch Release Request response:

Example of an invalid request response:

```
--remote_batch-4C4100AA
Content-Type: application/x-www-form-urlencoded
result=invalid_request
--remote_batch-4C4100AA--
```

### Example of a response for a successful batch release request:

```
--remote_batch-4C4100C6
Content-Type: application/x-www-form-urlencoded
result=release_report&batch_id=26&batch_status=pending
--remote_batch-4C4100C6--
```

When a batch release request has been successfully uploaded the response will display a batch Id number which will enable you to query the status of the batch after the initial request.

### Batch Release Query Request

After uploading your batch release file, the system will take around 5 minutes depending of the size of the file to complete the release of the transactions. To query the status of a batch release request that has been uploaded, you can periodically poll the service using a batch release **POST** query request.

To submit a batch release query request, you must **POST** the following parameters to the Acquirer API:

PARAMETER	DESCRIPTION
<b>profile_id</b>	Your Profile Id
<b>profile_pass</b>	Authentication password
<b>batch_id</b>	The Id of the batch that you wish to query
<b>batch_op</b>	Type of Batch operation. For a batch release query request the value must be 'onhold-release-query'

#### Example of a Batch Release Query Request

Example of the **POST** header sent in the batch release query request to the Acquirer API.

*POST /admin/remote\_batch HTTP/1.0*

*Content-Type: multipart/x-vcg-remote-api; boundary=\_partBoundary\_*

*Content-Length: 172*

*--\_partBoundary\_*

*Content-Type: application/x-www-form-urlencoded*

*profile\_id=73&profile\_pass=password1234&batch\_id=26&batch\_op=onhold-release-query*

*--\_partBoundary\_--*

#### Batch Release Query response

After you have sent a batch release query request to the Acquirer API, the response will contain one of following results:

QUERY RESULTS	DESCRIPTION
<b>invalid_request</b>	Error – Request cannot be parsed correctly
<b>invalid_credentials</b>	Error – Cannot verify the Profile Id or Authentication password
<b>internal_failure</b>	Error – There has been an internal error, please try again
<b>release_notfound</b>	Error – Cannot find the requested batch Id
<b>release_report</b>	Success – The batch query request has been successfully submitted and a batch has been found

If the query was successfully submitted (i.e. result=release\_report) the response will return a batch\_id and batch\_status of either pending, processing or complete. If the status of the batch is 'complete', the following additional information and an attachment providing status of each of the transactions will be included in the multipart response.

BATCH COMPLETE PARAMETERS	DESCRIPTION
<b>item_count</b>	Total number of items in the batch release
<b>item_succ</b>	Number of transactions that have been successfully released
<b>item_fail</b>	Number of transactions that have failed and not been released
<b>attachment_type</b>	Defines the format of the attachment

In the attachment part of the multipart response each transaction will contain one of the following results:

TRANSACTION RESULTS	DESCRIPTION
<b>batchrel_notonhold</b>	The transaction was not on hold at the time of the request as the transaction has been previously released and may have been already settled
<b>batchrel_invalref</b>	The transaction could not be found as it has an invalid reference
<b>batchrel_expired</b>	The transaction has expired and therefore has not been sent of authorisation
<b>batchrel_error</b>	There was an internal error, please resubmit this transaction release request
<b>batchrel_ok</b>	The transaction has been successfully released for authorisation

#### *Example of a successful Batch Release Query response*

Example of the multipart response you receive for a successful batch release query request. The first part of the response shows the details of the successfully query:

*--remote\_batch-4C4106B0*

*Content-Type: application/x-www-form-urlencoded*

*result=release\_report&batch\_id=26&batch\_status=complete&item\_count=4&item\_succ=4&item\_fail=0&attached\_type=onhold\_v0*

The final part of the multipart response shows the results of each of the transactions that where requested to be released.

*--remote\_batch-4C4106B0*

*Content-Type: text/csv*

*01S00001724,batchrel\_expired*

*01S00001725,batchrel\_ok*

*01S00001726,batchrel\_ok*

*--remote\_batch-4C4106B0--*

## RECURRING PAYMENTS

Using the Acquirer API, you can submit recurring payments, using an approach called continuous authority. To submit a recurring payment, you will first require getting an initial account verification of a consumer's card details including the card's CVV. This account verification request is then checked and if successful, authorised. We then store the card details securely on our PCI approved system and send you a request response containing an account verification Id. When sending a continuous (recurring) payment request you must include this account verification Id to enable us to use the initially stored card details to send the payment for authorisation.

