

Rapid Pay Collect

API Specification

Debit Order Collections and Payments Technical Document

Last updated: 15 April 2021

Contents

Introduction	5
Getting started	5
Response formats and structure.....	5
Successful request.....	5
Unsuccessful request	6
Duplicate checking	6
Using Tokens in place of account details	6
External links	7
Example.....	8
Payments.....	9
Overview	9
Request URL	9
Request structure.....	9
Request Example.....	11
Response structure	11
Response Example	12
Complete callback	12
Complete callback example	13
Unpays.....	13
Unpays example.....	14
Batch Bank Account Verification (BANV)	15
Overview	15
Request URL	15
Request structure.....	15
Request Example.....	17
Response structure	17
Response Example	17
Validation Responses	18
Validations Response Example	19
Real-time Bank Account Verification (BANVR)	20
Overview	20
Request URL	20

Request structure.....	20
Request Example.....	22
Response structure	22
Response Example	24
Check Digit Verification (CDV).....	25
Overview	25
Request URL	25
Request structure.....	25
Request Example.....	26
Response structure	26
Response Example - Valid	26
Response Example - Invalid.....	27
Debit Orders.....	28
Overview	28
Request URL	28
Request structure.....	28
Request Example.....	29
Response structure	29
Response Example	30
Complete callback	30
Complete callback example	31
Unpays.....	31
Unpays Example.....	32
Field Lengths and Data Types	33
Overview	33
Batch BANV Test Accounts.....	34
Overview	34
Note.....	34
Test Accounts	34
Request Example	36
Response Example.....	36
Call-back Example	37
Real-time BANV Test Accounts	38

Overview	38
Note.....	38
Test Accounts	38
Response Example.....	40
CDV Test Accounts	41
Overview	41
Note.....	41
Test Accounts	41
Request Example.....	42
Response Example.....	42
Payments Test Accounts	43
Overview	43
Note.....	43
Test Accounts	43
Request Example.....	44
Response Example.....	44
Call-back Example	45

Introduction

The Rapid Pay Collect API gives developers access to the processing platform used by Rapid Pay Collect. Developers can use the API to securely submit requests for services using their own software.

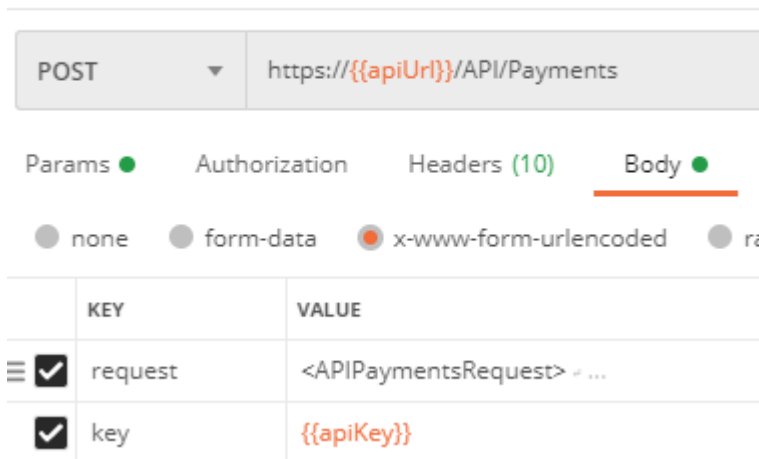
Getting started

Contact Rapid Pay Collect and request an API key. **Your API key will be used with every request in order to identify your application. If no key is provided, then access to the data will not be granted.** An incorrect key will result in the same error.

All request and response data are submitted as a form data in a POST action (i.e. application/x-www-form-urlencoded) with the key “response”. The API key will use the key “key”.

To assist in visualising the request, a screenshot of the values in Postman is displayed below.

► Payment (Creditors) example



To assist with integration, a Postman collection with the common requests and responses can be provided on request.

Response formats and structure

Successful request

The response is in a structured XML format and will always have the following basic structure:

```
<Response>
  <Result>OK</Result>
  <BatchCode>285906</BatchCode>
  <TotalFeeExcludingVAT>1.75</TotalFeeExcludingVAT>
</Response>
```

The *Result* element’s value is always “OK” if no errors were encountered. The *BatchCode* element’s value will display the batch code for that transaction. The batch code is a unique code that is returned with all responses so that you can determine which batch the response is for. The *TotalFeeExcludingVAT* element’s value will return the fees charged by Rapid Pay Collect for that batch. Other elements may be returned depending on the service.

Whenever a date is used it is always in the format yyyyMMdd e.g. 20120214 for 14 February 2012.

If a unique Id value has been included, it will be returned with the response as well.

Unsuccessful request

```
<Response>
  <Result>Error</Result>
  <ResultMessage>Your key is invalid</ResultMessage>
</Response>
```

If a request was an unsuccessful the *Result* element's value will always be "Error". The *ResultMessage* value will display the reason for the error.

Duplicate checking

All requests can include an optional *Uniquelid* field in the header of the request. When this field is included, it will be saved with the batch as a unique identifier. If another request arrives with the same unique Id value, it will be rejected. The original BatchCode will be included in the result.

```
<Response>
  <Result>Error</Result>
  <ResultMessage>This batch has the same unique Id as another batch and is rejected as
a duplicate</ResultMessage>
  <BatchCode>285907</BatchCode>
</Response>
```

Using Tokens in place of account details

Each beneficiary record includes an API token that can be used as a shortcut for filling in the account details for that transaction. If the token field is present in the *FileContents* section, all other fields except for **FileAmount** and **AmountMultiplier** will be ignored and can be excluded.

The token will be matched to a beneficiary record for the client code specified in the request and the relevant information will be filled in from the beneficiary record.

For security reason, CDV results will not include the **account number** or **branch code** in the API response but will include the **customer code** and **token** so that you can match it the original transaction.

```
<Response>
  <Result>OK</Result>
  <BatchCode>285909</BatchCode>
  <TotalFeeExcludingVAT>1.75</TotalFeeExcludingVAT>
  <CDVResults>
    <Result>
      <CustomerCode>EXA9292</CustomerCode>
      <Result>Valid</Result>
      <Token>377089C6C48CB38ACB4134F50D84004D3657CB03</Token>
    </Result>
  </CDVResults>
</Response>
```

A CDV error will be returned with the **token** value if no matching beneficiary record can be found.

```
<Response>
  <Result>OK</Result>
  <BatchCode>285926</BatchCode>
  <BatchValueSubmitted>549.01</BatchValueSubmitted>
  <CDVResults>
    <Result>
      <CustomerCode></CustomerCode>
      <Message>Token can not be matched to beneficiary. Ensure the beneficiary is
active and linked to this service.</Message>
      <Result>Invalid</Result>
      <Token>377089C6C48CB38ACB4134F50D84004D3657CB01</Token>
    </Result>
  </CDVResults>
</Response>
```

External links

External links are used to reference an object or page that is not part of the Rapid Pay Collect secure web interface. For example, a link to the original invoice for a payment can be included for display in reports. All requests can include an optional *ExternalLinks* section for each account (*FileContents*) field.

External links are displayed on the authorisation page of the secure web interface and is in the process of being rolled out to other reports.

Structure

An *ExternalLink* section consists of the following elements:

Element	Required	Description
Label	Yes	A description of the link. This will be displayed next to each link when reporting.
URL	Yes	The URL of the link

Multiple *ExternalLink* sections can be included under the *ExternalLinks* containing section.

