Cashflows

Remote Auth API Integration Guide

Version 3.2 – November 2023

Contents

Introduction

Welcome to the Cashflows Remote Auth API Integration Guide. This guide provides details on integrating into the Cashflows acquirer network. The Cashflows Remote Auth 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.

Sensitive data and PCI-DSS

Using the Remote Auth API model to send payment data means that you will be capturing, transmitting, and possibly storing card data.

The storage of Sensitive Authentication Data (track data and/or CVV2) post-authorisation is prohibited by Visa and Mastercard, as well as Requirement 3 of the Payment Card Industry Data Security Standard (PCI-DSS).

If you use Account Updater you need to demonstrate your systems handle this data securely and that you take full responsibility for your PCI compliance. This includes, but is not limited to, providing your current Attestation of Compliance certificate and evidence of a recent clean vulnerability scan.

A list of approved Security Assessors can be found at:
https://www.pcisecuritystandards.org/assessors_and_solutions/qualified_security_assessors.

For more information on PCI security standards, see https://www.pcisecuritystandards.org.

Submit a payment request

To request a payment, you need to submit a HTTPS **POST** request with a description of the goods or services being purchased, the total cost, your Cashflows profile ID, the card details, and the cardholder's details. The request must be UTF-8 encoded and submitted to:

- Test - https://secure-int.cashflows.com/gateway/remote_auth.
- Live - https://secure.cashflows.com/gateway/remote_auth.

Before you can send payment requests you need to send our Implementations team the IP addresses of your payment servers so that we can configure your profile.

Please contact techsupport@cashflows.com if you require an integration account.

Warning – Our payment services do not have fixed IP addresses and may change, so we recommend directing your requests to the DNS record of secure.cashflows.com.

Payment request parameters

To submit a payment request you need to send a request with the mandatory parameters. There are also additional, optional parameters you can include, see the API Reference for more information.

The mandatory parameters are:

Parameter	Description
`auth_id`	Your Profile Id
`auth_pass`	Authentication password
`card_num`	Customer's card number (Must be numeric only with no separators) (**Conditional**, not required if `card_token` provided)
`card_token`	Customer's card token (Max of 50 characters) (**Conditional**, not required if `card_num` provided)
`card_cvv`	Card security code
`card_expiry`	Card expiry date, format is MMYY
`tran_ref`	Your transaction reference *(e.g. cart ID)*
`tran_amount`	Transaction amount to 2 decimal places, e.g. 24.99
`tran_currency`	Transaction currency (3-character code)
`tran_testmode`	Transaction test mode = 0
`tran_type`	Transaction type = sale
`tran_class`	Transaction class = ecom or moto
`retry_number`	Indication of the number of retries attempts, 0 = initial attempt (See Retry Handing for more information)
`return_token`	If not Null the `card_token` will be included in the response only when we have processed a successful transaction

If you’re using 3D Secure with Visa, Mastercard, or American Express, you must also include:

Parameters (3D Secure)	Description
`acs_eci`	The response from the 3DS server. • **5** = VbyV **-** Full Authentication • **6** = VbyV **-** Attempted Authentication • **7** = VbyV - No Authentication • **2** = MasterCard SecureCode **-** Full Authentication • **1** = MasterCard SecureCode **-** Attempted Authentication • **0** = MasterCard SecureCode **-** No Authentication • **05** = American Express Safekey - Full Authentication • **06** = American Express Safekey - Attempted Authentication • **07** = American Express Safekey - No Authentication
`acs_cavv`	The Cardholder Authentication Verification Value from 3DS server, 28 Characters **American Express Safekey** – provide the `American Express Verification Value` (AEVV) – 20 characters long.
`acs_dstransid`	The universally unique transaction identifier assigned by the Directory Server (DS) to identify a single transaction, 36 characters. Required when acs_3dsversion = 2.1.0/2.2.0. **American Express Safekey** – provide the `American Express Safekey Transaction ID` (XID) – 20 characters long.

If your MCC is 6012, 6051, or 7299 (financial institutions) you must also include:

Parameters (financial institutions)	Description
`primary_recipient_dob`	Customer’s Date of Birth. Format is YYYYMMDD (8 numeric characters)
`primary_recipient_surname`	Customer’s Surname or Last name (2-64-characters alpha characters, including -)
`primary_recipient_postcode`	Customer’s Postcode (2 to 16-characters alpha characters, including spaces)
`primary_recipient_account_number`	Customer’s Account Number (1 to 32 alpha numeric characters, including /-) For PAN Numbers: First 6 and Last 4

