Mintroute Systems Voucher API Developer Guide

RELEASE

Version 1.3.1

CONTENTS

Version Control

Version	Release Date	Updates	Section
1.0	30 Aug 2021	Production Release	All
1.1	6 Oct 2021	Password removal from the API request	3
1.2	25 Oct 2021	Added examples of Authentication Parameters, String to Sign & Authentication Headers	1.1.2, 1.1.3, 1.1.5
1.3	6 Dec 2021	Add terminal_id to the table to match the example	3.1 a)
1.3.1	8 Mar 2022	Voucher cancellation API	2.3, 3.2

1 Mintroute System Overview

The Mintroute System is a JSON based API service that enables you to connect to Mintroute's eVouchers inventory with the purpose of enriching your current eVoucher distribution systems, eCommerce applications or to build completely new applications. The data is provided in such a way that you can maintain your branding conventions and not compromise your end users' experience.

1.1 Authentication

All communications with the Mintroute Systems are encrypted with an external layer of encryption "SSL".

API requests are carried out in three simple steps as given below:

1) Construction of request body JSON according to the consuming API service.
2) Generate signature using the secret key. (*Provided by Mintroute*)
3) Sending request body JSON with required Authentication Headers.

Detail description of authentication parameters, signature generation & authentication header is given below:

1.1.1 Vendor Keys

Once you are enrolled in the Mintroute Systems Program you will be provided with a unique Vendor Secret and Access keys for both a sandbox and production environments. The production environment keys will be provided upon successful completion of your sandbox integration, note that the keys are not interchangeable. Please keep your keys in a safe and secure place and never share them with anyone.

- Access Key
- Secret Key

1.1.2 Authentication Parameters

This section will describe authentication parameters required for creating a string-to-sign step by step:

- ***Step 1*** *- HTTP Verb:*
  String to sign starts with common HTTP verbs for web services as given below.

  Example:

```
POST / GET / PUT / DELETE / PATCH

String to Sign:
POSTusername=testvendor&data%5Bean%5D=PSUK10&data%5Blocation%
5D=UAE&data%5Bterminal_id%5D=T1259&data%5Border_id%5D=TRX9281
7312&data%5Brequest_type%5D=purchase&data%5Bresponse_type%5D=
short20211025T1118
```

- ***Step 2*** *- Url-encoded Request Data:*
  Convert JSON into an associative array (arrays have strings as keys) and then generate a URL-encoded string by performing RFC 1738 encoding on that associative array. This URL-encoded string will be concatenated with HTTP verb, and this will be the second part of string-to-sign.

  Example:

```
                    Original JSON data:

{"username":"testvendor","data":{"ean":"PSUK10","location":"U
AE","terminal_id":"T1259","order_id":"TRX92817312","request_t
          ype":"purchase","response_type":"short"}}

                Associative array-based URL:

username=testvendor&data[ean]=PSUK10&data[location]=UAE&data[
terminal_id]=T1259&data[order_id]=TRX92817312&data[request_ty
            pe]=purchase&data[response_type]=short

                    URL encoded data:

username=testvendor&data%5Bean%5D=PSUK10&data%5Blocation%5D=U
AE&data%5Bterminal_id%5D=T1259&data%5Border_id%5D=TRX92817312
&data%5Brequest_type%5D=purchase&data%5Bresponse_type%5D=shor
                                t
                    String to Sign:

POSTusername=testvendor&data%5Bean%5D=PSUK10&data%5Blocation%
5D=UAE&data%5Bterminal_id%5D=T1259&data%5Border_id%5D=TRX9281
7312&data%5Brequest_type%5D=purchase&data%5Bresponse_type%5D=
                    short20211025T1118
```

Important Note:

- Here associative array opening brackets **[** are encoded to **%5B**, closing backets **]** are encoded to **%5D** and spaces will be encoded to **+** character. Any key or value of associative array having special characters must be encoded with RFC 1738.

- URL-encoding can be checked on RFC 1738 encoding external tools like https://www.freeformatter.com/url-encoder.html.

- Boolean value of parameter must be encoded into 0 against false and 1 against true.

- ***Step 3*** *- Signature Time Stamp:*
  Coordinated Universal Time (UTC) time stamp represented in **ISO 8601** format, without ***seconds and milli seconds*****.** This time stamp sting will be concatenated with URL-encoded data, and this will be the third part of sting-to-sign.

  Example:

```
                YYYYMMDDThh:mm for 20211025T1118

                        String to Sign:

POSTusername=testvendor&data%5Bean%5D=PSUK10&data%5Blocation%
5D=UAE&data%5Bterminal_id%5D=T1259&data%5Border_id%5D=TRX9281
7312&data%5Brequest_type%5D=purchase&data%5Bresponse_type%5D=
                        short20211025T1118
```

1.1.3 String to Sign

Above mentioned authentication parameters are concatenated to form a “string to sign”, which will be passed into HMAC to generate a valid signature. Generation of the signature is further explained below.

STRING TO SIGN:

```
[HTTP_VERB][URL-ENCODED_REQUEST_DATA][SIGNATURE_TIME_STAMP]
```

Example:

```
POSTusername=testvendor&data%5Bean%5D=PSUK10&data%5Blocation%
5D=UAE&data%5Bterminal_id%5D=T1259&data%5Border_id%5D=TRX9281
7312&data%5Brequest_type%5D=purchase&data%5Bresponse_type%5D=
                        short20211025T1118
```

1.1.4 Signature Generation

Signature generation is carried out in following 3 simple steps as given below:

1. Convert “string to sign” to HMAC-SHA256 using SECRET KEY
2. Convert the generated HMAC hash into Raw Binary format.
3. Encode the Binary string with Base 64 encoding.

Pseudo of the *SIGNATURE* is given below:

```
BASE_64_ENCODED(BINARY_FORMAT(HMAC(SHA256,STRING_TO_SIGN,SECR
                        ET_KEY))))
```

NOTE: *Make sure you specify the HMAC parameters in the correct order according to the programming language you are using.*

1.1.5 Authentication Headers

Authentication process requires following HEADERS. The list of HTTP headers and example values are given below:

HTTP HEADERS	VALUES
Accept	application/json
Content-Type	application/json
Authorization	algorithm="hmac-sha256", credential="M1AKAH142/20200106", signature="8og48uRK5UpBd74/vOD651hXFDt3TGnDsBdIZ1k6Iws="
X-Mint-Date	20200106T072623Z

- ***Accept*:**

  The Accept request HTTP header advertises which content types, expressed as MIME types of the client can understand. Mconnect response will always be json format, so the user should be able to accept application/json response.

- ***Content-Type:***

  A Content-Type header tells the client what content type of the returned content is. For Mconnect the content type must be application/json.