Example

```
<APIPaymentsRequest>
  <Header>
    <PsVer>2.0.1</PsVer>
    <Client>ZER001</Client>
    <DueDate>20201125</DueDate>
    <Service>Wages</Service>
    <ServiceType>1day</ServiceType>
    <Reference>Example Batch</Reference>
  </Header>
  <Payments>
    <FileContents>
      <Initials>EX</Initials>
      <FirstNames>Example</FirstNames>
      <Surname>Recipient</Surname>
      <BranchCode>632009</BranchCode>
      <AccountNumber>123456789</AccountNumber>
      <FileAmount>549.01</FileAmount>
      <AccountType>0</AccountType>
      <AmountMultiplier>1</AmountMultiplier>
      <CustomerCode>EXA9292</CustomerCode>
      <Reference>Example Reference</Reference>
      <ExternalLinks>
        <ExternalLink>
          <Label>Invoice 12345</Label>
          <URL>https://example.com/View/12345</URL>
        </ExternalLink>
      </ExternalLinks>
    </FileContents>
  </Payments>
  <Totals>
    <Records>1</Records>
    <Amount>549.01</Amount>
    <BranchHash>632009</BranchHash>
    <AccountHash>123456789</AccountHash>
  </Totals>
</APIPaymentsRequest>
```


Payments

Overview

A payments request will submit data relating to a payments batch. A payments batch can be for Creditors, Salaries or Wages. Once a payment batch has been submitted, a CDV check will be performed on the account details provided and any accounts failing that check will be included in the response. Any accounts that passed the CDV check will continue to be processed.

If an account is rejected by a bank (usually because it has been closed) an unpaids response is created and POSTed back to the call-back URL provided. All response data are submitted as a form data in a POST action (i.e. application/x-www-form-urlencoded) with the key "response".

Request URL

<https://secure.rapidpaycollect.co.za/API/Payments>

Request structure

A payments request consists of 3 separate sections with a root element with the name *APIPaymentsRequest*. The *Header* section contains information relating the payments batch, the *Payments* sections is a collection of account details to be used as the destination for the payments and the *Totals* section is used to verify the information being sent is complete and correct.

Header

A payment request has a single *Header* section and it consists of the following elements:

Element	Required	Description
PsVer	Yes	The version of the Rapid Pay Collect file format. Currently 2.0.1.
Client	Yes	Your unique client code.
Service	Yes	The code for the service being used. See below for the service codes.
ServiceType	Yes	The service type of the payment. See below for the service types.
DueDate	Yes	The date the batch should be submitted to the bank for processing.
BankAccount	No	The bank account being used to deposit the funds. This is used to display the Rapid Pay Collect account details for your bank account.
CallBackUrl	Yes	The call-back URL to be used to send back unpaids.
Reference	Yes	The reference used to identify the batch for reporting
Uniqueld	No	Used to prevent duplicate requests

Payment service codes

The following service codes are available for payments.

Value	Description
Wages	For payments relating to wages.
Salaries	For payments relating to salaries.
Creditors	For payments relating to creditors.

Payment service types

The following service codes are available for payments.

Value	Description
1Day	For 1-day payments.
SDV	For SDV payments.
RTC	For real-time payments

Payments

A payment request has a single *Payments* section with multiple *FileContents* sections consisting of the following elements:

Element	Required	Description
Initials	No	The initials of the payment recipient.
FirstNames	Yes	The first name(s) of the payment recipient.
Surname	Yes	The surname of the payment recipient.
BranchCode	Yes	The branch code of the account where the payment will be deposited.
AccountNumber	Yes	The account number of the account where the payment will be deposited.
FileAmount	Yes	The amount to be deposited.
AmountMultiplier	Yes	Usually 1 if the FileAmount is specified in Rands otherwise 100 if the FileAmount is specified in cents.
AccountType	No	The type of account where the funds will be deposited. This can be left blank or 0 can be used if you don't know. Only 0 – 6 can be used.
CustomerCode	No	The customer code used to identify the payment in your system.
Reference	Yes	The reference that will appear on the recipient's bank statement.

Totals

A payment request has a single *Totals* section and it consists of the following elements:

Element	Required	Description
Records	Yes	The total number of payment records being submitted.
Amount	Yes	The total value of the payment records being submitted.
BranchHash	Yes	The sum of all the branch codes for the payment records.
AccountHash	Yes	The sum of all the account numbers for the payment records.

Request Example

```

<APIPaymentsRequest>
  <Header>
    <PsVer>2.0.1</PsVer>
    <Client>ZER001</Client>
    <DueDate>20200625</DueDate>
    <Service>Creditors</Service>
    <ServiceType>1day</ServiceType>
    <Reference>Example Batch</Reference>
    <CallBackUrl>https://example.com/Callback</CallBackUrl>
  </Header>
  <Payments>
    <FileContents>
      <Initials>EX</Initials>
      <FirstNames>Example</FirstNames>
      <Surname>Recipient</Surname>
      <BranchCode>632009</BranchCode>
      <AccountNumber>123456789</AccountNumber>
      <FileAmount>549.01</FileAmount>
      <AccountType>0</AccountType>
      <AmountMultiplier>1</AmountMultiplier>
      <CustomerCode>EXA9292</CustomerCode>
      <Reference>Example Reference</Reference>
    </FileContents>
  </Payments>
  <Totals>
    <Records>1</Records>
    <Amount>549.01</Amount>
    <BranchHash>632009</BranchHash>
    <AccountHash>123456789</AccountHash>
  </Totals>
</APIPaymentsRequest>

```

Response structure

The standard response structure will be returned with an extra element named `BatchValueSubmitted` which contains the total value of the batch submitted (less any accounts that failed the CDV check).

If any of the payment's `FileContents` records failed the CDV check, they will be returned as a `CDVResults` element with a collection of `Result` elements outlining the reasons for the accounts rejection. The `Result` section consists of the following elements:

Filter	Description
Result	The result of the CDV check. Will always be Invalid for a failed account.
Message	A message giving the reason for the rejection.
AccountNumber	The account number that failed the CDV test.

Please note: If any of the accounts passed the CDV check the batch will be accepted and will proceed to be processed.

Response Example

```

<Response>
  <Result>OK</Result>
  <BatchCode>285902</BatchCode>
  <BatchValueSubmitted>549.01</BatchValueSubmitted>
  <TotalFeeExcludingVAT>2.00</TotalFeeExcludingVAT>
  <CDVResults>
    <Result>
      <AccountNumber>123456789</AccountNumber>
      <BranchCode>632009</BranchCode>
      <CustomerCode>EXA9292</CustomerCode>
      <Reference>Example Reference</Reference>
      <Result>Valid</Result>
    </Result>
  </CDVResults>
</Response>

```

Complete Callback

Due to the manner in which the banks operate, a payment is assumed to be successful unless an unpaids response is returned. However, it remains useful to subscribe to a callback once a payment has been accepted by the bank in order to trigger a workflow based on the success of the payment. **A complete response is POSTed back to your server using the *CallbackUrl* provided in the initial request.**

Unlike unpaids callbacks, complete callbacks need to be enabled on your client profile in order to be sent. Please contact help@rapidpaycollect.co.za to request activation of the complete callback.

Note: The complete responses are sent once the bank has accepted the transactions and could be the day before the payment is received by the recipient.

The *Response* root element will consist of the following elements:

Filter	Description
Result	Will always be OK.
BatchCode	The unique code for the batch.
PaymentResults	The collection of <i>Results</i> elements with more information about each completed transaction.