Example 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
_n ame=Testing&cust_address=My%20house%0AMy%20street%0AMy%20Town&cust_postcode=CB22%205LD&cu
st_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=5dbc4a6a3
9b6730a360e42c3b 5f4&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 submits:

```
auth_id=1234&auth_pass=Password&card_num=4000000000000002&card_cvv=123&card_expiry=0121&cust_
n ame=Testing&cust_address=My%20house%0AMy%20street%0AMy%20Town&cust_postcode=CB22%205LD&cu
st_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=5dbc4a6a3
9b6730a360e42c3b 5f4&acs_xid=ef18 1c0031b5da142e2e8c49424c";
```

Example 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=1000000000030419&card_cvv=123&card_expiry=0121&cus
t_n ame=Testing&cust_address=My%20house%0AMy%20street%0AMy%20Town&cust_postcode=CB22%205LD&cu
st_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=5dbc4a6a3
9b6730a360e42c3b 5f4&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 submits:

```
auth_id=1234&auth_pass=Password&card_token=1000000000030419&card_cvv=123&card_expiry=0121&cus
t_n ame=Testing&cust_address=My%20house%0AMy%20street%0AMy%20Town&cust_postcode=CB22%205LD&cu
st_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=5dbc4a6a3
9b6730a360e42c3b 5f4&acs_xid=ef18 1c0031b5da142e2e8c49424c";
```

Submitting an eWallet payment

The Remote Auth API also supports the leading digital wallet providers:

- Apple Pay
- Google Pay
- Samsung Pay

To submit an eWallet payment you need to provide these parameters when submitting your payment:

Parameters	Description
`ewallet`	Indicates whether a Pay's wallet was used: true | false
`ewallet_type`	Indicates which Pay's wallet was used Accepted values: (case_sensitive): • **applepay** • **googlepay** • **samsungpay** • **other**

If these parameters are not provided the request will process as a card transaction.

Void or refund a transaction

To void or refund a transaction you can either use the administration system or send a request to the Remote Auth API. To make the request through the API you need to submit a HTTPS **POST** with the desired transaction details. The request must be UTF-8 encoded and submitted to:

- Test - https://secure-int.cashflows.com/gateway/remote_auth.
- Live - https://secure.cashflows.com/gateway/remote_auth.

Voiding a transaction stops it from being settled. If a transaction has already been settled, you'll need to request a refund instead.

To submit a void or refund request you need to send a request with the `trans_type` field set to either `void` or `refund`. You also need to provide the original transaction information, including the amount. To process a partial refund set the `trans_amount` to less than the original transaction 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 and refund request parameters

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

Example void request

Example of the **POST** string sent in the Void request to the Remote Auth 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 refund request

Example of the **POST** string sent in the Refund request to the Remote Auth API for administration:

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

Credit transfers

If you have Credit Transfers enabled, you can use the Remote Auth API to make a credit transfer request. To request a credit transfer, submit a HTTPS **POST** with the amount that you wish to credit. The request must be UTF-8 encoded and submitted to:

- Test - https://secure-int.cashflows.com/gateway/remote_auth.
- Live - https://secure.cashflows.com/gateway/remote_auth.

Credit transactions are only supported for these MCCs:

- 5262 – Marketplaces
- 6010 - Financial Institutions – Manual Cash Disbursements
- 6011 - Financial Institutions – Automated Cash Disbursements
- 6012 - Financial Institutions – Merchandise, Services, and Debt Repayment
- 6300 - Insurance Sales, Underwriting, and Premiums
- 6399 - Insurance, Not Elsewhere Classified
- 7994 - Game of skill
- 7995 - Gambling
- 8999 - Professional Services (Not Elsewhere Classified)

Contact your account manager to confirm if Credit Transfers have been enabled on your account.

To protect credit transfers whilst being transferred you **must** include a cryptographic hash digital signature.

The digital signature or ‘message digest’ must 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]
```

For Visa card Credit Transfers, where `tran_orig_id` is not supplied in the request, this parameter must be omitted from the hash string:

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

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

The ‘message digest’ can then be included in your credit transfer request using the `security_hash` parameter.

We compare the ‘message digest’ against our own ‘message digest’ created from the supplied credit. 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 pre-set secret key be included in any FORM or web page that is held on your server.

Credit transfer request parameters

To submit a credit transfer request, you need to send a request with the mandatory parameters. There are also additional, optional parameters you can include, see the API Reference for more information.

The mandatory parameters are:

Parameters	Description
`auth_id`	Your Profile Id
`auth_pass`	Authentication password
`tran_amount`	Credit Transfer amount to 2 decimal places, e.g. 24.99 (The currency symbol must not be included)
`tran_currency`	Transaction currency, 3-character code, e.g. GBP
`tran_ref`	Your transaction reference *(e.g. cart ID)*
`tran_testmode`	Transaction test mode. 0
`tran_type`	Transaction type = credit
`tran_class`	Transaction class = cred
`tran_orig_id`	**Mandatory for Mastercard**. Optional for Visa (where `card_num` or `card_token` is provided) Original transaction id to which the credit will be applied to.
`card_num`	**Visa only**: Customer's card number (Must be numeric only with no separators) (**Conditional**, not required if `tran_orig_id` or `card_token` is provided)
`card_token`	**Visa only**: Customer's card token (Max of 50 characters) (**Conditional**, not required if `card_ num` or `tran_orig_id` is provided)
`security_hash`	A security Hash value used to ensure that no-one has tampered with the credit transfer request

Example credit transfer request

Example of the **POST** string sent in the Credit Transfer request to the Remote Auth API for administration:

```
auth_id=1234&auth_pass=Password&tran_amount=9.99&tran_currency=GBP&tran_testmode=0&tran_type=c
redit&tran_class=cred&tran_orig_id=01S0001234&security_hash=e5446ea59340d867af9fed6ba92f267e17
d0119c7d972d7d84c0ab31ee4b1708
```

Release a transaction

Before the funds of a transaction are requested from the bank it is possible on the date of the transaction to place them **On Hold** for up to 7 days.

To release transactions that have been placed on hold you need to send a MIME multipart **POST** request to the Remote Auth API. The request must be UTF-8 encoded and submitted to:

- Test - https://secure-int.cashflows.com/gateway/remote_auth.
- Live - https://secure.cashflows.com/gateway/remote_auth.

Warning - If the transaction is not released within the 7 days, it will expire and will need to be authorised again.

The **POST** request contains two parts, the first includes instructions to the system for your batch request, and the second includes the attachment with transaction references that you want releasing from hold.

These are the parameters used in the first part of a batch release request to the Remote Auth API:

Parameters	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 the transaction references that you wish to release. The batch file must be in either a .csv or .txt format as specified in the request's Content-Disposition filename.

Example batch release request

Example of the **POST** header sent in the batch release request to the Remote Auth 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:

```
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:

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

Batch release response

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

Parameters	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

Example batch release request responses

Example of an invalid request response:

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

Example of a response for a successful batch release request:

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

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

Batch release query request

After uploading your batch release file, the system takes around 5 minutes to complete the release of the transactions, depending on file size. You can periodically poll the service using a batch release **POST** query request to query the status of a request.

To submit a batch release query request, **POST** parameters to the Remote Auth API:

Parameters	Description
`profile_id`	Your Profile Id
`profile_pass`	Authentication password
`batch_id`	The Id of the batch that you wish to query
`batch_op`	Type of Batch operation. For a batch release query request the value must be ‘onhold-release-query’

Example batch release query request

Example of the **POST** header sent in the batch release query request to the Remote Auth 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, the response will contain one of following results:

Query results	Description
`invalid_request`	Error – Request cannot be parsed correctly
`invalid_credentials`	Error – Cannot verify the profile id or authentication password
`internal_failure`	Error – There has been an internal error, please try again
`release_notfound`	Error – Cannot find the requested batch Id
`release_report`	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
`item_count`	Total number of items in the batch release
`item_succ`	Number of transactions that have been successfully released
`item_fail`	Number of transactions that have failed and not been released
`attachment_type`	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
`batchrel_notonhold`	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
`batchrel_invalref`	The transaction could not be found as it has an invalid reference
`batchrel_expired`	The transaction has expired and therefore has not been sent of authorisation
`batchrel_error`	There was an internal error, please resubmit this transaction release request
`batchrel_ok`	The transaction has been successfully released for authorisation

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 were requested to be released:

```
--remote_batch-4C4106B0
Content-Type: text/csv 01S00001724,batchrel_expired 01S00001725,batchrel_ok
01S00001726,batchrel_ok
--remote_batch-4C4106B0-
```

Recurring/continuous payments

You can submit a recurring payment using an approach called continuous authority. When sending a continuous (recurring) payment request you must include an Account Verification ID to enable us to use the initially stored card details.

Account Verification ID

To submit a recurring payment, you need to first get an initial account verification of a customer's card details, including the CVV. The request is then checked, and if successful, authorised. The card details are then securely held in our PCI approved systems and a request response sent to you with an Account Verification ID.

Account verification request parameters

To submit an account verification request you need to send a request with the mandatory parameters. There are also additional, optional parameters you can include, see the API Reference for more information.