- ***Authorization:***

Authorization Header consists of three parts, given below:

1. **Algorithm:** The default value for this parameter is "hmac-sha256"
2. **Credential:** Credentials contains two parameters given below:

    - **Access Key** e.g., M1AKAH142
    - **Date** e.g., "20200106"

   Final credential will be Access Key, slash (/) & Date e.g., "M1AKAH142/20200106"
3. **Signature:** Generated Signature.

So final **Authorization Header** will be a sting given below:

```
algorithm="hmac-
sha256",credential="M1AKAH142/20200106",signature="8og48uRK5U
pBd74/vOD651hXFDt3TGnDsBdIZ1k6Iws="
```

Important Note:

Make sure that you pass **Authorization Header** as mentioned and it should not be replaced by other default authorization header generated by your server

- ***X-Mint-Date:***

X-Mint-Date should be a time stamp using a *DateTime* object e.g (hours, minutes, and seconds). This is a fixed-length subset of the format defined by *ISO 8601*, represented in Coordinated Universal Time (UTC).
Example:
YYYYMMDDThhmmssZ (where T and Z are literals & hhmmss is in 24 hours format).
20200106T173600Z is a valid time stamp.

NOTE: *Do not include milliseconds in the time stamp.*

The *date part* (YYYYMMDD) of request header timestamp *(X-Mint-Date)* must match the *date part* (credential="M1AKAH142/20200106") of *Authorization Header*.
API Server will check the *X-Mint-Date* header parameter for a time stamp, and it will compare that with *date part of Authorization Header* for an eight-digit string representing year *(YYYY)*, month *(MM)*, and day *(DD)* of the request.

For example, if the *X-Mint-Date* header value is *20111015T080000Z* and the date part of the *Authorization Header* is *20111015*, Mconnect will allow the authentication process to go through.

1.1.6 Reference Code Examples

Following are some of the code snippets in different languages for signature generation.
Please note that these examples are just for reference. Please do not copy paste them in your production code.

PHP Example:

```
$signature = base64_encode(hash_hmac('sha256',
$string_to_sign, secret_key, true));
```

JavaScript Example:

```
var hash = CryptoJS.HmacSHA256(string_to_sign, secret_key);
var base64 = hash.toString(CryptoJS.enc.Base64);
```

C # Example:

```
var encoding = new System.Text.ASCIIEncoding();
byte[] keyByte = encoding.GetBytes(secret_key);
byte[] messageBytes = encoding.GetBytes(string_to_sign);
using (var hmacsha256 = new HMACSHA256(keyByte)){
byte[] hashmessage = hmacsha256.ComputeHash(messageBytes);
return Convert.ToBase64String(hashmessage);
}
```

Ruby Example:

```
hash  = OpenSSL::HMAC.digest('sha256', " secret_key ", "
string_to_sign")
puts Base64.encode64(hash)
```

Python Example:

```
signature = base64.b64encode(hmac.new(secret_key,
string_to_sign, digestmod=hashlib.sha256).digest())
```

Perl Example:

```
use Digest::SHA qw(hmac_sha256_base64);
$digest = hmac_sha256_base64("sting_to_sign ", " secret_key
")
```

Java Example:

```
Mac sha256_HMAC = Mac.getInstance("HmacSHA256");
SecretKeySpec secret_key = new
SecretKeySpec(secret_key.getBytes(), "HmacSHA256");
sha256_HMAC.init(secret_key);
String hash =
Base64.encodeBase64String(sha256_HMAC.doFinal(sting_to_sign.g
etBytes()));
```

1.1.7 Summary of API Request Generation

<table>
<tr><td>1. String to Sign</td><td>[HTTP_VERB][URL-ENCODED_REQUEST_DATA][SIGNATURE_TIME_STAMP]</td></tr>
<tr><td>2. Create Signature</td><td>1. Convert “string to sign” to HMAC-SHA256 using SECRET KEY
2. Convert the generated HMAC hash into Raw Binary format.
3. Encode the Binary string with Base 64 encoding.</td></tr>
<tr><td>3. Set Request Headers</td><td>
<table>
<tr><th>HTTP HEADERS</th><th>VALUES</th></tr>
<tr><td>Accept</td><td>application/json</td></tr>
<tr><td>Content-Type</td><td>application/json</td></tr>
<tr><td>Authorization</td><td>algorithm="hmac-sha256”, credential="M1AKAH142/20200106”, signature="8og48uRK5UpBd74/vOD651hXFDt3TGnDsBdIZ1k6Iws="</td></tr>
<tr><td>X-Mint-Date</td><td>20200106T072623Z</td></tr>
</table>
</td></tr>
<tr><td>4. Send Request to API Endpoint</td><td>Send request to API Endpoint with JSON Data (as required by the consuming api service) along with generated HTTP Headers.</td></tr>
</table>

2 Authentication and Security

2.1 Vendor User Authorization

Mintroute Systems require user identification for transaction level Authentication, Authorization and Accounting. As an option each user of the Vendor’s System that will

integrate onto the Mintroute Systems can be issued a unique username and password with a defined User Role.

You will be requested by your Mintroute account manager to fill out an on-boarding form which will include your Users information and their designated roles before going live on production.

2.2 API Requests Format

Requests should be made using the **POST method** unless otherwise stated for the given service.

Response data is provided in JSON format. All data is encoded using the UTF-8 character set. Requesting an unknown service will return a '404 Not Found' response. For known API requests the Mintroute Systems returns a detailed error code and error description to help in handling API errors on the Vendor's System. Please ensure to capture the error message in the response.

2.3 API Functions

The API functions are used to initiate an order and obtain information about an existing order. The data passing through these functions are sensitive and therefore will be authenticated using the signature generated with Vendor Secret Key "Vendor Pre-Shared Key".

Below is a list and description of the API functions for endpoint (https://sandbox.mintroute.com/voucher/v2/api/):

API Request	Description
https://sandbox.mintroute.com/voucher/v2/api/voucher	Request single eVoucher from Mintroute Systems. As an option the Vendor can: 1) Pass a Vendor specific Order ID to simplify the tracking of the Vouchers ordered through Mintroute. The same optional Vendor Order ID can be used in "api/order_details" request to retrieve details on a specific eVoucher 2) Pass a Location ID such as the Vendor's PoS ID or Store Location to help the Vendor track the sales activities by location 3) Reserve an eVoucher for 5 minutes prior to purchasing the eVoucher. This feature can be used to verify that the eVoucher is available prior to collecting money from the Customer
https://sandbox.mintroute.com/voucher/v2/api/cancel	Request to cancel a purchased eVoucher (Applicable only if the Issuer allows for cancelation)

Below is a list and description of the API functions for endpoint (https://sandbox.mintroute.com/vendor/api/):

API Request	Description
https://sandbox.mintroute.com/vendor/api/stock	Request to verify if certain denomination is available in stock prior to purchasing the eVoucher
https://sandbox.mintroute.com/vendor/api/order_details	Request order details for a specific order. The Vendor should pass the optional Vendor Order ID attribute in the "api/voucher" request to use this feature
https://sandbox.mintroute.com/vendor/api/get_all_orders	Request all order details by Vendor. This function also allows for selecting order details within a specified time period
https://sandbox.mintroute.com/vendor/api/get_current_balance	Request Vendor's available balance. This function can be used to verify the if the Vendor has enough funds with Mintroute prior to accepting the payment from the Customer. Additionally, Vendors can request balance of their non-USD wallets by passing the "currency" parameter in the request
https://sandbox.mintroute.com/vendor/api/brand	Request for all the brand available to vendor based on passed category ID
https://sandbox.mintroute.com/vendor/api/denomination	Request for all the denomination available to vendor based on passed brand ID

Important Note: In order to ensure the success of the transactions we recommend setting API request timeouts to 60 secs.