The *PaymentResult* section consists of the following elements:

Element	Description
FirstName	The first name(s) of the payment recipient.
Surname	The surname of the payment recipient.
BranchCode	The branch code of the account where the payment will be deposited.
AccountNumber	The account number of the account where the payment will be deposited.
CustomerCode	The customer code used to identify the payment in your system.
Reference	The reference that will appear on the recipient's bank statement.
Result	Will always be 'Complete'
ResultMessage	Will always be empty

Complete callback example

```

<Response>
  <Result>OK</Result>
  <BatchCode>30104</BatchCode>
  <PaymentResults>
    <Result>
      <AccountNumber>123456789</AccountNumber>
      <BranchCode>632005</BranchCode>
      <FirstName>Example</FirstName>
      <Surname>Callback</Surname>
      <Reference>SALARY</Reference>
      <CustomerCode>000001</CustomerCode>
      <Result>Complete</Result>
      <ResultMessage>Processing Complete</ResultMessage>
      <ResponseCode></ResponseCode>
    </Result>
  </PaymentResults>
</Response>

```

Unpays

An unpaid occurs when the bank rejects a payment. This can happen for many reasons like the recipients account being closed. When this happens, an unpays response is POSTed back to your server using the **CallBackUrl** provided in the initial request.

Note: The unpaid responses can continue for several days after the batch is submitted.

The *Response* root element will consist of the following elements:

Filter	Description
Result	Will always be OK.
BatchCode	The unique code for the batch the unpays belong to.
PaymentResults	The collection of <i>Results</i> elements with more information about each unpaid.

The *PaymentResult* section consists of the following elements:

Element	Description
FirstName	The first name(s) of the payment recipient.
Surname	The surname of the payment recipient.
BranchCode	The branch code of the account where the payment will be deposited.
AccountNumber	The account number of the account where the payment will be deposited.
CustomerCode	The customer code used to identify the payment in your system.
Reference	The reference that will appear on the recipient's bank statement.
Result	The result status which will always be Rejected for an unpaid.
ResultMessage	The reason for the unpaid.
ResponseCode	The unpaid code provided by the bank.

Unpays example

```
<Response>
  <Result>OK</Result>
  <BatchCode>30104</BatchCode>
  <PaymentResults>
    <Result>
      <AccountNumber>123456789</AccountNumber>
      <BranchCode>632005</BranchCode>
      <FirstName>Example</FirstName>
      <Surname>Callback</Surname>
      <Reference>SALARY</Reference>
      <CustomerCode>000001</CustomerCode>
      <Result>Rejected</Result>
      <ResultMessage>ACCOUNT CLOSED</ResultMessage>
      <ResponseCode>12</ResponseCode>
    </Result>
  </PaymentResults>
</Response>
```

Batch Bank Account Verification (BANV)

Overview

A BANV request will submit data relating to the verification of a batch of account numbers against the information provided. Once a batch has been submitted, a CDV check will be performed on the account details provided and any accounts failing that check will be included in the response. Any accounts that passed the CDV check will continue to be processed.

As of Version 2, additional fields have been catered for, namely, Email Address, Telephone Number (specifically a mobile number) and Account Type. These fields are optional and are flagged as such in the tables which follow.

Information regarding test accounts can be accessed here: [BANV Test Accounts](#).

Request URL

<https://secure.rapidpaycollect.co.za/API/Verification>

Request structure

A BANV request consists of 3 separate sections with a root element with the name *APIVerificationRequest*. The *Header* section contains information relating to the BANV batch, the *Records* section is a collection of account details to be used for verification and the *Totals* section is used to verify the information being sent is complete and correct.

Header

A BANV request has a single *Header* section and it consists of the following elements:

Element	Required	Max Length	Description
PsVer	Yes	5	The version of the RapidCollect file format. Currently 2.0.1.
Client	Yes	8	Your unique client code.
Service	Yes	10	The code for the service being used. Must be BANV for bank account verification.
Reference	Yes	50	The batch reference. Allows you to identify the batch in the reports and the verification responses.
CallbackUrl	Yes	250	The call-back URL to be used to send back verification response.
Uniqueld	No	50	Used to prevent duplicate requests

Records

A BANV request has a single *Records* section with multiple *FileContents* sections consisting of the following elements:

Element	Required	Max Length	Description
Initials	No	5	The initials of the account holder.
Name	Yes	100	The account holder name (could be a company name).
IdNumber	No	15	The ID/Passport or company registration number of the account holder.
AccountNumber	Yes	18	The account number of the account to be verified.

BranchCode	Yes	6	The branch code of the account number to be verified.
Reference	No	20	A transaction reference. Can be used to uniquely identify a verification record.
Email	No	50	The email address of the account holder whose account is being verified. A maximum of 100 characters is allowed.
Telephone	No	50	<p>The local (South African) mobile telephone number of the account holder whose account is being verified. Although the field is called Telephone, only mobile numbers are verified.</p> <p>Note: All mobile numbers will be formatted as follows when passed to the bank, however, their original input value will be displayed in the result. All numbers are trimmed of non-numeric values and spaces. Then, 9 characters are extracted starting from the right-hand side of the trimmed value. This value is then prefixed with 0, thus totalling 10 characters, and making up a standard local (South African) mobile number. Examples are provided below;</p> <p>+27 83 987 6543 = 0839876543 987 654 3214 = 0876543214 082 985 6549 = 0829856549</p>
AccountType	No	1	The Account Type associated with the account being verified. This value must be parsed as an Integer. Please refer to Field Lengths and Data Types on the last page for each account type's corresponding number.

Totals

A BANV request has a single *Totals* section and it consists of the following elements:

Element	Required	Description
Records	Yes	The total number of BANV records being submitted.
BranchHash	Yes	The sum of all the branch codes for the BANV records.
AccountHash	Yes	The sum of all the account numbers for the BANV records.

Request Example

```

<APIVerificationRequest>
  <Header>
    <PsVer>2.0.1</PsVer>
    <Client>ZER001</Client>
    <Service>BANV</Service>
    <Reference>Example BANV API Batch</Reference>
    <CallbackUrl>https://example.com/Callback</CallbackUrl>
  </Header>
  <Records>
    <FileContents>
      <Initials>EX</Initials>
      <Name>Example Name</Name>
      <BranchCode>632009</BranchCode>
      <AccountNumber>123456789</AccountNumber>
      <AccountType>0</AccountType>
      <CustomerCode>EXA9292</CustomerCode>
      <Reference>Example Reference</Reference>
    </FileContents>
  </Records>
  <Totals>
    <Records>1</Records>
    <BranchHash>632009</BranchHash>
    <AccountHash>123456789</AccountHash>
  </Totals>
</APIVerificationRequest>

```

Response structure

The standard response structure will be returned. All response data are submitted as a form data in a POST action (i.e. application/x-www-form-urlencoded) with the key "response".

If any of the BANV *FileContents* records failed the CDV check they will be returned as a *CDVResults* element with a collection of *Result* elements outlining the reasons for the account's rejection. The *Result* section consists of the following elements:

Filter	Description
Result	The result of the CDV check. Will always be Invalid for a failed account.
Message	A message giving the reason for the rejection.
AccountNumber	The account number that failed the CDV test.

Please note: If any of the accounts passed the CDV check the batch will be accepted and will proceed to be verified with the bank.

Response Example

```

<Response>
  <Result>OK</Result>
  <BatchCode>285904</BatchCode>
  <TotalFeeExcludingVAT>1.10</TotalFeeExcludingVAT>
  <CDVResults>
    <Result>
      <AccountNumber>123456789</AccountNumber>
      <BranchCode>632009</BranchCode>
      <CustomerCode>EXA9292</CustomerCode>
      <Reference>Example Reference</Reference>
      <Result>Valid</Result>
    </Result>
  </CDVResults>
</Response>

```

Validation Responses

A validation response occurs when the bank returns the result of the bank account verification. When this happens, a validation response is POSTed back to your server using the *CallBackUrl* provided in the initial request.

Note: The validation responses are POSTed back to the *CallBackUrl* whenever a response is returned by the bank. This is likely to occur multiple times as each bank has different response times.

The *Response* root element will consist of the following elements:

Filter	Description
Result	Will always be OK.
BatchCode	The unique code for the batch the validations belong to.