The mandatory parameters are:

Parameter	Description
`auth_id`	Must be set to the Profile ID
`auth_pass`	Authentication password
`card_num`	Customer's card number (Must be numeric only with no separators) (**Conditional**, not required where `card_token` is provided)
`card_token`	Customer's card token (Max of 50 characters) (**Conditional**, not required where `card_num` is provided)
`return_token`	If not Null, the `card_token` will be included in the response only when we have processed a live payment
`card_cvv`	Card security code
`card_expiry`	Card expiry date, format is MMYY
`tran_ref`	Your transaction reference (e.g., cart ID)
`tran_currency`	Transaction currency, 3-character code (For a list of currencies code you can use / accept, please contact support@cashflows.com)
`tran_testmode`	Transaction test mode. 0
`tran_type`	Transaction type = verify
`tran_class`	Transaction class = ecom
`retry_number`	Indication of the number of retries attempts, 0 = initial attempt (See Retry Handling)
`return_acq_ref`	If not Null, the Acquirer Reference Number (ARN) will be included in the response only when we have processed a live payment

If you're using 3D Secure with Visa, Mastercard, or American Express, you must also include:

Parameters (3D Secure)	Description
`acs_eci`	The response from the 3DS server: • **5** = VbyV **-** Full Authentication • **6** = VbyV **-** Attempted Authentication • **7** = VbyV - No Authentication • **2** = MasterCard SecureCode **-** Full Authentication • **1** = MasterCard SecureCode **-** Attempted Authentication • **0** = MasterCard SecureCode **-** No Authentication • **05** = American Express Safekey - Full Authentication • **06** = American Express Safekey - Attempted Authentication • **07** = American Express Safekey - No Authentication
`acs_cavv`	The Cardholder Authentication Verification Value from 3DS server, 28 Characters **American Express Safekey** – provide the `American Express Verification Value` (AEVV) – 20 characters long.
`acs_dstransid`	The universally unique transaction identifier assigned by the Directory Server (DS) to identify a single transaction, 36 characters. Required when acs_3dsversion = 2.1.0/2.2.0. **American Express Safekey** – provide the `American Express Safekey Transaction ID` (XID) – 20 characters long.

If your MCC is 6012, 6051, or 7299 (financial institutions) you must also include:

Parameters (financial institutions)	Description
`primary_recipient_dob`	Customer's Date of Birth. Format is YYYYMMDD (8 numeric characters)
`primary_recipient_surname`	Customer's Surname or Last name (2-64 characters alpha characters, including –)
`primary_recipient_postcode`	Customer's Postcode (2 to 16-characters alpha characters, including spaces)
`primary_recipient_account_number`	Customer's Account Number (1 to 32 alpha numeric characters, including /-) For PAN Numbers: First 6 and Last 4. (Never include full PAN in primary_recipient_account_number field.)

Example 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 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=1000000000030419&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 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.

Setting up a recurring/continuous payment

When sending a continuous (recurring) payment request you must always include the Account Verification ID to enable us to use the initially stored card details to send the payment for authorisation.

Recurring/continuous payment request parameters

To send a recurring payment request you must **exclude** the `card_num` (or `card_token`), `card_expiry`, and `card_cvv` parameters, 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. There are also additional, optional parameters you can include, see the API Reference for more information.

The mandatory parameters are:

Parameter	Description
`auth_id`	Must be set to the Profile ID
`auth_pass`	Authentication password
`tran_ref`	Your transaction reference (e.g. cart ID)
`tran_amount`	Transaction amount to 2 decimal places, e.g. 24.99 (The currency symbol must not be included)
`tran_currency`	Transaction currency, 3-character code
`tran_testmode`	Transaction test mode. 0

tran_type	Transaction type = sale
tran_class	Transaction class = cont
tran_orig_id	Verification ID or Sales ID (e.g. 05P00001724 or 06S00001724)
scheme_transaction_id	ID provided by Acquirers and if the original transaction was processed via another Acquirer, they will need to obtain the id from them **Note** – Mandatory if tran_orig_id is **not** provided, if using an Acquirer other than Cashflows you can obtain the ID from them
retry_number	Indication of the number of retries attempts, 0 = initial attempt (For more information see Retry Handing)
return_acq_ref	If not Null the Acquirer Reference Number (ARN) will be included in the response only when we have processed a live payment

If your MCC is 6012, 6051, or 7299 (financial institutions) you must also include:

Parameters (financial institutions)	Description
primary_recipient_dob	Customer's Date of Birth. Format is YYYYMMDD (8 numeric characters)
primary_recipient_surname	Customer's Surname or Last name (2-64 characters alpha characters, characters, including -)
primary_recipient_postcode	Customer's Postcode (2 to 16-characters alpha characters, including spaces)
primary_recipient_account_number	Customer's Account Number (1 to 32 alpha numeric characters, Including

Example continuous payment request

Example of the **POST** string sent in the continuous payment request to the Remote Auth API for authorisation:

```
auth_id=1234&auth_pass=Password&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=cont&tran_orig_id=05P00001724
```

Authorisation response code

The response consists of:

- Authorisation status code
- Transaction ID
- CVV/AVS result
- Authorisation code
- Authorisation message
- Acquirer Reference 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
`A|01S00001724|232|031971|Authorised|74698692333601146452212|1000000000030419`	Authorised: A Transaction Id: 01S00001724 CVV/AVS:232 Authorisation code: 031971 Authorisation Request Response: 74698... Card token: 1000000000030419
`D|01S00001723|400|D102|Not Authorised|74698692333601146452212`	Not authorised: D Transaction Id: 01S00001723 CVV/AVS:400 Authorisation Request Response: 74698...
`V|99E5D84B40F|000|V226|Invalid  request`	Invalid request: V Transaction Id: 99E5D84B40F CVV/AVS:000 See API reference for more information
`B|01S00632BE2|000|D090|Not authorised`	Blocked: D Transaction Id: 01S00632BE2 CVV/AVS:000 See API reference for more information

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 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, timing, or connection issues our system will return a response code informing you of the issue.