### Account Verification Request parameters

The following table lists the account verification request parameters that must be passed to the API:

PARAMETER	DESCRIPTION
<b>auth_id</b>	Must be set to the Profile ID
<b>auth_pass</b>	Authentication password
<b>card_num</b>	Customer's card number <i>Must be numeric only with no separators (Optional)</i>
<b>card_token</b>	Customer's card token (Max of 50 characters) (Optional)
<b>return_token</b>	If not Null the card_token will be included in the response only when we have processed a live payment
<b>card_cvv</b>	Card security code
<b>card_start</b>	Card start date, format is MMY (Optional)
<b>card_issue</b>	Card issue number (Optional)
<b>card_expiry</b>	Card expiry date, format is MMY
<b>cust_name</b>	Customer's name (Optional)
<b>cust_address</b>	Customer's address (Multiple lines can be separated using the new line break character (ASCII code 10)) (Optional)
<b>cust_postcode</b>	Customer's post/zip/area code (Optional)
<b>cust_country</b>	Customer's country, ISO3166 2-character code (Optional)
<b>cust_ip</b>	Customer's IP address (Optional) (IPV4 Format only)
<b>cust_email</b>	Customer's email address (Optional)
<b>cust_tel</b>	Customer's telephone number (Optional)
<b>tran_ref</b>	Your transaction reference (e.g. cart ID)
<b>tran_desc</b>	Your transaction description (Optional)
<b>tran_currency</b>	Transaction currency, 3-character code (For a list of currencies code you can use / accept, please contact support@cashflows.com)
<b>tran_testmode</b>	Transaction test mode. 0
<b>tran_type</b>	Transaction type = verify
<b>tran_class</b>	Transaction class = ecom
<b>retry_number</b>	Indication of the number of retries attempts, 0 = initial attempt (For more information refer to <a href="#">Retry Handling</a> )
<b>return_acq_ref</b>	If not Null the Acquirer's Authorisation Request Response number (ARN) will be included in the response only when we have processed a live payment

<b>descriptor</b>	A soft descriptor that is added to your Company Name when displayed on the Cardholders statement (Max of 12 characters) <i>(Optional)</i>
<b>tran_recurrence</b>	To be used to override default MID settings: Single transaction with no recurrence = <b>sing</b> Transaction in recurring subscription = <b>subs</b> Transaction in recurring instalment = <b>inst</b> Unscheduled transaction with a stored card = <b>unsc</b> Cardholder initiated transaction with a stored card = <b>card</b> (Optional)
<b>Additional fields for merchants with the MCC 6012, 6051 or 7299. (Financial Institutions)</b>	
<b>primary_recipient_dob</b>	Customer's Date of Birth. Format is YYYYMMDD (8 numeric characters)
<b>primary_recipient_surname</b>	Customer's Surname or Last name (2-64 characters alpha characters, including -)
<b>primary_recipient_postcode</b>	Customer's Postcode (2 to 16-characters alpha characters, including spaces)
<b>primary_recipient_account_number</b>	Customer's Account Number (1 to 32 alpha numeric characters, including /-) For PAN Numbers: First 6 and Last 4. <i>(Never include full PAN in primary_recipient_account_number field.)</i>

**Please note:**

- 1) One of card\_num or card\_token must be provided.

**Example of an Account Verification Request (with card number)**

Example of the **POST** string sent in the account verification request to the API for authorisation.

```
auth_id=1234&auth_pass=Password&card_num=4000000000000002&card_cvv=123&card_expiry=0121&cust_name=Testing&cust_address=My%20house%0AMy%20street%0AMy%20Town&cust_postcode=CB22%205LD&cust_country=GB&cust_ip=123.45.67.89&cust_email=test@test.com&tran_ref=abc123&tran_currency=GBP&tran_testmode=0&tran_type=verify&tran_class=ecom
```

**Example of an Account Verification Request (with card token)**

Example of the **POST** string sent in the account verification request to the API for authorisation.

```
auth_id=1234&auth_pass=Password&card_token=100000000030419&card_cvv=123&card_expiry=0121&cust_name=Testing&cust_address=My%20house%0AMy%20street%0AMy%20Town&cust_postcode=CB22%205LD&cust_country=GB&cust_ip=123.45.67.89&cust_email=test@test.com&tran_ref=abc123&tran_currency=GBP&tran_testmode=0&tran_type=verify&tran_class=ecom
```

**Note:** An Account Verification request checks if the account is valid, it will not perform a check for available funds on the account and is not an authorisation of a sale.