The *BanvResults* section consists of the following elements:

Element	Description
AccountNumber	The account number of the account that was verified.
IDNumber	The ID/Passport or company registration number of the account holder.
Initials	The initials of the account holder.
Name	The account holder name (could be a company name).
Reference	The batch reference. Allows you to identify the batch in the reports.
Flags	Y/N/Blank characters indicating a corresponding result. See section below.
Result	The result status which is a text representation of the verification result.

BANV Flags

The flags of the verification result are a character indication of each of the 8 verification options. Not all banks are able to supply all the information in which case the character will be a blank space. An example Flags return would be: YYYYYYYYYY which is the equivalent to a result of "A/c exists, ID matches, Inits match, Name matches, A/c open, accepts Dr, accepts Cr & open more than 3 months, Email matches, Mobile matches & Account type matches.". If the first 5 flags are "Y" then then account can be considered valid.

Note: Names and Initials must be identical to the name or initial the bank has on record otherwise it will not match. If the Initials provided are "WJ" and the bank has just "W" on record, then the initials flag will be displayed as an "N".

The 11 flag options are:

Position	Options
1	Account exists
2	ID number or Company registration match
3	Initials match (should be blank to match for company)
4	Surname or company name match
5	Account open
6	Account accepts debits
7	Account accepts credits
8	Accounts open for three or more months
9	Email matches
10	Mobile matches
11	Account type matches

Validations Response Example

```
<Response>
  <Result>OK</Result>
  <BatchCode>1195751</BatchCode>
  <BanvResults>
    <Result>
      <AccountNumber>123456789</AccountNumber>
      <IDNumber>830918568084</IDNumber>
      <Initials>E</Initials>
      <Name>Example</Name>
      <Reference>Example 1</Reference>
      <Flags>YY YYYY N</Flags>
      <Result>A/c exists, ID matches, Initials not validated, Name matches, A/c open,
accepts Dr, accepts Cr, open more than 3 months, Email not validated, Mobile not
validated & Account type doesn't match</Result>
    </Result>
  </BanvResults>
</Response>
```

Real-time Bank Account Verification (BANVR)

Overview

A BANVR request will submit data relating to the verification of a single account number against the information provided. Real-time account verification has a guaranteed response time from the bank of 60 seconds. If no response has been received after 60 seconds, then no result is returned. Due to the real-time response, no callback URL is supplied as no callback will be sent.

As of Version 2, additional fields have been catered for, namely, Email Address, Telephone Number (specifically a mobile number) and Account Type. These fields are optional and are flagged as such in the tables which follow.

Request URL

<https://secure.rapidpaycollect.co.za/API/RealtimeVerification>

Request structure

A BANVR request consists of 3 separate sections with a root element with the name *APIRealTimeVerificationRequest*. The *Header* section contains information relating to the BANVR batch, the *Records* section is a collection of account details to be used for verification (only a single account is valid for real-time verification) and the *Totals* section is used to verify the information being sent is complete and correct.

Header

A BANVR request has a single *Header* section and it consists of the following elements:

Element	Required	Max Length	Description
PsVer	Yes	5	The version of the Rapid Pay Collect file format. Currently 2.0.1.
Client	Yes	8	Your unique client code.
Reference	Yes	50	The batch reference. Allows you to identify the batch in the reports and the verification responses.
Uniqueld	No	50	Used to prevent duplicate requests
CallbackUrl	No	250	If a callback URL is supplied and the real-time verification fails, then a standard BANV service will be used and the results will be sent to the callback URL supplied

Records

A BANVR request has a single *Records* section with a single *FileContents* section consisting of the following elements:

Element	Required	Max Length	Description
Initials	No	5	The initials of the account holder.
Name	Yes	100	The account holder name (could be a company name).
IdNumber	No	15	The ID/Passport or company registration number of the account holder.
AccountNumber	Yes	18	The account number of the account to be verified.
BranchCode	Yes	6	The branch code of the account number to be verified.

Reference	No	20	A transaction reference. Can be used to uniquely identify a verification record.
Email	No	50	The email address of the account holder whose account is being verified. A maximum of 100 characters is allowed.
Telephone	No	50	<p>The local (South African) mobile telephone number of the account holder whose account is being verified. Although the field is called Telephone, only mobile numbers are verified.</p> <p>Note: All mobile numbers will be formatted as follows when passed to the bank, however, their original input value will be displayed in the result. All numbers are trimmed of non-numeric values and spaces. Then, 9 characters are extracted starting from the right-hand side of the trimmed value. This value is then prefixed with 0, thus totalling 10 characters, and making up a standard local (South African) mobile number. Examples are provided below;</p> <p>+27 83 987 6543 = 0839876543 987 654 3214 = 0876543214 082 985 6549 = 0829856549</p>
AccountType	No	1	The Account Type associated with the account being verified. This value must be parsed as an Integer. Please refer to Field Lengths and Data Types on the last page for each account types corresponding number.

Totals

A BANVR request has a single *Totals* section and it consists of the following elements:

Element	Required	Description
Records	Yes	The total number of BANVR records being submitted.
BranchHash	Yes	The sum of all the branch codes for the BANVR record.
AccountHash	Yes	The sum of all the account numbers for the BANVR record.

Request Example

```

<APIRealTimeVerificationRequest>
  <Header>
    <PsVer>2.0.1</PsVer>
    <Client>ZER001</Client>
    <Reference>Example BANVR API Batch</Reference>
    <CallbackUrl>https://example.com/Callback</CallbackUrl>
  </Header>
  <Records>
    <FileContents>
      <Initials>EX</Initials>
      <Name>Example Name</Name>
      <BranchCode>632009</BranchCode>
      <AccountNumber>123456789</AccountNumber>
      <AccountType>0</AccountType>
      <CustomerCode>EXA9292</CustomerCode>
      <Reference>Example Reference</Reference>
    </FileContents>
  </Records>
  <Totals>
    <Records>1</Records>
    <BranchHash>632009</BranchHash>
    <AccountHash>123456789</AccountHash>
  </Totals>
</APIRealTimeVerificationRequest>

```

Response structure

The standard response structure will be returned with a *BanvRealtimeResult* element containing the validation results.

The *Response* root element will consist of the following elements:

Filter	Description
Result	Will always be OK.
BatchCode	The unique code for the batch the validation belong to.

The *BanvRealTimeResult* section consists of the following elements:

Element	Description
AccountNumber	The account number of the account that was verified.
IDNumber	The ID/Passport or company registration number of the account holder.
Initials	The initials of the account holder.
AccountType	The account type value as provided for verification
Email	The email address which was provided for verification
Telephone	The mobile telephone number which was provided for verification
Name	The account holder name (could be a company name).
Reference	The batch reference. Allows you to identify the batch in the reports.
Flags	Y/N/Blank characters indicating a corresponding result. See section below.
Result	The result status which is a text representation of the verification result.

If the real-time was unable to be processed, the **Result** field will have the text “Unprocessed” and the **Flags** field will have the text “U”.

The *CDVResults* element will always be returned regardless of whether or not the account passed the CDV check.

The *Result* section consists of the following elements:

Filter	Description
Result	The result of the CDV check. Will always be Invalid for a failed account.
Message	A message giving the reason for the rejection.
AccountNumber	The account number that failed the CDV test.

BANVR Flags

The flags of the verification result are a character indication of each of the 8 verification options. Not all banks are able to supply all the information in which case the character will be a blank space. An example Flags return would be: YYYYYYYYYY which is the equivalent to a result of "A/c exists, ID matches, Initials match, Name matches, A/c open, accepts Dr, accepts Cr, open more than 3 months, Email matches, Mobile matches & Account type matches."

If the first 5 flags are "Y" then then account can be considered valid.

Note: Names and Initials must be identical to the name or initial the bank has on record otherwise it will not match. If the Initials provided are "WJ" and the bank has just "W" on record, then the initials flag will be displayed as an "N".