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

If the system was unable to send the request of authorisation you will be returned the following response codes:

- S001 or S101: Connection failure

In this case you can resubmit authorisation request without risk of double authorisation.

If the system cannot determine whether an attempted authorisation was successful or not, the system will return the following response codes:

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

For the full list of response codes, see API reference.

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 these results:

- If the retry request is a duplicate request of a finished transaction, then the original transaction response is returned.
- If original transaction is still being processed, you will receive the `V249` response code.
- If our system has no record of a previous duplicate transaction request, then the transaction is processed, and the results returned.

Test your integration

You can test your Remote Auth integration by setting your **POST** request to the Integration environment (https://secure-int.cashflows.com/gateway/remote_auth) and using test cards details:

Card Number	Token	Expiry Date	CVV
`4000000000000002 (VISA Credit)`	1000000000030419	Any valid expiry date *(mm/yy)*	123
`4462030000000000 (VISA prepaid)`	1000000000030554	Any valid expiry date *(mm/yy)*	444
`5555555555554444 (MasterCard Credit)`	1000000000030567	Any valid expiry date *(mm/yy)*	321
`5597507644910558 (MasterCard prepaid)`	1000000000030568	Any valid expiry date *(mm/yy)*	888
`340001916255521 (American Express)`	1000000000030565	Any valid expiry date *(mm/yy)*	1234

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

API Reference

To make a request to the Remote Auth API you need to submit a UTF-8 encoded HTTPS **POST** to:

- Test - https://secure-int.cashflows.com/gateway/remote_auth.
- Live - https://secure.cashflows.com/gateway/remote_auth.

Payment request parameters

Parameters for making payments:

Parameter	Description
`auth_id`	Your Profile Id
`auth_pass`	Authentication password
`card_num`	Customer's card number (Must be numeric only with no separators) (**Conditional**, not required where card_token is provided)
`card_token`	Customer's card token (Max of 50 characters) (**Conditional**, not required where card_num is provided)
`card_cvv`	Card security code
`card_start`	Card start date, format is MMYY (Optional)
`card_issue`	Card issue number (Optional)
`card_expiry`	Card expiry date, format is MMYY
`cust_name`	Customer's name (Optional)
`cust_address`	Customer's address (Multiple lines can be separated using the new line break character (ASCII code 10)) (Optional)
`cust_postcode`	Customer's post/zip/area code (Optional)
`cust_country`	Customer's country *(ISO3166 2-character code)* (Optional)
`cust_ip`	Customer's IP address *(IPV4 Format only)* (Optional)
`cust_email`	Customer's email address (Optional)
`cust_tel`	Customer's telephone number (Optional)
`ewallet`	Indicates whether a Pay's wallet was used: true | false (*Optional*)
`ewallet_type`	Indicates which Pay's wallet was used (*Optional*) Accepted values: (case_sensitive): • **applepay** • **googlepay** • **samsungpay** • **other**
`tran_ref`	Your transaction reference *(e.g. cart ID)*
`tran_desc`	Your transaction description (Max of 99 characters) (Optional)
`tran_amount`	Transaction amount to 2 decimal places, e.g. 24.99 (The currency symbol must not be included)
`tran_currency`	Transaction currency (3-character code)
`tran_testmode`	Transaction test mode = 0
`tran_type`	Transaction type = sale

`tran_class`	Transaction class = ecom or moto
`tran_recurrence`	To be used to override default MID settings: • **sing** = Single transaction with no recurrence • **subs** = Transaction in recurring subscription • **inst** = Transaction in recurring instalment • **unsc** = Unscheduled transaction with a stored card • **card** = Cardholder initiated transaction with a stored card (Optional)
`retry_number`	Indication of the number of retries attempts, 0 = initial attempt (For more information see Retry Handing)
`return_acq_ref`	If not Null, the Acquirer Reference Number (ARN) will be included in the response only when we have processed a live payment. (Optional)
`return_issuer_response_code`	If not Null, the raw issuer response code will be included in the response when we have processed a live payment. (Optional)
`descriptor`	A soft descriptor that is added to your Company Name when displayed on the Cardholders statement (Max of 12 characters) (Optional)
`return_token`	If not Null the `card_token` will be included in the response only when we have processed a successful transaction
`sca_exemption_indicator`	Indication of why the transaction may be exempt from SCA, possible values: • **lowvalue** = Applicable to transactions where amount is less than €30 or currency equivalent) (Optional)