2.4 URL Endpoints

The URL endpoint is created by combining the base URL of the environment in which you want to execute your request; Sandbox vs. Production, the appropriate security tokens for that environment, and the function you want to execute along with the posted parameters as shown in the example below.

For example, if you want to purchase voucher in the Sandbox environment, you will specify the following URL when executing an HTTPs POST:

https://sandbox.mintroute.com/

Note: The posted string of the API request should be URL Encoded. This will ensure proper handling of "/" & "+" on Mintroute Systems side.

3 API Requests

The set of Parameters, which varies from one Request to another, need to be converted to JSON and then a signature should be calculated based on HMAC specification. The results will be returned as an array of Parameters in JSON format.

Product Setup:

Our product EAN do not change as we expand our product portfolio. As we add new products Mintroute's account management team will share with you the new EAN of the products before activating the products in your account.

As part of the on-boarding process we will share with you an initial list of denomination with EAN's that are configured in your sandbox account.

Important Note: In order to ensure the success of the transactions we recommend setting API request timeouts to 60 secs.

3.1 api/v2/voucher

a) eVoucher Reservation:

This API request can be used to reserve an eVoucher for 5 minutes prior to purchasing the eVoucher. This feature can be used to verify that the eVoucher is available prior to collecting money from the Customer at the PoS. eVoucher reservation is only possible for fixed denomination vouchers.

*** Our fair usage policy prohibits Vendors from reserving more than 5 Vouchers within a 5 minutes window. Exceeding that limit will result in failed reservation requests and potentially blockage of the Vendor account.**

Required Parameters:

Parameter	Type	Required	Description
username	String	Yes	Unique Username provided to each employee or PoS of the Vendor
data	Object	Yes	Data Object
ean	String	Yes	Unique denomination identifier of the eVoucher you would like to order from the Mintroute Systems
terminal_id	String	Yes	terminal ID of the platform from where the request has been generated
request_type	String	Yes	If set as reserve, the eVoucher will be reserved for 3 minutes to allow the Vendor to verify the availability of the eVoucher before collecting money of the Customer

Request Example:

For voucher reservation.

```
{"username":"testvendor","data":{"ean":"PSUK10","terminal_id"
:"TRS4123","request_type":"reserve"}}
```

Returned Parameters:

L	Parameter	Type	Description
1	status	Boolean	Will return “true” if reserved successfully
1	message	String	Will return “Vouchers Reserved Successfully” if reserved successfully
1	data	Object	Data object
2	reservation_id	String	Reserved ID of the reserved voucher
2	brand_name	String	Brand name of the reserved voucher
2	denomination_name	String	Denomination name of the reserved voucher

Response Examples:

JSON Format

```
{
     "status": true,
     "message": "Vouchers reserved successfully",
     "data": {
          "reservation_id": "1233",
          "brand_name": "iTunes",
          "denomination_name": "$10"
     }
}
```

If an error occurs during reservation process

```
{"status":false,"error":"You do not have a reservation quota
or your available reservation slots are already full. Contact
administrator","error_code":"1056"}

OR

{"status":false,"error":"No contract defined between vendor
and MConnect for selected denomination. Contact
administrator","error_code": "1032"}
```

b) eVoucher Purchase:

The following Parameters are required when purchasing an eVoucher on the Mintroute Systems, As an option the Vendor can:

1) Pass a Vendor specific Order ID to simplify the tracking of the Vouchers ordered through Mintroute. The same optional Vendor Order ID can be used in "api/order_details" request to retrieve details on a specific eVoucher
2) Pass a Location ID such as the Vendor's PoS ID or Store Location to help the Vendor track the sales activities by location.

*** Our fair usage policy prohibits Vendors from purchasing more than 10 Vouchers in a 1-minute window. Exceeding that limit will result in rejecting the request or blockage of the Vendor account.**

Required Parameters:

Parameter	Type	Required	Description
username	String	Yes	Unique Username provided to each employee or PoS of the Vendor
data	Object	Yes	Data Object
ean	String	Yes	Unique denomination identifier of the eVoucher you would like to order from the Mintroute Systems
location	String	No	Location ID such as the Vendor's PoS ID or Store Location to help the Vendor track the sales activities by location
order_id	String	No	Unique Vendor generated Order ID
terminal_id	String	Yes	terminal ID of the platform from where the request has been generated
request_type	String	Yes	If set as purchase, the eVoucher will be purchased directly without reserving.
response_type	String	No	If set as short response will be shorter, if set as long the response will be longer having more details of brand and denomination etc. if not passed the response will be set to long by default.

* Mintroute Systems at the moment only allows for ordering one eVoucher at a time

Request Example:

For voucher purchase.


```
{"username":"testvendor","data":{"ean":"PSUK10","location":"UAE","terminal_id":"T1259","order_id":"TRX92817312","request_type":"purchase","response_type":"short"}}
```

```
OR

{"username":"testvendor","data":{"ean":"PSUK10","location":"U
AE","terminal_id":"T1259","order_id":"TRX92817312","request_t
ype":"purchase","response_type":"long"}}
```

Returned Parameters: (short)

L	Parameter	Type	Description
1	status	Boolean	Will return “true” if purchased successfully
1	message	String	Will return “Vouchers purchased successfully” if purchased successfully
1	data	Object	Data object
2	voucher	Object	Voucher object
3	voucher_value	String	Value of the purchased voucher
3	voucher_currency	String	Currency of the purchased voucher
3	Pincode	String	Purchased voucher pin code
3	serial_number	String	Serial number of the purchased voucher
3	barcode	String	Barcode of the denomination

Response Examples: (short)

JSON Format

```
{
     "status": true,
     "message": "Vouchers purchased successfully",
     "data": {
          "voucher": {
               "voucher_value": "10",
               "voucher_currency": "USD",
               "pincode": "76354GF18X554547",
               "serial_number": "148706473258a2ce9cd5fe6",
               "barcode": "0000016291591"
          }
     }
}
```