The 11 flag options are:

Position	Options
1	Account exists
2	ID number or Company registration match
3	Initials match (should be blank to match for company)
4	Surname or company name match
5	Account open
6	Account accepts debits
7	Account accepts credits
8	Accounts open for three or more months
9	Email matches
10	Mobile matches
11	Account type matches

Response Example

```

<Response>
  <Result>OK</Result>
  <BatchCode>285905</BatchCode>
  <TotalFeeExcludingVAT>2.50</TotalFeeExcludingVAT>
  <BanvRealtimeResult>
    <AccountNumber>123456789</AccountNumber>
    <AccountType>Unknown</AccountType>
    <Flags>YYUNYYYYUUN</Flags>
    <IDNumber>5410126456087</IDNumber>
    <Initials>EX</Initials>
    <Name>EXAMPLE NAME</Name>
    <Reference>Example Reference</Reference>
    <Result>A/c exists, ID matches, Initials not validated, Name doesn't match, A/c
open, accepts Dr, accepts Cr, open more than 3 months, Email not validated, Mobile not
validated & Account type doesn't match </Result>
  </BanvRealtimeResult>
  <CDVResults>
    <Result>
      <AccountNumber>123456789</AccountNumber>
      <BranchCode>632009</BranchCode>
      <CustomerCode>EXA9292</CustomerCode>
      <Reference>Example Reference</Reference>
      <Result>Valid</Result>
    </Result>
  </CDVResults>
</Response>

```


Check Digit Verification (CDV)

Overview

A CDV request will submit data relating to the verification of a batch of account numbers against a bank supplied algorithm to determine if the account numbers fall within a valid range for that bank.

Request URL

<https://secure.rapidpaycollect.co.za/API/CDV>

Request structure

A CDV request consists of 3 separate sections with a root element with the name *APICDVRequest*. The *Header* section contains information relating to the CDV batch, the *Records* section is a collection of account details to be verified and the *Totals* section is used to verify the information being sent is complete and correct.

Header

A CDV request has a single *Header* section and it consists of the following elements:

Element	Required	Description
PsVer	Yes	The version of the Rapid Pay Collect file format. Currently 2.0.1.
Client	Yes	Your unique client code.
Service	Yes	The code for the service being used. Must be CDV for check digit verification.
Reference	Yes	The batch reference. Allows you to identify the batch on the history page on the secure website.
Uniqueld	No	Used to prevent duplicate requests

Records

A CDV request has a single *Records* section with multiple *FileContents* sections consisting of the following elements:

Element	Required	Description
AccountNumber	Yes	The account number of the account to be verified.
BranchCode	Yes	The branch code of the account number to be verified.
CustomerCode	No	The customer identifier. This will be returned with results, CDV results, etc.

Totals

A CDV request has a single *Totals* section and it consists of the following elements:

Element	Required	Description
Records	Yes	The total number of CDV records being submitted.
BranchHash	Yes	The sum of all the branch codes for the CDV records.
AccountHash	Yes	The sum of all the account numbers for the CDV records.

Request Example

```

<APICDVRequest>
  <Header>
    <PsVer>2.0.1</PsVer>
    <Client>ZER001</Client>
    <Service>CDV</Service>
    <Reference>Example CDV API Batch</Reference>
  </Header>
  <Records>
    <FileContents>
      <BranchCode>632005</BranchCode>
      <AccountNumber>123456789</AccountNumber>
    </FileContents>
  </Records>
  <Totals>
    <Records>1</Records>
    <BranchHash>632005</BranchHash>
    <AccountHash>123456789</AccountHash>
  </Totals>
</APICDVRequest>

```

Response structure

The standard response structure will be returned with a *CDVResults* element containing the validation results.

The *Response* root element will consist of the following elements:

Filter	Description
Result	Will always be OK.
BatchCode	<i>Deprecated.</i> Will be empty.
TotalFeeIncludingVat	<i>Deprecated.</i> Will be 0.
Uniqueld	The unique Id submitted

The *Result* section consists of the following elements:

Filter	Description
Result	The result of the CDV check. Will always be Invalid for a failed account.
Message	A message giving the reason for the rejection.
AccountNumber	The account number that failed the CDV test.
BranchCode	The branch code submitted
Reference	The reference submitted

Response Example - Valid

```

<Response>
  <Result>OK</Result>
  <BatchCode></BatchCode>
  <TotalFeeExcludingVAT>0</TotalFeeExcludingVAT>
  <CDVResults>
    <Result>
      <AccountNumber>123456789</AccountNumber>
      <BranchCode>632005</BranchCode>
      <Result>Valid</Result>
    </Result>
  </CDVResults>
</Response>

```

Response Example - Invalid

```
<Response>
  <Result>OK</Result>
  <BatchCode></BatchCode>
  <TotalFeeExcludingVAT>0</TotalFeeExcludingVAT>
  <CDVResults>
    <Result>
      <AccountNumber>123456759</AccountNumber>
      <BranchCode>632005</BranchCode>
      <Message>Account number is not valid for ABSA Bank.</Message>
      <Result>Invalid</Result>
    </Result>
  </CDVResults>
</Response>
```

Debit Orders

Overview

A debit orders request will submit data relating to a debit order batch. Once a debit order batch has been submitted, a CDV check will be performed on the account details provided and any accounts failing that check will be included in the response. Any accounts that passed the CDV check will continue to be processed.

If an account is rejected by a bank (usually because it has been closed) an unpaids response is created and POSTed back to the call-back URL provided.

Request URL

<https://secure.rapidpaycollect.co.za/API/DebitOrder>

Request structure

A debit order request consists of 3 separate sections with a root element with the name *APIDebitOrdersRequest*. The *Header* section contains information relating to the debit order batch, the *DebitOrders* sections is a collection of account details to be used as the destination for the debit orders and the *Totals* section is used to verify the information being sent is complete and correct.

Header

A debit order request has a single *Header* section and it consists of the following elements:

Element	Required	Description
PsVer	Yes	The version of the Rapid Pay Collect file format. Currently 2.0.1.
Client	Yes	Your unique client code.
Service	Yes	Always DebitOrder.
ServiceType	Yes	Always 2Day.
DueDate	Yes	The date the batch should be submitted to the bank for processing.
CallBackUrl	Yes	The call-back URL to be used to send back unpaids.
Reference	Yes	The reference used to identify the batch for reporting
Uniqueld	No	Used to prevent duplicate requests

Debit Orders

A debit order request has a single *DebitOrders* section with multiple *FileContents* sections consisting of the following elements:

Element	Required	Description
Initials	No	The initials of the account holder.
FirstNames	Yes	The first name(s) of the account holder.
Surname	Yes	The surname of the account holder.
BranchCode	Yes	The branch code of the account to be debited.
AccountNumber	Yes	The account number of the account to be debited.
FileAmount	Yes	The amount to be debited
AmountMultiplier	Yes	Usually 1 if the FileAmount is specified in Rands otherwise 100 if the FileAmount is specified in cents.
CustomerCode	No	The customer code used to identify the debit order in your system.

Reference	Yes	The reference that will appear on the recipient's bank statement.
------------------	-----	---

Totals

A debit order request has a single *Totals* section and it consists of the following elements:

Element	Required	Description
Records	Yes	The total number of debit order records being submitted.
Amount	Yes	The total value of the debit order records being submitted.
BranchHash	Yes	The sum of all the branch codes for the debit order records.
AccountHash	Yes	The sum of all the account numbers for the debit order records.