If you're using 3D Secure with Visa, Mastercard, or American Express, you must also include:

Parameters (3D Secure)	Description
`acs_eci`	The response from the 3DS server. • **5** = VbyV **-** Full Authentication • **6** = VbyV **-** Attempted Authentication • **7** = VbyV - No Authentication • **2** = MasterCard SecureCode **-** Full Authentication • **1** = MasterCard SecureCode **-** Attempted Authentication • **0** = MasterCard SecureCode **-** No Authentication • **05** = American Express Safekey - Full Authentication • **06** = American Express Safekey - Attempted Authentication • **07** = American Express Safekey - No Authentication
`acs_cavv`	The Cardholder Authentication Verification Value from 3DS server, 28 characters **American Express Safekey** – provide the `American Express Verification Value` (AEVV) – 20 characters long.
`acs_dstransid`	The universally unique transaction identifier assigned by the Directory Server (DS) to identify a single transaction, 36 characters. Required when acs_3dsversion = 2.1.0/2.2.0. **American Express Safekey** – provide the `American Express Safekey Transaction ID` (XID) – 20 characters long.

If your MCC is 6012, 6051, or 7299 (financial institutions) you must also include:

Parameters (financial institutions)	Description
`primary_recipient_dob`	Customer's Date of Birth. Format is YYYYMMDD (8 numeric characters)
`primary_recipient_surname`	Customer's Surname or Last name (2-64-characters alpha characters, including -)
`primary_recipient_postcode`	Customer's Postcode (2 to 16-characters alpha characters, including spaces)
`primary_recipient_account_number`	Customer's Account Number (1 to 32 alpha numeric characters, including /-) For PAN Numbers: First 6 and Last 4

Void and refund request parameters

Parameters for voiding and refunding payments:

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

Credit transfer request parameters

Parameters for requesting credit transfers:

Parameters	Description
`auth_id`	Your Profile Id
`auth_pass`	Authentication password
`tran_amount`	Credit Transfer amount to 2 decimal places, e.g. 24.99 (The currency symbol must not be included)
`tran_currency`	Transaction currency, 3-character code, e.g. GBP
`tran_ref`	Your transaction reference *(e.g. cart ID)*
`tran_testmode`	Transaction test mode. 0
`tran_type`	Transaction type = credit

`tran_class`	Transaction class = cred
`descriptor`	Mastercard only. A descriptor that is added to your Company Name when displayed on the Cardholders statement (Max of 12 characters) (Optional)
`tran_orig_id`	*Mandatory for Mastercard.* *Optional for Visa (where card_num or card_token is provided)* Original transaction Id to which the credit will be applied to.
`card_num`	Visa cards only: Customer’s card number (*Must be numeric only with no separators)* *(Optional, not required where tran_orig_id* or *card_token is provided)*
`card_token`	Visa cards only: Customer’s card token (Max of 50 characters) (Optional*, not required where card_ num or tran_orig_id is provided*)
`security_hash`	A security Hash value used to ensure that no-one has tampered with the credit transfer request

Batch release parameters

Parameters for requesting batch release:

Parameters	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’

Batch release response

Response to batch release request:

Parameters	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

Batch release query request

Parameters for querying batch release requests:

Parameters	Description
`profile_id`	Your Profile Id
`profile_pass`	Authentication password

`batch_id`	The Id of the batch that you wish to query
`batch_op`	Type of Batch operation. For a batch release query request the value must be ‘onhold-release-query’

Batch release query response

Response to batch release query request:

Query results	Description
`invalid_request`	Error – Request cannot be parsed correctly
`invalid_credentials`	Error – Cannot verify the profile id or authentication password
`internal_failure`	Error – There has been an internal error, please try again
`release_notfound`	Error – Cannot find the requested batch Id
`release_report`	Success – The batch query request has been successfully submitted and a batch has been found

If the query was successfully submitted, 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
`item_count`	Total number of items in the batch release
`item_succ`	Number of transactions that have been successfully released
`item_fail`	Number of transactions that have failed and not been released
`attachment_type`	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
`batchrel_notonhold`	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
`batchrel_invalref`	The transaction could not be found as it has an invalid reference
`batchrel_expired`	The transaction has expired and therefore has not been sent of authorisation
`batchrel_error`	There was an internal error, please resubmit this transaction release request
`batchrel_ok`	The transaction has been successfully released for authorisation

Recurring/Continuous payments

Account verification request parameters

Parameters for account verification requests:

Parameter	Description
`auth_id`	Must be set to the Profile ID
`auth_pass`	Authentication password
`card_num`	Customer's card number (Must be numeric only with no separators) (**Conditional**, not required where `card_token` is provided)
`card_token`	Customer's card token (Max of 50 characters) (**Conditional**, not required where `card_num` is provided)
`return_token`	If not Null the `card_token` will be included in the response only when we have processed a live payment
`card_cvv`	Card security code
`card_start`	Card start date, format is MMYY (Optional)
`card_issue`	Card issue number (Optional)
`card_expiry`	Card expiry date, format is MMYY
`cust_name`	Customer's name (Optional)
`cust_address`	Customer's address (Multiple lines can be separated using the new line break character (ASCII code 10)) (Optional)
`cust_postcode`	Customer's post/zip/area code (Optional)
`cust_country`	Customer's country, ISO3166 2-character code (Optional)
`cust_ip`	Customer's IP address (Optional) (IPV4 Format only)
`cust_email`	Customer's email address (Optional)
`cust_tel`	Customer's telephone number (Optional)
`ewallet`	Indicates whether a Pay's wallet was used: true | false (Optional)
`ewallet_type`	Indicates which Pay's wallet was used (Optional) Accepted values (case-sensitive): applepay googlepay samsungpay other
`tran_ref`	Your transaction reference (e.g. cart ID)
`tran_desc`	Your transaction description (Optional)
`tran_currency`	Transaction currency, 3-character code (For a list of currencies code you can use / accept, please contact support@cashflows.com)

`tran_testmode`	Transaction test mode. 0
`tran_type`	Transaction type = verify
`tran_class`	Transaction class = ecom
`retry_number`	Indication of the number of retries attempts, 0 = initial attempt (See Retry Handling)
`return_acq_ref`	If not Null the Acquirer Reference Number (ARN) will be included in the response only when we have processed a live payment
`descriptor`	A soft descriptor that is added to your Company Name when displayed on the Cardholders statement (Max of 12 characters) (Optional)
`tran_recurrence`	To be used to override default MID settings (optional): • **sing** = Single transaction with no recurrence • **subs** = Transaction in recurring subscription • **inst** = Transaction in recurring instalment • **unsc** = Unscheduled transaction with a stored card • **card** = Cardholder initiated transaction with a stored card

If you're using 3D Secure with Visa, Mastercard, or American Express, you must also include:

Parameters (3D Secure)	**Description**
`acs_eci`	The response from the 3DS server: • **5** = VbyV - Full Authentication • **6** = VbyV - Attempted Authentication • **7** = VbyV - No Authentication • **2** = MasterCard SecureCode - Full Authentication • **1** = MasterCard SecureCode - Attempted Authentication • **0** = MasterCard SecureCode - No Authentication • **05** = American Express Safekey - Full Authentication • **06** = American Express Safekey - Attempted Authentication • **07** = American Express Safekey - No Authentication
`acs_cavv`	The Cardholder Authentication Verification Value from 3DS server, 28 characters. **American Express Safekey** – provide the `American Express Verification Value` (AEVV) – 20 characters long.
`acs_dstransid`	The universally unique transaction identifier assigned by the Directory Server (DS) to identify a single transaction, 36 characters. Required when acs_3dsversion = 2.1.0/2.2.0. **American Express Safekey** – provide the `American Express Safekey Transaction ID` (XID) – 20 characters long.

If your MCC is 6012, 6051, or 7299 (financial institutions) you must also include:

Parameters (financial institutions)	Description
`primary_recipient_dob`	Customer's Date of Birth. Format is YYYYMMDD (8 numeric characters)
`primary_recipient_surname`	Customer's Surname or Last name (2-64 characters alpha characters, including –)
`primary_recipient_postcode`	Customer's Postcode (2 to 16-characters alpha characters, including spaces)
`primary_recipient_account_number`	Customer's Account Number (1 to 32 alpha numeric characters, including /-) For PAN Numbers: First 6 and Last 4. (Never include full PAN in primary_recipient_account_number field.)

Recurring/continuous payment request parameters

Parameters for making recurring/continuous payment requests:

Parameter	Description
`auth_id`	`Must be set to the Profile ID`
`auth_pass`	`Authentication password`
`cust_name`	Customer's name (Optional)
`cust_address`	Customer's address *(Multiple lines can be separated using the new line break character (ASCII code 10))* (Optional)
`cust_postcode`	Customer's post/zip/area code (Optional)
`cust_country`	Customer's country, ISO3166 2-character code (Optional)
`cust_ip`	Customer's IP address *(IPV4 Format only)* (Optional)
`cust_email`	Customer's email address (Optional)
`cust_tel`	Customer's telephone number (Optional)
`tran_ref`	Your transaction reference (e.g. cart ID)
`tran_desc`	Your transaction description (Optional)
`tran_amount`	Transaction amount to 2 decimal places, e.g. 24.99 (*The currency symbol must not be included)*
`tran_currency`	Transaction currency, 3-character code
`tran_testmode`	Transaction test mode. 0
`tran_type`	Transaction type = sale
`tran_class`	Transaction class = cont