**Example of an Account Verification Response**

Example of the account verification response sent to you after submitting an account verification request:

EXAMPLE RESPONSE	MEANING
A/05P00001724 232 031971 Authorised	Authorised: A Account Verification Id: 05P00001724 CVV/AVS:232 Authorisation code: 031971

This includes the account verification Id denoted with a 05P prefix and the CVV/AVS check response. A continuous authorised payment request can only be performed where the CVV comparison check has been returned as a MATCH (i.e. the first check value must be a 2), irrespective of the authorisation status of the Account Verification.

For more information about request responses, please refer to: [Authorisation Request Response](#).

### Continuous Payment request parameters

To send a continuous payment request you must **exclude** *card\_num* (or *card\_token*), *card\_expiry* & *card\_cvv* parameter, and include the *tran\_orig\_id* which has the value of initial *verification* or *sale Id*. The following table lists the continuous payment request parameters that must be passed to the API:

PARAMETER	DESCRIPTION
<b>auth_id</b>	Must be set to the Profile ID
<b>auth_pass</b>	Authentication password
<b>cust_name</b>	Customer's name (Optional)
<b>cust_address</b>	Customer's address ( <i>Multiple lines can be separated using the new line break character (ASCII code 10)</i> ) (Optional)
<b>cust_postcode</b>	Customer's post/zip/area code (Optional)
<b>cust_country</b>	Customer's country, ISO3166 2-character code (Optional)
<b>cust_ip</b>	Customer's IP address ( <i>IPV4 Format only</i> ) (Optional)
<b>cust_email</b>	Customer's email address (Optional)
<b>cust_tel</b>	Customer's telephone number (Optional)
<b>tran_ref</b>	Your transaction reference (e.g. cart ID)
<b>tran_desc</b>	Your transaction description (Optional)
<b>tran_amount</b>	Transaction amount to 2 decimal places, e.g. 24.99 ( <i>The currency symbol must <b>not</b> be included</i> )
<b>tran_currency</b>	Transaction currency, 3-character code
<b>tran_testmode</b>	Transaction test mode. 0
<b>tran_type</b>	Transaction type = sale
<b>tran_class</b>	Transaction class = cont
<b>tran_orig_id</b>	Verification ID or Sales ID (e.g. <i>05P00001724</i> or <i>06S00001724</i> )
<b>retry_number</b>	Indication of the number of retries attempts, 0 = initial attempt ( <i>For more information refer to <a href="#">Retry Handling</a></i> )
<b>return_acq_ref</b>	If not Null the Acquirer's Authorisation Request Response number (ARN) will be included in the response only when we have processed a live payment
<b>descriptor</b>	A soft descriptor that is added to your Company Name when displayed on the Cardholders statement. (Max of 12 characters) ( <i>Optional</i> )
<b>tran_recurrence</b>	To be used to override default MID settings: Single transaction with no recurrence = <b>sing</b> Transaction in recurring subscription = <b>subs</b> Transaction in recurring instalment = <b>inst</b> Unscheduled transaction with a stored card = <b>unsc</b> Cardholder initiated transaction with a stored card = <b>card</b> (Optional)
<b>Additional fields for merchants with the MCC 6012, 6051 or 7299. (Financial Institutions)</b>	
<b>primary_recipient_dob</b>	Customer's Date of Birth. Format is YYYYMMDD (8 numeric characters)
<b>primary_recipient_surname</b>	Customer's Surname or Last name (2-64 characters alpha characters, characters, including -)
<b>primary_recipient_postcode</b>	Customer's Postcode (2 to 16-characters alpha characters, including spaces)
<b>primary_recipient_account_number</b>	Customer's Account Number (1 to 32 alpha numeric characters, including

**Example of a Continuous Payment Request**

Example of the **POST** string sent in the continuous payment request to the Acquirer API for authorisation.

```
auth_id=1234&auth_pass=Password&cust_name=Testing&cust_address=My%20house%0AMy%20street%0AMy%20To  
wn&cust_postcode=CB22%205LD&cust_country=GB&cust_ip=123.45.67.89&cust_email=test@test.com&tran_ref=abc1  
23&tran_amount=9.99&tran_currency=GBP&tran_testmode=0&tran_type=sale&tran_class=cont&tran_orig_id=05P00  
001724
```

# AUTHORISATION REQUEST RESPONSE

The response consists of the authorisation status code, transaction ID, CVV/AVS result, authorisation code, authorisation message and Authorisation Request Response Number (ARN). These fields are separated using the vertical bar character. An authorisation status of 'A' indicates that the transaction was authorised, anything else indicates that it was not.