Request Example

```
<APIDebitOrdersRequest>
  <Header>
    <PsVer>2.0.1</PsVer>
    <Client>ZER001</Client>
    <Duedate>20200625</Duedate>
    <Service>DebitOrder</Service>
    <ServiceType>2day</ServiceType>
    <Reference>Example Debit Order API Batch</Reference>
    <CallBackUrl>https://example.com/Callback</CallBackUrl>
  </Header>
  <DebitOrders>
    <FileContents>
      <Initials>EX</Initials>
      <FirstNames>Example</FirstNames>
      <Surname>Debtor</Surname>
      <BranchCode>632009</BranchCode>
      <AccountNumber>123456789</AccountNumber>
      <FileAmount>549.01</FileAmount>
      <AccountType>0</AccountType>
      <AmountMultiplier>1</AmountMultiplier>
      <CustomerCode>EXA9292</CustomerCode>
      <Reference>Example Reference</Reference>
    </FileContents>
  </DebitOrders>
  <Totals>
    <Records>1</Records>
    <Amount>549.01</Amount>
    <BranchHash>632009</BranchHash>
    <AccountHash>123456789</AccountHash>
  </Totals>
</APIDebitOrdersRequest>
```

Response structure

The standard response structure will be returned with an extra element named `BatchValueSubmitted` which contains the total value of the batch submitted (less any accounts that failed the CDV check).

The results of the CDV check will be returned as a *CDVResults* element with a collection of *Result* elements outlining the reasons for the accounts rejection. The *Result* section consists of the following elements:

Filter	Description
Result	The result of the CDV check. Will always be Invalid for a failed account.
Message	A message giving the reason for the rejection.
AccountNumber	The account number that failed the CDV test.

Please note: If any of the accounts passed the CDV check the batch will be accepted and will proceed to be processed.

Response Example

```
<Response>
  <Result>OK</Result>
  <BatchCode>285900</BatchCode>
  <BatchValueSubmitted>549.01</BatchValueSubmitted>
  <TotalFeeExcludingVAT>3.00</TotalFeeExcludingVAT>
  <CDVResults>
    <Result>
      <AccountNumber>123456789</AccountNumber>
      <BranchCode>632009</BranchCode>
      <CustomerCode>EXA9292</CustomerCode>
      <Reference>Example Reference</Reference>
      <Result>Valid</Result>
    </Result>
  </CDVResults>
</Response>
```

Complete Callback

Due to the manner in which the banks operate, a debit order is assumed to be successful unless an unpaids response is returned. However, it remains useful to subscribe to a callback once a debit order has been accepted by the bank in order to trigger a workflow based on the success of the debit order. A complete response is POSTed back to your server using the *CallbackUrl* provided in the initial request.

Unlike unpaids callbacks, complete callbacks need to be enabled on your client profile in order to be sent. Please contact help@rapidpaycollect.co.za to request activation of the complete callback.

Note: The complete responses are sent once the bank has accepted the transactions and could be up to two days before the debit order is actioned on the target account.

The *Response* root element will consist of the following elements:

Filter	Description
Result	Will always be OK.
BatchCode	The unique code for the batch.
DebitOrderResults	The collection of <i>Results</i> elements with more information about each completed transaction.

The *PaymentResult* section consists of the following elements:

Element	Description
FirstName	The first name(s) of the payment recipient.
Surname	The surname of the payment recipient.
BranchCode	The branch code of the account where the payment will be deposited.

AccountNumber	The account number of the account where the payment will be deposited.
CustomerCode	The customer code used to identify the payment in your system.
Reference	The reference that will appear on the recipient's bank statement.
Result	Will always be 'Complete'
ResultMessage	Will always be empty

Complete callback example

```
<Response>
  <Result>OK</Result>
  <BatchCode>152933</BatchCode>
  <DebitOrderResults>
    <Result>
      <AccountNumber>62541931574</AccountNumber>
      <BranchCode>250655</BranchCode>
      <FirstName>Keabetswe</FirstName>
      <Surname>Sojane</Surname>
      <Reference>Fitness Gym</Reference>
      <CustomerCode>430</CustomerCode>
      <Result>Complete</Result>
      <ResultMessage></ResultMessage>
      <ResponseCode></ResponseCode>
    </Result>
  </DebitOrderResults>
</Response>
```

Unpays

An unpaid occurs when the bank rejects a debit order. This can happen for many reasons like the account being closed. When this happens, an unpays response is POSTed back to your server using the *CallBackUrl* provided in the initial request. All response data are submitted as a form data in a POST action (i.e. application/x-www-form-urlencoded) with the key "response".

Note: The unpaid responses can continue for several days after the batch is submitted.

The *Response* root element will consist of the following elements:

Filter	Description
Result	Will always be OK.
BatchCode	The unique code for the batch the unpays belong to.
DebitOrderResults	The collection of <i>Results</i> elements with more information about each unpaid.

The *DebitOrderResults* section consists of the following elements:

Element	Description
FirstName	The first name(s) of the account holder.
Surname	The surname of the account holder.
BranchCode	The branch code of the account to be debited.
AccountNumber	The account number of the account to be debited.
CustomerCode	The customer code used to identify the debit order in your system.
Reference	The reference that would have appeared on the recipient's bank statement.
Result	The result status which will always be Rejected for an unpaid.
ResultMessage	The reason for the unpaid.

Unpays Example

```
<Response>
  <Result>OK</Result>
  <BatchCode>1193910</BatchCode>
  <DebitOrderResults>
    <Result>
      <AccountNumber>123456789</AccountNumber>
      <BranchCode>632005</BranchCode>
      <FirstName>Example</FirstName>
      <Surname>Debtor</Surname>
      <Reference>Reference 1</Reference>
      <CustomerCode>12345</CustomerCode>
      <Result>Rejected</Result>
      <ResultMessage>NOT PROVIDED FOR</ResultMessage>
      <ResponseCode>2</ResponseCode>
    </Result>
  </DebitOrderResults>
</Response>
```


Field Lengths and Data Types

Overview

The table below lists the respective data types and lengths of each of the fields.

Element	Description	Data Type	Max Length
Initials	The initials of the payment recipient.	Alphanumeric	50
FirstNames	The first name(s) of the payment recipient.	Alphanumeric	250
Surname	The surname of the payment recipient.	Alphanumeric	250
ContactNumber	The contact number to be used for the search	Alphanumeric	255
IdNumber	The ID number of the person linked to the contact number	Numeric	100
BranchCode	The branch code of the account where the payment will be deposited.	Numeric	10
AccountNumber	The account number of the account where the payment will be deposited.	Numeric	15
FileAmount	The amount to be deposited.	Decimal	18 Character and 2 decimal points
AmountMultiplier	Usually 1 if the FileAmount is specified in Rands otherwise 100 if the FileAmount is specified in cents.	Numeric	3
AccountType	The type of account where the funds will be deposited. This can be left blank or 0 can be used if you don't know. Only 0 – 6 can be used. Unknown = 0 Current/Cheque = 1 Savings = 2 Transmission = 3 Bond = 4 Subscription Share = 6	Numeric	1
CustomerCode	The customer code used to identify the payment in your system.	Alphanumeric	50
Reference	The reference that will appear on the recipient's bank statement.	Alphanumeric	20

Batch BANV Test Accounts

Overview

BANV Integrators have the option to test their integration against the Rapid Pay Collect Test API, using a predefined list of account numbers. The table which follows lists all predefined test accounts as well as their expected responses.

Note

Test accounts should only be processed using the test API Endpoint, test accounts processed against the live service, will be processed as actual BANV requests, and will potentially incur a fee.

The test service only makes provision for testing one account during any given request. Should integrators attempt to send more than one account during a single API request, the service will return an exception.

Test records will go through basic request validation, however, they will only pass or fail Check Digit Validation (CDV) or Bank Account Verification (BANV) according to their assigned, expected result, as defined in the table which follows.

Please make use of the following API end point for sending requests to the test service.

<https://sandbox.rapidpaycollect.co.za/API/Verification>

Please contact Rapid Pay Collect with regards to setting up your test account, as this will use a different API Key from your live account.

