Cashflows

Remote Auth API Integration Guide

Version 3.1 – October 2020

Table of contents

About this guide

Welcome to the Cashflows Remote Auth API Integration Guide. This document is designed to provide gateway providers with 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

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 limited. Cashflows Europe limited, 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 limited. Cashflows Europe limited 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 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 Remote Auth API

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.

Security Requirements

Using the Remote Auth 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 Remote Auth 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/qsa_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:

URL	Environment
https://secure-int.cashflows.com/gateway/remote_auth	Integration
https://secure.cashflows.com/gateway/remote_auth	Production

Please contact 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 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 Remote Auth API to request a payment authorisation.

Note: All payment request parameters are mandatory unless specified.

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 refer to Retry Handing)
return_acq_ref	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. (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)

Additional fields for 3d Secure Transactions	
Version 1:	
acs_3dsversion	The version of 3DS being used for authentication (Optional, when not supplied version 1.0.2 is assumed)
acs_eci	The response from the Access Control Server, stating the 3DS method and result: • **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
acs_cavv	The Cardholder Authentication Verification Value from the Access Control Server, 28 characters
acs_xid	The unique Authentication Id for transaction from the Access Control Server, 28 characters
Version 2:	
acs_3dsversion	The version of 3DS being used for authentication (Optional - when not supplied version 1 is assumed) otherwise 2.1.0 or 2.2.0 should be set
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
acs_cavv	The Cardholder Authentication Verification Value from 3DS server, 28 characters
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.
Additional fields for merchants with the MCC 6012, 6051 or 7299. (Financial Institutions)	
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 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_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=5dbc4a6a39b6730a360e42c3b
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 **POST** request will send the following payment details to the Remote Auth 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_cou
ntry=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=5dbc4a6a39b6730a360e42c3b5
f4&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=1000000000030419&card_cvv=123&card_expiry=0121&cust_
name=Testing&cust_address=My%20house%0AMy%20street%0AMy%20Town&cust_postcode=CB22%205LD&c
ust_country=GB&cust_ip=123.45.67.89&cust_email=test@test.com&tran_ref=abc123&tran_amount=9.99&tran_currenc
y=GBP&tran_testmode=0&tran_type=sale&tran_class=ecom&acs_eci=5&acs_cavv=5dbc4a6a39b6730a360e42c3
b5f4&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 Remote Auth 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_co
untry=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=5dbc4a6a39b6730a360e42c3b
5f4&acs_xid=ef181 c0031b5da142e2e8c49424
```

Supporting Apple Pay, Samsung Pay and Google Pay (Digital Wallet providers)

The Cashflows Remote Auth API has added support for Apple Pay, Google Pay and Samsung Pay, the leading wallet providers. We've increased our support for Cardholder Not Present transactions by allowing the wallet type used by you and your gateway providers to be included within authorisation and continuous payment transactions. A new field called ewallet_type, when used in conjunction with the ewallet field, indicates to our system that a wallet has been used.

With these fields, you can increase your understanding of which payment methods are being used within your business, to identify these transactions help your decisions. You can see this increased detail of payment types within your extracted reports and within the detail page of each transaction.

Action

For Cashflows Remote Auth API to correctly categorise a Pays (ewallet) transaction (beyond just Card), we use two fields to set whether a wallet was used in the transactions. The details are outlined here:

- When both the ewallet and ewallet_type fields are not present (null), it will yield a Card transaction
- The ewallet field indicates that a wallet transaction is being done
  - When set to **true**,
    - When ewallet_type field is present, it will use information presented in that field to distinguish the wallet used in the transaction
    - When ewallet_type is not presented, it yields as **other** wallet transaction
    - 4 additional field are needed with the transaction to communicate the details of the wallet information forward to the schemes

Field	Usage
card_num	Token (DPAN)
card_expiry	Token expiration date (MMYY)
acs_cavv	Cryptogram
acs_eci	Status

  - When set to **false** or not presented (null value), it yields a Card transaction
- The second field, ewallet_type will distinguish the type of wallet used; it requires the ewallet field to be set to **true**

  Expected values are

  - "applepay"
  - "googlepay"
  - "samsungpay"
  - "other"

- No other value will be accepted; if another value is presented is will be considered an invalid request
- The presence of the ewallet_type, requires first field ewallet to be present and **true**

Results

Both fields are optional, so if neither are presented, the transaction will be categorised as a card transaction.

Remote administration

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

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 Remote Auth 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 Remote Auth API:

PARAMETER	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 of a 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 of a 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=2.99&tran_currency=GBP&tran_testmode=0&tran_type=refund&tr an_class=ecom&tran_orig_id=01S0001

Credit Transfers

If you 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

Credit Transfer Request parameters

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

PARAMETER	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

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]
For Visa card Credit Transfers, where tran_orig_id not supplied in the request, this parameter must be omitted from the hash string, as indicated below:

tran_type:tran_amount:tran_currency::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: Please contact your account manager to confirm if Credit Transfers have been enabled on your account.

Credit transactions are only supported for the following MCC's:
7995 - Gambling
7994 - Game of skill
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
8999 - Professional Services (Not Elsewhere Classified)
5262 - Marketplaces

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 Remote Auth API. The **POST** request must be UTF-8 encoded and submitted to:

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 Remote Auth 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 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:

```
--_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 Remote Auth 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 Remote Auth API:

PARAMETER	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 of a 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 request to the Remote Auth API, 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 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 Remote Auth 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
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 *(For more information refer to Retry Handing)*
return_acq_ref	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
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

Additional fields for 3DSecure Transactions	
Please note that for transactions that are wallet-initiated, these field will carry details of from the wallet itself; so these field will be used for wallet transactions	
Version 1:	
acs_3dsversion	The version of 3DS being used for authentication (Optional, when not supplied version 1.0.2 is assumed)
acs_eci	The response from the Access Control 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
acs_cavv	The Cardholder Authentication Verification Value from the Access Control Server, 28 characters
acs_xid	The unique Authentication Id for transaction from the Access Control Server, 28 characters
Version 2:	
acs_3dsversion	The version of 3DS being used for authentication (Optional - when not supplied version 1 is assumed) otherwise 2.1.0 or 2.2.0 should be set
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
acs_cavv	The Cardholder Authentication Verification Value from 3DS server, 28 characters
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.
Additional fields for merchants with the MCC 6012, 6051 or 7299. (Financial Institutions)	
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 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=T
esting&cust_address=My%20house%0AMy%20street%0AMy%20Town&cust_postcode=CB22%205LD&c
ust_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=1000000000030419&card_cvv=123&card_expiry=0121&cus
t_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 Payments

When sending a continuous (recurring) payment request you must always include the account verification ID (see above in Recurring Payments section) to enable us to use the initially stored card details to send the payment for authorisation.

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
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's Authorisation *Request Response 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)
Additional fields for merchants with the MCC 6012, 6051 or 7299. (Financial Institutions)	
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 of a 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 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
*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* *(please refer to Appendix A for further information.)*
B|01S00632BE2|000|D090|Not authorised	*Blocked: D* *Transaction Id: 01S00632BE2* *CVV/AVS:000* *(please refer to Appendix A for further 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 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 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 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 Remote Auth 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 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.

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

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

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
26/09/2019		2.5
September 2020	eWallet and eWallet type added	3.0
October 2020	Added - 3DS v2 Parameters - sca_exemption_indicator parameters for Sales and Continuous Authority transactions requests	3.1