EXAMPLE RESPONSE	MEANING
<i>A 01S00001724 232 031971 Authorised 74698692333601146452212 100000000030419</i>	<i>Authorised: A Transaction Id: 01S00001724 CVV/AVS:232 Authorisation code: 031971 Authorisation Request Response: 74698... Card token: 100000000030419</i>
<i>D 01S00001723 400 D102 Not Authorised 74698692333601146452212</i>	<i>Not authorised: D Transaction Id: 01S00001723 CVV/AVS:400 Authorisation Request Response: 74698...</i>
<i>V 99E5D84B40F 000 V226 Invalid request</i>	<i>Invalid request: V Transaction Id: 99E5D84B40F CVV/AVS:000 (please refer to Appendix A for further information.)</i>
<i>B 01S00632BE2 000 D090 Not authorised</i>	<i>Blocked: D Transaction Id: 01S00632BE2 CVV/AVS:000 (please refer to Appendix A for further information.)</i>

## CVV/AVS check values

The CVV/AVS result is a 3-digit value, each digit representing a different check. The first value is the CVV check, the second is the address and the third is the postcode. The possible values for each digit are as follows:

VALUE	MEANING
0	Not Checked
1	Check was not available
2	Full match
3	Partial match
4	Not matched
5	Error

A partial match is only possible for the address or postcode data, not for CVV check. Not all acquirers or issuers support all of these checks, in which case the results will be either 0 or 1.

Example Response	CVV	Address	Postcode
232	Full match	Partial match	Full match
400	Not matched	Not checked	Not checked

## RETRY HANDLING

If there are any network problems, timing or connections issues our system will return a response code informing you of the issue. For the full list of response codes, please refer to: [Appendix A](#).

If you receive any of the following response codes, you should retry your authorisation request as the retry will return a different response to the original request:

- *S201: API to gateway connect fail*
- *S203: API layer timeout*
- *V249: Duplicate transaction still processing*

In addition to the above response codes the following response may be deemed appropriate for a 'retry':

If the system hasn't been able to attempt to send the request of authorisation you will be returned the following response codes. If you resubmit the request for authorisation you can do so freely without risk of *double authorisation*.

- *S001 or S101: Connection failure*

If the system cannot determine whether an attempted authorisation has been successful or not, the system will return the following response codes.

- *S003 or S103: Response Timeout*
- *S002 or S102: Invalid response*

### **Sending a retry request**

To submit a retry request enter a value greater than zero into the *retry\_number* parameter and resubmit the authorisation request.

**Note:** For this functionality to work correctly, all transactions must have a unique *tran\_ref* and must be submitted within 5 minutes of the initial authorisation request.

When our system receives a retry request, you will be presented with one of results:

- If the retry request is a duplicate request of a finished transaction, then the original transaction response is replayed, or
- If original transaction is still being processed you will receive the *V249* response code informing you of such, or
- If our system has no record of a previous duplicate transaction request, then the transaction is processed and the results returned (as though it was the first attempt).

## TESTING YOUR INTEGRATION

You can test your integration to our Acquirer API by setting your **POST** request to the Integration environment and use the test cards listed below:

CARD NUMBER	TOKEN	EXPIRY DATE	CVV
4000000000000002 (VISA)	1000000000030419	Any valid expiry date (mm/yy)	123
4462030000000000 (VISA prepaid)	1000000000030554	Any valid expiry date (mm/yy)	444
555555555554444 (MasterCard)	1000000000030567	Any valid expiry date (mm/yy)	321
5597507644910558 (MasterCard prepaid)	1000000000030568	Any valid expiry date (mm/yy)	888

**Warning:** Test card numbers will only work in the Integration environment, if used in Production environment an error will be returned.

# APPENDIX A: ACQUIRER SYSTEM RESPONSE CODES

Status 'A' is authorised, anything else is not. The auth code and auth message for authorised transactions cannot be predicted (as they can change from one bank/issuer to the next).

'V' is a validation error (e.g. invalid card number)

'D' is a decline

'R' is a referral (has to be treated as a decline)

'B' is a blocked transaction

'C' is a cancelled transaction (e.g. user pressed cancel on payment page)

'S' is a system error

These will be followed by a 3-digit code; the first digit is an internal code which can be ignored. The second two digits are the actual error code for the given status. Attached is a list of the current error codes. (Please note this list is subject to change).