Returned Parameters: (long)

L	Parameter	Type	Description
1	status	Boolean	Will return “true” if purchased successfully

1	message	String	Will return “Vouchers purchased successfully” if purchased successfully
1	data	Object	Data object
2	category_id	String	Category ID of the purchased voucher
2	category_name	String	Category name of the purchased voucher
2	category_description	String	Category description
2	category_imagepath	String	Image path of the category
2	category_imagename	String	Image name of the category
2	brand_id	String	Brand ID of the purchased voucher
2	brand_name	String	Brand name of the purchased voucher
2	receipt_title1	String	title 1 that prints on the receipt
2	receipt_title2	String	title 2 that prints on the receipt
2	brand_description	String	Brand description
2	brand_imagepath	String	Image path of the brand
2	brand_icon	String	Icon of the brand
2	brand_imagename	String	Image name of the brand
2	denomination_id	String	Denomination ID of the purchased voucher
2	denomination_name	String	Denomination name of the purchased voucher
2	denomination_value	String	Denomination value of the purchased voucher
2	denomination_currency	String	Denomination currency of the purchased voucher
2	denomination_description	String	Denomination description
2	denomination_imagepath	String	Image path of the denomination
2	denomination_imagename	String	Image name of the denomination
2	order_date	String	Date of the order placed
2	order_id	String	Order ID passed by the vendor on the purchase request
2	location	String	Location passed by the vendor on the purchase request
2	vendor_pre_balance	String	Vendor balance before transaction
2	vendor_post_balance	String	Vendor balance after transaction
2	voucher	String	Voucher object
3	pincode	String	Purchased voucher pin code
3	serial_number	String	Serial number of the purchased voucher
3	barcode	String	Barcode of the denomination

Response Examples: (long)

JSON Format

```
{
     "status": true,
     "message": "Vouchers purchased successfully",
     "data": {
          "category_id": "2",
          "category_name": "General",
          "category_description": "A wide selection of
popular prepaid gift cards, such as: iTunes, Google Play,
Amazon, and may others.",
          "category_imagepath":
"https://mconnect.mintroute.com/assets/uploads/categoryimages
",
          "category_imagename": "Untitled.png",
          "brand_id": "2",
          "brand_name": "iTunes",
          "receipt_title1": "First Title",
          "receipt_title2": "Second Title",
          "brand_description": "Codes redeemable only on US
iTunes Store.",
          "brand_imagepath":
"https://sandbox.mintroute.com/assets/uploads/brandimages",
          "brand_icon": "itunes5.png",
          "brand_imagename": "itunes4.png",
          "denomination_id": "3",
          "denomination_name": "$10",
          "denomination_value": "10",
          "denomination_currency": "USD",
          "denomination_description": "How to redeem the
iTunes Gift voucher on computer:\r\n1.\tIn the quick link
window of your iTunes program, click Redeem.\r\n2.\tEnter the
16-digit code you have purchased.",
          "denomination_imagepath":
"https://dl.dropboxusercontent.com",
          "denomination_imagename": "itunes.jpg",
          "order_date": "2021-08-02 12:40:39",
          "order_id": "TRX92817312",
          "location": "Dubai",
          "vendor_pre_balance": "45806.810",
          "vendor_post_balance": "45797.81",
          "voucher": {
               "pincode": "76354GF18X254547",
               "serial_number": "148706473158a2ce9b710bb",
               "barcode": "0000016291591"
          }
     }
}
```


Some excepted errors during the process

```
{"status":false,"error":"Duplicate order ID
provided","error_code":"1033"}

OR

{"status":false,"error":"No contract defined between vendor
and MConnect for selected denomination. Contact
administrator","error_code": "1032"}
```

3.2 api/v2/cancel

This API request can be used to cancel a purchased eVoucher, **not all eVouchers can be cancelled as cancellation of eVoucher highly depends on the product, merchant (service provider) and eVoucher redemption status.**

For Example, eVoucher from merchants like Paysafe, BIGO Live, BIGO Likee, Yalla Live, Tango Me, LivU, and Yalla Ludo allow cancellation of vouchers but other merchants like SONY and Apple Card do not allow for eVoucher cancellation. Therefore, the message should be configurable for each product type.

Note: eVoucher that has been already redeemed (consumed) can not be cancelled.

Below is the list of parameters required by the end point for cancelling a voucher.

Required Parameters:

Parameter	Type	Required	Description
username	String	Yes	Unique Username provided to each employee or PoS of the Vendor
data	Object	Yes	Data Object
serial_number	String	Yes	Serial number of the purchased voucher

Request Example:

For voucher cancellation.

```
{"username":"testvendor","data":{"serial_number":"SEZGIT10110
808"}}
```

Returned Parameters:

L	Parameter	Type	Description
1	status	Boolean	Will return “true” if voucher cancelled successfully
1	message	String	Will return “Voucher cancelled successfully” if cancelled successfully
1	data	Object	Data object
2	voucher	Object	Voucher object
3	brand_name	String	Brand name of the cancelled voucher
3	denomination_name	String	Denomination name of the cancelled voucher
3	voucher_value	String	Value of the cancelled voucher
3	voucher_currency	String	Currency of the cancelled voucher
3	serial_number	String	Serial number of the cancelled voucher

Response Examples:

JSON Format

```
{
     "status": true,
     "message": "Voucher cancelled successfully",
     "data": {
          "voucher": {
               "brand_name": "Tango",
               "denomination_name": " USD 1 (INT)",
               "voucher_value": "1",
               "voucher_currency": "USD",
               "serial_number": "SEZGIT10110808"
          }
     }
}
```

If an error occurs during cancellation **process**

```
{"status":false,"error":"Unable to cancel the
voucher","error_code":"1410"}

OR

{"status":false,"error":"Invalid serial number
provided","error_code": "1411"}
```

3.3 api/stock

Below is the list of parameters required by the end point for checking stock availability for a product.

Required Parameters:

Parameter	Type	Required	Description
username	String	Yes	Unique Username provided to the Vendor
data	Object	Yes	Object containing unique EAN of the product you would like to enquire about its availability in the Mintroute Systems
ean	String	Yes	Unique Denomination identifier for which product availability is requested, this parameter should be put in the data object

Request Example:

For checking product stock.

```
{"username":"testvendor","data":{"ean":"PSUK10"}}
```

Returned Parameters:

L	Parameter	Type	Description
1	Status	Boolean	Will return “true” if stock available and “false” if out of stock
1	Message	String	Will return “Quantity is available” if stock available and “Quantity not available” if out of stock

Response Examples:

JSON Format

```
{"status":true,"message":"Quantity is available"}
```

Some excepted errors during the process

```
{"status":false,"error":"Quantity is not
available","error_code":"1300"}}

OR

{"status": false,"error": "Invalid denomination
identifier","error_code": "1166"}

OR

{"status": false,"error": "Unable to find denomination
identifier","error_code": "1154"}
```

3.4 api/order_details

Below is the list of parameters required by the end point for getting a specific order detail.