Please take note of the API Request/Response examples following this table.

Test Accounts

Account Number	Result	Expected CDV Result	Expected Call-back result Flags	Expected Call-Back Result Message
1222222221	OK	Valid	YYYYYYYYYY	A/c exists, ID matches, Initials match, Name matches, A/c open, accepts Dr, accepts Cr, open more than 3 months, Email matches, Mobile matches & Account type matches
1333333331	OK	Valid	YYNYYYYYY	A/c exists, ID matches, Initials match, Name matches, A/c closed , accepts Dr, accepts Cr, open more than 3 months, Email matches, Mobile matches & Account type matches
1444444441	OK	Valid	YYNYYNYYN	A/c exists, ID matches, Initials don't match , Name matches, A/c open, doesn't accept Dr , accepts Cr, not open more than 3 months , Email matches, Mobile doesn't match & Account type matches
1555555551	OK	Valid	YYYYYYYYYN	A/c exists, ID matches, Initials match, Name matches, A/c open, accepts Dr,

				accepts Cr, open more than 3 months, Email matches, Mobile matches & Account type doesn't match
16666666661	OK	Valid	YNNNYYYYNNY	A/c exists, ID doesn't match, Initials don't match, Name doesn't match , A/c open, accepts Dr, accepts Cr, open more than 3 months, Email doesn't match, Mobile doesn't match & Account type matches
17777777771	OK	Valid	Empty	Bank result is undetermined
18888888881	OK	Valid	N	A/c doesn't exist
19999999991	OK	Valid	Empty	No response from agent bank
21111111112	Error	Invalid - Invalid account for bank branch	No Call-back	No Call-back
23333333332	Error	Invalid - Account number cannot be zero.	No Call-back	No Call-back
24444444442	Error	Invalid - Account number is too long (appears to be a credit card).	No Call-back	No Call-back
25555555552	Error	Invalid - Account Type not found	No Call-back	No Call-back
26666666662	Error	Invalid - Account number could not be verified against the 'weighting digits'.	No Call-back	No Call-back

Request Example

```

<APIVerificationRequest>
  <Header>
    <PsVer>2.0.1</PsVer>
    <Client>ZER001</Client>
    <Service>BANV</Service>
    <Reference>Example BANV API Batch</Reference>
  </Header>
  <Records>
    <FileContents>
      <Initials>EX</Initials>
      <Name>Example Name</Name>
      <BranchCode>12345</BranchCode>
      <AccountNumber>1222222221</AccountNumber>
      <AccountType>0</AccountType>
      <CustomerCode>EXA9292</CustomerCode>
      <Reference>Example Reference</Reference>
    </FileContents>
  </Records>
  <Totals>
    <Records>1</Records>
    <BranchHash>12345</BranchHash>
    <AccountHash>1222222221</AccountHash>
  </Totals>
</APIVerificationRequest>

```

Response Example

```

<Response>
  <Result>OK</Result>
  <BatchCode>3542578</BatchCode>
  <TotalFeeExcludingVAT>1.14</TotalFeeExcludingVAT>
  <CDVResults>
    <Result>
      <AccountNumber>1222222221</AccountNumber>
      <BranchCode>012345</BranchCode>
      <CustomerCode>EXA9292</CustomerCode>
      <Reference>Example Reference</Reference>
      <Result>Valid</Result>
    </Result>
  </CDVResults>
</Response>

```

Call-Back Example

```
<Response>
  <Result>OK</Result>
  <BatchCode>1195751</BatchCode>
  <BanvResults>
    <Result>
      <AccountNumber>1222222221</AccountNumber>
      <IDNumber>830918568084</IDNumber>
      <Initials>E</Initials>
      <Name>Example</Name>
      <Reference>Example 1</Reference>
      <Flags>YYYYYYYYYY</Flags>
      <Result> A/c exists, ID matches, Initials match, Name matches, A/c open, accepts
Dr, accepts Cr, open more than 3 months, Email matches, Mobile matches & Account type
matches</Result>
    </Result>
  </BanvResults>
</Response>
```

Real-Time BANV Test Accounts

Overview

BANV Integrators have the option to test their integration against the Rapid Pay Collect Test API, using a predefined list of account numbers. The table which follows lists all predefined test accounts as well as their expected responses.

Note

Test accounts should only be processed using the test API Endpoint, test accounts processed against the live service, will be processed as actual BANV requests, and will potentially incur a fee.

The test service only makes provision for testing one account during any given request. Should integrators attempt to send more than one account during a single API request, the service will return an exception.

Test records will go through basic request validation, however, they will only pass or fail Check Digit Validation (CDV) or Bank Account Verification (BANV) according to their assigned, expected result, as defined in the table which follows.

Please make use of the following API end point for sending requests to the test service.

<https://sandbox.rapidpaycollect.co.za/API/RealTimeVerification>

Please contact Rapid Pay Collect with regards to setting up your test account, as **this will use a different API Key from your live account.**

Please take note of the API Request/Response examples following this table.

Test Accounts

Account Number	Result	Expected CDV Result	Expected result Flags	Expected Result Message
1222222221	OK	Valid	YYYYYYYYYY	A/c exists, ID matches, Initials match, Name matches, A/c open, accepts Dr, accepts Cr, open more than 3 months, Email matches, Mobile matches & Account type matches
1333333331	OK	Valid	YYNYYYYYYY	A/c exists, ID matches, Initials match, Name matches, A/c closed , accepts Dr, accepts Cr, open more than 3 months, Email matches, Mobile matches & Account type matches
1444444441	OK	Valid	YNYNYNYNY	A/c exists, ID matches, Initials don't match , Name matches, A/c open, doesn't accept Dr , accepts Cr, not open more than 3 months , Email matches, Mobile doesn't match & Account type matches
1555555551	OK	Valid	YYYYYYYYYN	A/c exists, ID matches, Initials match, Name matches, A/c open, accepts Dr,

				accepts Cr, open more than 3 months, Email matches, Mobile matches & Account type doesn't match
16666666661	OK	Valid	YNNNYYYYNNY	A/c exists, ID doesn't match, Initials don't match, Name doesn't match , A/c open, accepts Dr, accepts Cr, open more than 3 months, Email doesn't match, Mobile doesn't match & Account type matches
17777777771	OK	Valid	Empty	Bank result is undetermined
18888888881	OK	Valid	N	A/c doesn't exist
19999999991	OK	Valid	Empty	No response from agent bank
21111111112	Error	Invalid - Invalid account for bank branch	Empty	Empty
23333333332	Error	Invalid - Account number cannot be zero.	Empty	Empty
24444444442	Error	Invalid - Account number is too long (appears to be a credit card).	Empty	Empty
25555555552	Error	Invalid - Account Type not found	Empty	Empty
26666666662	Error	Invalid - Account number could not be verified against the 'weighting digits'.	Empty	Empty

Request Example

```

<APIRealTimeVerificationRequest>
  <Header>
    <PsVer>2.0.1</PsVer>
    <Client>ZER001</Client>
    <Reference>Example BANVR API Batch</Reference>
    <CallbackUrl>https://example.com/Callback</CallbackUrl>
  </Header>
  <Records>
    <FileContents>
      <Initials>EX</Initials>
      <Name>Example Name</Name>
      <BranchCode>632009</BranchCode>
      <AccountNumber>1222222221</AccountNumber>
      <AccountType>0</AccountType>
      <CustomerCode>EXA9292</CustomerCode>
      <Reference>Example Reference</Reference>
    </FileContents>
  </Records>
  <Totals>
    <Records>1</Records>
    <BranchHash>632009</BranchHash>
    <AccountHash>1222222221</AccountHash>
  </Totals>
</APIRealTimeVerificationRequest>