The list is given as, for example, Vx01 which means it is the result for V101, V201, V301 etc.

CODE	REASON
Vx01	Invalid merchant details
Vx02	Invalid expiry date
Vx03	Invalid start date
Vx04	Invalid issue number
Vx05	Invalid CVV
Vx06	Invalid card number
Vx07	Card holder name not set
Vx08	Insufficient address details
Vx09	Invalid country code
Vx10	Invalid cart ID
Vx11	Invalid email address
Vx12	Invalid phone number
Vx13	Invalid amount
Vx14	Invalid currency code
Vx15	Invalid customer IP
Vx16	Original trans not found
Vx17	Invalid merchant IP
Vx18	Unknown transaction type
Vx19	Card number changed
Vx20	Currency changed
Vx21	Original trans ref required
Vx22	Amount exceeds original
Vx23	Can not refund this type of transaction
Vx24	Amount changed
Vx25	User account details required
Vx26	Invalid request
Vx27	Original trans not pre-auth
Vx28	Transaction mode changed
Vx29	Card/Currency combination not supported
Vx30	Unknown card type
Vx31	Issue number required
Vx32	Issue number not required
Vx33	Duplicate transaction

<b>Vx34</b>	Unable to void transaction
<b>Vx35</b>	Original trans was not authorised
<b>Vx36</b>	Invalid PIN
<b>Vx37</b>	Unknown transaction class
<b>Vx38</b>	Original transaction type does not match
<b>Vx39</b>	Card expired
<b>Vx40</b>	CVV Required
<b>Vx41</b>	Original transaction already settled
<b>Vx42</b>	Original transaction already cancelled
<b>Vx43</b>	This card does not support the required transaction type
<b>Vx44</b>	Transaction details do not match original
<b>Vx48</b>	User Details not valid
<b>Vx52</b>	3DS Not Enabled
<b>Vx53</b>	3DS Data Invalid
<b>Vx54</b>	Concurrent Authorisations
<b>Vx55</b>	Invalid Funds Recipient Date (MCC 6012, 6051 or 7299 merchants)
<b>Vx56</b>	Terminal mismatch
<b>Vx57</b>	Transaction not allowed on this card
<b>Vx58</b>	Original transaction requires 3DS attempt/auth
<b>Vx59</b>	ECOM transactions require 3DS attempt/auth
<b>Vx60</b>	Verify for Amex card not supported
<b>Vx61</b>	Recurrence Flag usage invalid
<b>Vx62</b>	Initial Sale/Verify ARN missing for subsequent sale
<b>Vx63</b>	Initial Sale/Verify for subsequent sale not approved
<b>Vx64</b>	Initial transaction on card expired
<b>Dx01</b>	Non-specific decline
<b>Dx02</b>	Declined due to funds (insufficient/limit exceeded)
<b>Dx03</b>	Retain card response
<b>Dx05</b>	On our blacklist
<b>Dx07</b>	Live/test mismatch
<b>Dx08</b>	Refund: Insufficient merchant funds in account
<b>Dx10</b>	Card authorisation attempt limit reached
<b>Dx11</b>	Monthly Scheme Decline Rate limit reached
<b>Dx40</b>	Continuous Authority cancelled for the transaction
<b>Dx41</b>	Continuous Authorities cancelled for the merchant
<b>Dx43</b>	Continuous Authorities cancelled for the card
<b>Dx90</b>	Pre-Authorisation anti-fraud block
<b>Dx91</b>	Post-Authorisation anti-fraud block
<b>Rx01</b>	Not Authorised
<b>Ex01</b>	Transaction error
<b>Cx01</b>	Transaction cancelled
<b>Cx02</b>	Transaction expired
<b>Sx00</b>	Invalid transaction Request
<b>Sx01</b>	Connection failure
<b>Sx02</b>	Invalid response
<b>Sx03</b>	Response timeout
<b>Sx04</b>	Server error
<b>Sx05</b>	Server error
<b>Sx06</b>	No response from issuer
<b>Sx07</b>	Service not available
<b>Sx99</b>	Unknown Error

## REVISION HISTORY

Date	Summary of Changes	Version No.
18/06/2019	Updated with card tokenisation changes. Addition of new validation error codes from Vx56 through to Vx64.	2.4