Required Parameters:

Parameter	Type	Required	Description
username	String	Yes	Unique Username provided to each employee or PoS of the Vendor
data	Object	Yes	Data Object
order_id	String	Yes	Unique Vendor generated Order ID of the denomination you would like to enquire the Mintroute Systems on. It is important that the Vendor have filled the optional attribute "order_id" with a unique Vendor Order ID when requesting an eVoucher using the api/voucher call
response_type	String	No	If set as short response will be shorter, if set as long the response will be longer having more details of brand and denomination etc. if not passed the response will be set to long by default.

Request Example:

For getting order detail

```
{"username":"testvendor","data":{"order_id":"VID54321","respo
nse_type":"short"}}
```

Returned Parameters: (short)

L	Parameter	Type	Description
1	Status	Boolean	Will return “true” if order details fetched successfully
1	Message	String	Will return “Order details fetched successfully” if order exists
1	data	Object	Data object
2	brand_name	String	Brand name of voucher
2	denomination_name	String	Denomination name of voucher
2	denomination_value	String	Denomination value of voucher
2	denomination_currency	String	Denomination currency of the voucher
2	order_date	String	Date of the order placed
2	order_id	String	Order ID passed by the vendor on the purchase request
2	Location	String	Location passed by the vendor on the purchase request
2	vendor_pre_balance	String	Vendor balance before transaction
2	vendor_post_balance	String	Vendor balance after transaction
2	contract_price	String	Vendor contract price of the denomination
2	order_type	String	Type of order it will be either topup or voucher
2	voucher	String	Voucher object
3	pincode	String	Purchased voucher pin code
3	serial_number	String	Serial number of the purchased voucher
3	barcode	String	Barcode of the denomination

Response Examples: (short)

JSON Format


```
{
     "status": true,
     " message": "Order details fetched successfully"
     "data": {
          "brand_name": "iTunes",
          "denomination_name": "$10",
          "denomination_value": "10",
          "denomination_currency": "USD",
          "order_date": "2021-08-02 14:16:45",
          "order_id": "TRX92817313",
          "location": "UAE",
          "vendor_pre_balance": "45746.810",
          "vendor_post_balance": "45737.81",
          "contract_price": "9",
          "order_type": "voucher"
          "voucher": {
               "pincode": "76354GF18X074547",
               "serial_number": "148706474058a2cea4ee9e2",
               "barcode": "0000016291591"
          },

     }
}
```


Returned Parameters: (long)

L	Parameter	Type	Description
1	status	Boolean	Will return “true” if order details fetched successfully
1	message	String	Will return “Order details fetched successfully” if order exists
1	data	Object	Data object
2	category_id	String	Category ID of the voucher
2	category_name	String	Category name of the voucher
2	brand_id	String	Brand ID of the voucher
2	brand_name	String	Brand name of the voucher

2	receipt_title1	String	title 1 that prints on the receipt
2	receipt_title2	String	title 2 that prints on the receipt
2	brand_description	String	Brand description
2	brand_imagepath	String	Image path of the brand
2	brand_icon	String	Icon of the brand
2	brand_imagename	String	Image name of the brand
2	denomination_id	String	Denomination ID of the voucher
2	denomination_name	String	Denomination name of the voucher
2	denomination_value	String	Denomination value of the voucher
2	denomination_currency	String	Denomination currency of the voucher
2	denomination_description	String	Denomination description
2	denomination_imagepath	String	Image path of the denomination
2	denomination_imagename	String	Image name of the denomination
2	order_date	String	Date of the order placed
2	order_id	String	Order ID passed by the vendor on the purchase request
2	location	String	Location passed by the vendor on the purchase request
2	vendor_pre_balance	String	Vendor balance before transaction
2	vendor_post_balance	String	Vendor balance after transaction
2	contract_price	String	Vendor contract price of the denomination

2	order_type	String	Type of order it will be either topup or voucher
2	voucher	String	Voucher object
3	pincode	String	Purchased voucher pin code
3	serial_number	String	Serial number of the purchased voucher
3	Barcode	String	Barcode of the denomination

Response Examples: (long)

JSON Format


```
{
     "status": true,
     " message": "Order details fetched successfully",
     "data": {
          "category_id": "2",
          "category_name": "General",
          "brand_id": "2",
          "brand_name": "iTunes",
          "receipt_title1": "First Title",
          "receipt_title2": "Second Title",
          "brand_description": "Codes redeemable only on US
iTunes Store.",
          "brand_imagepath":
"https://sandbox.mintroute.com/assets/uploads/brandimages",
          "brand_icon": "itunes5.png",
          "brand_imagename": "itunes4.png",
          "denomination_id": "3",
          "denomination_name": "$10",
          "denomination_value": "10",
          "denomination_currency": "USD",
          "denomination_description": "How to redeem the
iTunes Gift voucher on computer:\r\n1.\tIn the quick link
window of your iTunes program, click Redeem.\r\n2.\tEnter
the 16-digit code you have purchased.",
          "denomination_imagepath":
"https://dl.dropboxusercontent.com",
          "denomination_imagename": "itunes.jpg",
          "order_date": "2021-08-02 14:16:45",
          "order_id": "TRX92817313",
          "location": "UAE",
          "vendor_pre_balance": "45746.810",
          "vendor_post_balance": "45737.81",
```

```
          "contract_price": "9",
          "order_type": "voucher"
          "vouchers": {
               "pincode": "76354GF18X074547",
               "serial_number": "148706474058a2cea4ee9e2",
               "barcode": "0000016291591"
          },

     }
}
```


Some excepted errors during the process


```
{"status": false,"error": "Invalid order ID or no such order
exists in the system","error_code": "1073"}

OR

{"status": false,"error": "Unable to find order
ID","error_code": "1181"}

OR

{"status": false,"error": "Unable to fetch order
detail","error_code": "1180"}
```

3.5 api/get_all_orders

Below is the list of parameters required by the end point for fetching order list based on duration.

Required Parameters:

Parameter	Type	Required	Description
username	String	Yes	Unique Username provided to each employee or PoS of the Vendor
data	Object	Yes	Data Object
datefrom	String	Yes	Start Order Date presented in this format 2014-09-20 10:42:52
dateto	String	Yes	End Order Date presented in this format 2014-09-20 10:42:55
page	Integer	No	List or orders are divided in pages, page number required for browsing pages, if page was not passed it will show order list of page 1
order_type	String	Yes	Type of order you would like to enquire e.g., voucher or topup
response_type	String	No	If set as short response will be shorter, if set as long the response will be longer having more details of brand and denomination etc. if not passed the response will be set to long by default.