`tran_orig_id`	Verification ID or Sales ID (e.g. *05P00001724 or 06S00001724*)
`retry_number`	Indication of the number of retries attempts, 0 = initial attempt *(For more information refer to Retry Handing)*
`return_acq_ref`	If not Null the Acquirer Reference Number *(ARN)* will be included in the response only when we have processed a live payment
`descriptor`	A soft descriptor that is added to your Company Name when displayed on the Cardholders statement. (Max of 12 characters) *(Optional)*
`tran_recurrence`	To be used to override default MID settings: • **sing** = Single transaction with no recurrence • **subs** = Transaction in recurring subscription • **inst** = Transaction in recurring instalment • **unsc** = Unscheduled transaction with a stored card • **card** = Cardholder initiated transaction with a stored (Optional)
`sca_exemption_indicator`	Indication of why the transaction may be exempt from SCA, possible values: • **recurring =** Applicable to transactions where recurrence type = ‘**subs**’ or ‘**inst**’ • **merchantinitiated** = Applicable to merchant-initiated transactions where recurrence type = ‘**sing’** or ‘**unsc’** (Optional)

If your MCC is 6012, 6051, or 7299 (financial institutions) you must also include:

Parameters (financial institutions)	Description
`primary_recipient_dob`	Customer’s Date of Birth. Format is YYYYMMDD (8 numeric characters)
`primary_recipient_surname`	Customer’s Surname or Last name (2-64 characters alpha characters, characters, including -)
`primary_recipient_postcode`	Customer’s Postcode (2 to 16-characters alpha characters, including spaces)
`primary_recipient_account_number`	Customer’s Account Number (1 to 32 alpha numeric characters, including

Acquirer system response codes

Responses consist of a letter followed by a 3-digit code:

- Letter - indicates the type of status
  - **A** - authorised
  - **V** - validation error (e.g. invalid card number)
  - **D** - declined
  - **R** - referral (treated as declined)
  - **B** - blocked
  - **C** - cancelled (e.g. user pressed cancel on payment page)
  - **S** - system error
- First number – internal code that can be ignored
- Last 2 numbers – specific error code.

Below is a list of current error codes (this list is subject to change):

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
Vx34	Unable to void transaction
Vx35	Original trans was not authorised
Vx36	Invalid PIN
Vx37	Unknown transaction class
Vx38	Original transaction type does not match
Vx39	Card expired
Vx40	CVV Required
Vx41	Original transaction already settled
Vx42	Original transaction already cancelled
Vx43	This card does not support the required transaction type
Vx44	Transaction details do not match original
Vx48	User Details not valid
Vx52	3DS Not Enabled
Vx53	3DS Data Invalid
Vx54	Concurrent Authorisations
Vx55	Invalid Funds Recipient Date (MCC 6012, 6051 or 7299 merchants)
Vx56	Terminal mismatch
Vx57	Transaction not allowed on this card
Vx58	Original transaction requires 3DS attempt/auth
Vx59	ECOM transactions require 3DS attempt/auth
Vx60	Verify for Amex card not supported
Vx61	Recurrence Flag usage invalid
Vx62	Initial Sale/Verify ARN missing for subsequent sale
Vx63	Initial Sale/Verify for subsequent sale not approved
Vx64	Initial transaction on card expired
Dx01	Non-specific decline
Dx02	Declined due to funds (insufficient/limit exceeded)

Dx03	Retain card response
Dx05	On our blacklist
Dx07	Live/test mismatch
Dx08	Refund: Insufficient merchant funds in account
Dx10	Card authorisation attempt limit reached
Dx11	Monthly Scheme Decline Rate limit reached
Dx40	Continuous Authority cancelled for the transaction
Dx41	Continuous Authorities cancelled for the merchant
Dx43	Continuous Authorities cancelled for the card
Dx49	Additional customer authentication required
Dx90	Pre-Authorisation anti-fraud block
Dx91	Post-Authorisation anti-fraud block
Rx01	Not Authorised
Ex01	Transaction error
Cx01	Transaction cancelled
Cx02	Transaction expired
Sx00	Invalid transaction Request
Sx01	Connection failure
Sx02	Invalid response
Sx03	Response timeout
Sx04	Server error
Sx05	Server error
Sx06	No response from issuer
Sx07	Service not available
Sx99	Unknown Error

Cashflows

+44 (0)1223 550920

London
20 Farringdon St
London
EC4A 4AB

The Netherlands
Noorderhof 24
5804 BV Venray