```

Response Example

```

<Response>
  <Result>OK</Result>
  <BatchCode>285905</BatchCode>
  <TotalFeeExcludingVAT>2.50</TotalFeeExcludingVAT>
  <BanvRealtimeResult>
    <AccountNumber>1222222221</AccountNumber>
    <AccountType>Unknown</AccountType>
    <Flags>YYYYYYYYYY</Flags>
    <IDNumber>5410126456087</IDNumber>
    <Initials>EX</Initials>
    <Name>EXAMPLE NAME</Name>
    <Reference>Example Reference</Reference>
    <Result>A/c exists, ID matches, Initials match, Name matches, A/c open, accepts
Dr, accepts Cr, open more than 3 months, Email matches, Mobile matches & Account type
matches</Result>
  </BanvRealtimeResult>
  <CDVResults>
    <Result>
      <AccountNumber>1222222221</AccountNumber>
      <BranchCode>632009</BranchCode>
      <CustomerCode>EXA9292</CustomerCode>
      <Reference>Example Reference</Reference>
      <Result>Valid</Result>
    </Result>
  </CDVResults>
</Response>

```


CDV Test Accounts

Overview

CDV Integrators have the option to test their integration against the Rapid Pay Collect Test API, using a predefined list of account numbers. The table which follows lists all predefined test accounts as well as their expected responses.

Note

Test accounts should only be processed using the test API Endpoint, test accounts processed against the live service, will be processed as actual CDV requests.

The test service only makes provision for testing one account during any given request. Should integrators attempt to send more than one account during a single API request, the service will return an exception.

Test records will go through basic request validation, however, they will only pass or fail Check Digit Validation (CDV) according to their assigned, expected result, as defined in the table which follows.

Please make use of the following API end point for sending requests to the test service.

<https://sandbox.rapidpaycollect.co.za/API/CDV>

Please contact Rapid Pay Collect with regards to setting up your test account, as **this will use a different API Key from your live account.**

Please take note of the API Request/Response examples following this table.

Test Accounts

Account Number	Result	Expected CDV Result
1222222221	OK	Valid Account Number
1333333331	OK	Bank does not support account verification
1444444441	OK	Invalid account for bank branch.
0000000000	OK	Account number cannot be zero.
16656646616	OK	Account number could not be verified against the 'weighting digits'.
1999999991	OK	Invalid for bank branch.
2444444442	OK	Account number is too long (appears to be a credit card).
2555555552	OK	Account Type not found.
2666666662	Error	This account has been blocked
1222222221	Error	Valid Account Number
1333333331	Error	Bank does not support account verification

Request Example

```
<APICDVRequest>
  <Header>
    <PsVer>2.0.1</PsVer>
    <Client>ZER001</Client>
    <Service>CDV</Service>
    <Reference>Example CDV API Batch</Reference>
  </Header>
  <Records>
    <FileContents>
      <BranchCode>632005</BranchCode>
      <AccountNumber>1222222221</AccountNumber>
    </FileContents>
  </Records>
  <Totals>
    <Records>1</Records>
    <BranchHash>632005</BranchHash>
    <AccountHash>1222222221</AccountHash>
  </Totals>
</APICDVRequest>
```

Response Example

```
<Response>
  <Result>OK</Result>
  <BatchCode></BatchCode>
  <TotalFeeExcludingVAT>0</TotalFeeExcludingVAT>
  <CDVResults>
    <Result>
      <AccountNumber>1222222221</AccountNumber>
      <BranchCode>632005</BranchCode>
      <Result>Valid</Result>
    </Result>
  </CDVResults>
</Response>
```

Payments Test Accounts

Overview

Payment Integrators have the option to test their integration against the Rapid Pay Collect Test API, using a predefined list of account numbers. The table which follows lists all predefined test accounts as well as their expected responses.

Note

Test accounts should only be processed using the test API Endpoint, test accounts processed against the live service, will be processed as actual Payment requests, and will potentially incur a fee.

The test service only makes provision for testing one account during any given request. Should integrators attempt to send more than one account during a single API request, the service will return an exception.

Test records will go through basic request validation, however, they will only pass or fail Check Digit Validation (CDV) or successfully process a dummy payment according to their assigned, expected result, as defined in the table which follows.

Please make use of the following API end point for sending requests to the test service.

<https://sandbox.rapidpaycollect.co.za/API/Payments>

Please contact Rapid Pay Collect with regards to setting up your test account, as **this will use a different API Key from your live account.**

Please take note of the API Request/Response examples following this table.

Test Accounts

Account Number	Result	Expected CDV Result	Expected Call-back result	Expected Call-Back Result Message
1222222221	OK	CDV Passed	Complete	Processing Complete
1333333331	Error	Account Type not found.	Rejected	CDV Failed
1444444441	OK	Valid Account Number	Rejected	ACCOUNT FROZEN
1555555551	OK	Valid Account Number	Rejected	DEBITS NOT ALLOWED
1666666661	OK	Valid Account Number	Rejected	INSUFFICIENT FUNDS
1777777771	OK	Valid Account Number	Rejected	Account Closed
1888888881	OK	Valid Account Number	Rejected	AUTHORIZATION CANCELLED
1999999991	OK	Valid Account Number	Rejected	DISTRIBUTION UPFRONT REJECTION
2111111112	OK	Valid Account Number	Rejected	BANKSERV REJECTION

2333333332	OK	Valid Account Number	Rejected	CLIENT DID NOT AUTHORISE DEBIT
------------	----	----------------------	----------	--------------------------------

Request Example

```

<APIPaymentsRequest>
  <Header>
    <PsVer>2.0.1</PsVer>
    <Client>ZER001</Client>
    <DueDate>20200625</DueDate>
    <Service>Creditors</Service>
    <ServiceType>1day</ServiceType>
    <Reference>Example Batch</Reference>
    <CallBackUrl>https://example.com/Callback</CallBackUrl>
  </Header>
  <Payments>
    <FileContents>
      <Initials>EX</Initials>
      <FirstNames>Example</FirstNames>
      <Surname>Recipient</Surname>
      <BranchCode>632009</BranchCode>
      <AccountNumber>1222222221</AccountNumber>
      <FileAmount>549.01</FileAmount>
      <AccountType>0</AccountType>
      <AmountMultiplier>1</AmountMultiplier>
      <CustomerCode>EXA9292</CustomerCode>
      <Reference>Example Reference</Reference>
    </FileContents>
  </Payments>
  <Totals>
    <Records>1</Records>
    <Amount>549.01</Amount>
    <BranchHash>632009</BranchHash>
    <AccountHash>123456789</AccountHash>
  </Totals>
</APIPaymentsRequest>

```

Response Example

```

<Response>
  <Result>OK</Result>
  <BatchCode>285902</BatchCode>
  <BatchValueSubmitted>549.01</BatchValueSubmitted>
  <TotalFeeExcludingVAT>2.00</TotalFeeExcludingVAT>
  <CDVResults>
    <Result>
      <AccountNumber>1222222221</AccountNumber>
      <BranchCode>632009</BranchCode>
      <CustomerCode>EXA9292</CustomerCode>
      <Reference>Example Reference</Reference>
      <Result>Valid</Result>
    </Result>
  </CDVResults>
</Response>

```

Call-back Example

```
<Response>
  <Result>OK</Result>
  <BatchCode>30104</BatchCode>
  <PaymentResults>
    <Result>
      <AccountNumber>123456789</AccountNumber>
      <BranchCode>632005</BranchCode>
      <FirstName>Example</FirstName>
      <Surname>Callback</Surname>
      <Reference>SALARY</Reference>
      <CustomerCode>000001</CustomerCode>
      <Result>Rejected</Result>
      <ResultMessage>ACCOUNT CLOSED</ResultMessage>
      <ResponseCode>12</ResponseCode>
    </Result>
  </PaymentResults>
</Response>
```