Request Example:

For getting orders based on duration.

```
{"username":"testvendor","data":{"datefrom":"2015-11-01
00:00:00","dateto":"2015-11-01
23:59:59","page":1,"order_type":"voucher","response_type":"sh
ort"}}
```

Returned Parameters: (short)

L	Parameter	Type	Description
1	status	Boolean	Will return “true” if orders fetched successfully
1	message	String	Will return “Orders fetched successfully” if orders are fetched
1	page	Integer	Shows the page number of the order list
1	total_orders	Integer	Total number of orders in the selected duration
1	total_pages	Integer	Total number of pages that you can browse through
1	data	Object	Data object

2	order_number	String	System generated order number
3	brand_name	String	Brand name of the voucher
3	denomination_name	String	Denomination name of the voucher
3	denomination_value	String	Denomination value of the voucher
3	denomination_currency	String	Denomination currency of the voucher
3	order_quantity	String	Quantity of the order
3	order_date	String	Date of the order placed
3	order_id	String	Order ID passed by the vendor on the purchase request
3	location	String	Location passed by the vendor on the purchase request
3	voucher	String	Voucher object
4	pincode	String	Purchased voucher pin code
4	serial_number	String	Serial number of the voucher
4	barcode	String	Barcode of the denomination

Response Examples: (short)

JSON Format


```
{
     "status": true,
     " message": "Orders fetched successfully",
     "page":1,
     "total_orders":3,
     "total_pages":1,
     "data": {
          "10905": {
               "brand_name": "iTunes",
               "denomination_name": "$10",
               "denomination_value": "$10.00",
               "denomination_currency": "USD",
               "order_quantity": "1",
               "order_date": "2021-08-03 10:38:39",
               "order_id": "TRX92817613",
               "location": "UAE",
               "contract_price": "9.99",
               "order_type": "voucher",
               "voucher": {
                    "pincode": "76354GF18X674547",
                    "serial_number":
"148706474358a2cea799785",
                    "barcode": "0000016291591"
               }
          },
          "10904": {
               "brand_name": "Paysafecard UAE",
```

```
                "denomination_name": "AED 100.00",
                "denomination_value": "$100.00",
                "denomination_currency": "USD",
                "order_quantity": "1",
                "order_date": "2021-08-03 10:32:53",
                "order_id": "TRX92817513",
                "location": "UAE",
                "contract_price": "99.99",
                "order_type": "voucher",
                "voucher": {
                     "pincode": "76354GF18X450384",
                     "serial_number": "SFB501014023",
                     "barcode": "0000016291591"
                }
           },
           "10903": {
                "brand_name": "Paysafecard UAE",
                "denomination_name": "AED 100.00",
                "denomination_value": "$100.00",
                "denomination_currency": "USD",
                "order_quantity": "1",
                "order_date": "2021-08-03 10:31:47",
                "order_id": "TRX92817413",
                "location": "UAE",
                "contract_price": "99.99",
                "order_type": "voucher",
                "voucher": {
                     "pincode": "76354GF18X350384",
                     "serial_number": "SFB501014022",
                     "barcode": "0000016291591"
                }
           }
     }
}
```


Returned Parameters: (long)

L	Parameter	Type	Description
1	status	Boolean	Will return “true” if orders fetched successfully
1	message	String	Will return “Orders fetched successfully” if orders are fetched
1	page	Integer	Shows the page number of the order list
1	total_orders	Integer	Total number of orders in the selected duration
1	total_pages	Integer	Total number of pages that you can browse through
1	data	Object	Data object
2	order_number	String	System generated order number
3	category_id	String	Category ID of the voucher

3	category_name	String	Category name of the voucher
3	brand_id	String	Brand ID of the voucher
3	brand_name	String	Brand name of the voucher
3	receipt_title1	String	title 1 that prints on the receipt
3	receipt_title2	String	title 2 that prints on the receipt
3	brand_description	String	Brand description
3	brand_imagepath	String	Image path of the brand
3	brand_icon	String	Icon of the brand
3	brand_imagename	String	Image name of the brand
3	denomination_id	String	Denomination ID of the voucher
3	denomination_name	String	Denomination name of the voucher
3	denomination_value	String	Denomination value of the voucher
3	denomination_currency	String	Denomination currency of the voucher
3	denomination_description	String	Denomination description
3	denomination_imagepath	String	Image path of the denomination
3	denomination_imagename	String	Image name of the denomination
3	order_quantity	String	Quantity of the order
3	order_date	String	Date of the order placed
3	order_id	String	Order ID passed by the vendor on the purchase request
3	location	String	Location passed by the vendor on the purchase request
3	voucher	String	Voucher object
4	pincode	String	Purchased voucher pin code
4	serial_number	String	Serial number of the purchased voucher
4	barcode	String	Barcode of the denomination

Response Examples: (long)

JSON Format


```
{
     "status": true,
     " message": "Orders fetched successfully",
     "page": 1,
     "total_orders": 3,
     "total_pages": 1,
     "data": {
          "10905": {
               "category_id": "2",
               "category_name": "General",
               "brand_id": "2",
```

```
                "brand_name": "iTunes",
                "receipt_title1": "First Title",
                "receipt_title2": "Second Title",
                "brand_description": "Codes redeemable only on
US iTunes Store.",
                "brand_imagepath":
"https://sandbox.mintroute.com/assets/uploads/brandimages",
                "brand_icon": "itunes5.png",
                "brand_imagename": "itunes4.png",
                "denomination_id": "3",
                "denomination_name": "$10",
                "denomination_value": "$10.00",
                "denomination_currency": "USD",
                "denomination_description": "How to redeem the
iTunes Gift voucher on computer:\r\n1.\tIn the quick link
window of your iTunes program, click Redeem.\r\n2.\tEnter the
16-digit code you have purchased.",
                "denomination_imagepath":
"https://dl.dropboxusercontent.com",
                "denomination_imagename": "itunes.jpg",
                "order_quantity": "1",
                "order_date": "2021-08-03 10:38:39",
                "order_id": "TRX92817613",
                "location": "UAE",
                "contract_price": "9.99",
                "order_type": "voucher",
                "voucher": {
                     "pincode": "76354GF18X674547",
                     "serial_number":
"148706474358a2cea799785",
                     "barcode": "0000016291591"
                }

           },
           "10904": {
                "category_id": "2",
                "category_name": "General",
                "brand_id": "64",
                "brand_name": "Paysafecard UAE",
                "receipt_title1": "First Title",
                "receipt_title2": "Second Title",
                "brand_description": "Codes redeemable only on
US.",
                "brand_imagepath":
"https://sandbox.mintroute.com/assets/uploads/brandimages",
                "brand_icon": "paysafecard.png",
                "brand_imagename": "paysafecard.png",
                "denomination_id": "2234",
                "denomination_name": "AED 100.00",
                "denomination_value": "$100.00",
                "denomination_currency": "USD",
```

```
                "denomination_description": "How to redeem the
paysafecard Gift voucher.",
                "denomination_imagepath":
"https://dl.dropboxusercontent.com",
                "denomination_imagename": "paysafecard.jpg",
                "order_quantity": "1",
                "order_date": "2021-08-03 10:32:53",
                "order_id": "TRX92817513",
                "location": "UAE",
                "contract_price": "99.99",
                "order_type": "voucher",
                "voucher": {
                     "pincode": "76354GF18X450384",
                     "serial_number": "SFB501014023",
                     "barcode": "0000016291591"
                }

          },
          "10903": {
                "category_id": "2",
                "category_name": "General",
                "brand_id": "64",
                "brand_name": "Paysafecard UAE",
                "receipt_title1": "First Title",
                "receipt_title2": "Second Title",
                "brand_description": "Codes redeemable only on
US.",
                "brand_imagepath":
"https://sandbox.mintroute.com/assets/uploads/brandimages",
                "brand_icon": "paysafecard.png",
                "brand_imagename": "paysafecard.png",
                "denomination_id": "2234",
                "denomination_name": "AED 100.00",
                "denomination_value": "$100.00",
                "denomination_currency": "USD",
                "denomination_description": "How to redeem the
paysafecard Gift voucher.",
                "denomination_imagepath":
"https://dl.dropboxusercontent.com",
                "denomination_imagename": "paysafecard.jpg",
                "order_quantity": "1",
                "order_date": "2021-08-03 10:31:47",
                "order_id": "TRX92817413",
                "location": "UAE",
                "contract_price": "99.99",
                "order_type": "voucher",
                "voucher": {
                     "pincode": "76354GF18X350384",
                     "serial_number": "SFB501014022",
                     "barcode": "0000016291591"
                }
```

```
            }
      }
}
```

Some excepted errors during the process

```
{"status": false,"error": "Invalid duration start
date","error_code": "1301"}

OR

{"status": false,"error": "Invalid duration end
date","error_code": "1302"}

OR

{"status": false,"error": "Invalid page number","error_code":
"1303"}

OR

{"status": false,"error": "Unable to find order
type","error_code": "1192"}

OR

{"status": false,"error": "Unable to identify the order
type","error_code": "1193"}
```

3.6 api/get_current_balance

Below is the list of parameters required by the end point for checking current balance for a specific currency wallet.

Required Parameters:

Parameter	Type	Required	Description
username	String	Yes	Unique Username provided to each employee or PoS of the Vendor
data	Object	Yes	Data Object
currency	String	Yes	Wallet currency type e.g., USD, AED, SAR etc.

Request Example:

For checking vendor currency balance.

```
{"username":"testvendor","data":{"currency":"USD"}}
```

Returned Parameters:

L	Parameter	Type	Description
1	status	Boolean	Will return “true” if fetched the balance successfully
1	message	String	Will return “Balance fetched successfully” if fetched the balance
1	data	Object	Data object
2	vendor_name	String	Name of the vendor
2	available_balance	String	Current vendor balance
2	currency	String	Currency type
2	credit_limit	String	If there is a credit limit assigned to the vendor will reflect here.

Response Examples:

JSON Format

```
{
     "status": true,
     " message": "Balance fetched successfully",
     "data":{
          "vendor_name": "testvendor",
          "available_balance": "46216.81",
          "currency": "SAR"
     }
}
```

Some excepted errors during the process

```
{"status":false,"error":"Invalid
currency","error_code":"1304"}}

OR

{"status": false,"error": "Unable to fetch vendor
balance","error_code": "1183"}

OR

{"status": false,"error": "Unable to find wallet
currency","error_code": "1182"}
```

3.7 api/brand

Below is the list of parameters required by the end point for getting available vendor brand list.

Required Parameters:

Parameter	Type	Required	Description
username	String	Yes	Unique Username provided to each employee or PoS of the Vendor
data	Object	Yes	Data Object
category_id	String	Yes	Category ID for which you like to get brand list. e.g., 1, 2 etc.

Request Example:

For getting brands list.

```
{"username":"testvendor","data": {"category_id": "2"}}
```

Returned Parameters:

L	Parameter	Type	Description
1	status	Boolean	Will return “true” if fetched the brands successfully
1	message	String	Will return “List of brands fetched successfully” if fetched the brands
1	data	Object	Data object
2	category_id	String	Category ID that was passed on the request

3	brand_id	String	Brand ID of the voucher
3	brand_name	String	Brand name of the voucher
3	brand_description	String	Brand description
3	brand_imagepath	String	Image path of the brand
3	brand_imagename	String	Image name of the brand

Response Examples:

JSON Format


```
{
     "status": true,
     "message": "List of brands fetched successfully",
     "data": {
          "2": [{
                    "brand_id": "2",
                    "brand_name": "iTunes",
                    "brand_description": "Codes redeemable
only on US iTunes Store.",
                    "brand_imagepath":
"https://sandbox.mintroute.com/assets/uploads/brandimages",
                    "brand_imagename": "itunes4.png"
               },
               {
                    "brand_id": "63",
                    "brand_name": "Big Bazaar E-Gift
Vouchers",
                    "brand_description": "Big Bazaar is the
largest chain of hypermarket in India.Big Bazaar gift
vouchers offers a wide range of apparel and accessories, baby
accessories, dress material, suiting and shirting, sarees and
much more and all these at a discount.",
                    "brand_imagepath":
"https://sandbox.mintroute.com/assets/uploads/brandimages",
                    "brand_imagename": "234_microsite.jpg"
               }
          ]
     }
}
```


Some excepted errors during the process


```
{"status": false,"error": "Invalid category ID","error_code":
"1048"}

OR
```

```
{"status": false,"error": "Unable to find category
ID","error_code": "1186"}

OR

{"status": false,"error": "Unable to fetch brands for given
category ID","error_code": "1187"}
```

3.8 api/denomination

Below is the list of parameters required by the end point for getting available vendor denomination list.

Required Parameters:

Parameter	Type	Required	Description
username	String	Yes	Unique Username provided to each employee or PoS of the Vendor
data	Object	Yes	Data Object
brand_id	String	Yes	Brand ID for which you like to get demolition list. e.g., 1, 2, 3 etc.

Request Example:

For getting denomination list.

```
{"username":"testvendor","data":{"brand_id":"2"}}
```

Returned Parameters:

L	Parameter	Type	Description
1	status	Boolean	Will return “true” if fetched the denominations successfully
1	message	String	Will return “List of denomination fetched successfully” if fetched the denominations
1	data	Object	Data object
3	category_id	String	Category ID
3	category_name	String	Category name
3	category_description	String	Category description
3	category_imagepath	String	Image path of the category image
3	category_imagename	String	Category image name

3	receipt_title1	String	title 1 that prints on the receipt
3	receipt_title2	String	title 2 that prints on the receipt
3	brand_id	String	Brand ID
3	brand_name	String	Brand name
3	brand_description	String	Brand description
3	brand_imagepath	String	Image path of the brand
3	brand_imagename	String	Image name of the brand
3	denominations	Array/Object	Array of denominations
3	denomination_id	String	Denomination ID
3	denomination_name	String	Denomination name
3	denomination_description	String	Denomination description
3	denomination_value	String	Denomination value
3	denomination_currency	String	Denomination currency
3	denomination_barcode	String	Barcode of the denomination
3	denomination_contract_price	String	Contract price of the denomination
3	denomination_imagepath	String	Image path of the denomination
3	denomination_imagename	String	Image name of the denomination
3	denomiantion_type	String	Type of denomination, gift card or top up

Response Examples:

JSON Format


```
{
     "status": true,
     " message": "List of denominations fetched
successfully",
     "data": {
          "2": {
               "category_id": "2",
               "category_name": "General",
               "category_description": "A wide selection of
popular prepaid gift cards, such as: iTunes, Google Play,
Amazon, and may others.",
               "category_imagepath":
"https://mconnect.mintroute.com/assets/uploads/categoryimages
",
               "category_imagename": "cat.png",
               "2": {
                    "receipt_title1": "First Title",
```

```
                    "receipt_title2": "Second Title",
                    "brand_id": "2",
                    "category_id": "2",
                    "brand_name": "iTunes",
                    "brand_description": "Codes redeemable
only on US iTunes Store.",
                    "brand_imagepath":
"https://sandbox.mintroute.com/assets/uploads/brandimages",
                    "brand_icon": "itunes5.png",
                    "brand_imagename": "itunes4.png",
                    "denominations": [{
                              "denomination_id": "3",
                              "denomination_name": "$10",
                              "denomination_description":
"How to redeem the iTunes Gift voucher on computer:\r\n1.\tIn
the quick link window of your iTunes program, click
Redeem.\r\n2.\tEnter the 16-digit code you have purchased.",
                              "denomination_value": "10",
                              "denomination_currency": "USD",
                              "denomination_barcode":
"0000016291591",
                              "denomination_contract_price":
"10.00",
                              "denomination_imagepath":
"https://dl.dropboxusercontent.com",
                              "denomination_imagename":
"itunes.jpg",
                              "denomiantion_type":
"gift_card"
                         },
                         {
                              "denomination_id": "130",
                              "denomination_name": "$300",
                              "denomination_description":
"How to redeem the iTunes Gift voucher on computer:\r\n1.\tIn
the quick link window of your iTunes program, click
Redeem.\r\n2.\tEnter the 16-digit code you have purchased.",
                              "denomination_value": "300",
                              "denomination_currency": "USD",
                              "denomination_barcode":
"18191642",
                              "denomination_contract_price":
"300.00",
                              "denomination_imagepath":
"https://mconnect.mintroute.com/assets/uploads/denominationim
ages",
                              "denomination_imagename":
"iTunes10.png",
                              "denomiantion_type":
"gift_card"
                         }
```

```
                        ]
                    }
                }
            }
}
```

Some excepted errors during the process

```
{"status": false,"error": "Invalid brand ID","error_code":
"1042"}

OR

{"status": false,"error": "Unable to fetch denominations for
given brand ID","error_code": "1189"}

OR

{"status": false,"error": "Unable to find brand
ID","error_code": "1188"}
```

10 Error Handling

This section provides a table of Error Codes and the matching Error Message to assist the developers with error handling.

CODE	DESCRIPTION
1031	Unable to find username
1032	No contract defined between vendor and MConnect for selected denomination. Contact administrator
1033	Duplicate order ID provided
1034	No such user exist or user does not belong to a vendor
1035	Incorrect username
1036	Required data is missing
1037	You do not have access to the requested section
1039	You do not have enough balance to complete this purchase

1040	You have tried to exceed the per minute transaction limit, your account has been blocked, contact administrator
1042	Invalid brand ID
1048	Invalid category ID
1049	User is locked, contact administrator to unlock
1050	Input data format was not as required, please provide correct data format
1051	Unable to identify denomination, wrong denomination identifier provided
1052	You do not have a valid contract for selected denomination
1055	Length of the provided location exceeds 255 character limit
1056	You do not have a reservation quota or your available reservation slots are already full. Contact administrator
1057	Unable to create an order, try again or contact administrator
1059	Something went wrong, try again or contact administrator
1060	Something went wrong, problem with balance, try again or contact administrator
1161	Unable to create a transaction, try again or contact administrator
1170	Unexpected issue, problem with placing order, kindly retry
1173	Invalid order ID or no such order exists in the system
1178	Unexpected issue with API merchants, contact administrator
1082	Unable to create order, quantity not available. Please check with support@mintroute.com
1083	Unable to find denomination identifier in your request. Please check your request or contact administrator
1105	Invalid input found
1106	Daily purchase limit is exceeded for the brand
1111	Unable to parse response from merchant
1118	Concurrent issue, problem with placing order, kindly retry
1119	Unable to retrieve code, please contact support@mintroute.com
1129	Re delivery stock status not updated
1132	Length of the provided order ID exceeds 24 character limit
1136	No such vendor exists, or vendor is not active
1139	Request fails authentication due to invalid header format
1140	Invalid date parameter
1141	Signature expired
1149	Request fails authentication
1150	Request contains invalid strings
1153	No such user exists, or user is not active

1154	Unable to find denomination identifier
1161	Unable to find the request type
1162	Unable to identify the request type
1163	Unable to identify the response type
1166	Invalid denomination identifier
1167	Unable to find terminal id
1168	Length of the provided terminal ID exceeds 50 character limit
1169	Transaction not found
1170	Unable to purchase voucher from re-delivery stock. Please contact administrator
1177	Internal system error, please contact techsupport@mintroute.com
1180	Unable to fetch order details
1181	Unable to find order ID
1182	Unable to find wallet currency
1183	Unable to fetch vendor balance
1186	Unable to find category ID
1187	Unable to fetch brands for given category ID
1188	Unable to find brand ID
1189	Unable to fetch denominations for given brand ID
1190	Unable to fetch order list
1191	Invalid page number
1192	Unable to find order type
1193	Unable to identify the order type
1300	Quantity is not available
1301	Invalid duration start date
1302	Invalid duration end date
1304	Invalid currency
1410	Unable to cancel the voucher
1411	Invalid serial number provided