



Mintroute Systems Top Up API Developer Guide

RELEASE

Version 1.5

CONTENTS

1 Mintroute System Overview	4
1.1 Authentication.....	4
2 Authentication and Security.....	11
2.1 Vendor User Authorization.....	11
2.2 API Requests Format	11
2.3 API Functions.....	11
2.4 URL Endpoints.....	11
3 API Requests	12
4 Free Fire Top Up	12
4.1 api/account_validation	12
4.2 api/account_topup.....	14
5 Mobile Legends Top Up	17
5.1 api/account_validation	17
5.2 api/account_topup.....	19
6 Razer Gold Top Up	22
6.1 api/account_validation	22
6.2 api/account_topup.....	24
7 Jio Saavn Top Up	28
7.1 api/account_validation	28
7.2 api/account_topup.....	30
8 Net Dragon (Eudemons Online, Conquer Online & Heroes Evolved) Top Up.....	32
8.1 api/account_initialization.....	32
8.2 api/account_validation	35
8.3 api/account_topup.....	30
9 Error Handling.....	40
10 Handling Timeouts.....	42
11 Top Up Test Accounts.....	42

Version Control

Version	Release Date	Updates	Section
1.1	10 Dec 2020	Production Release	All
1.2	12 Jan 2021	Added support for Vendor Order ID	5.2
1.3	28 Feb 2021	Added information about user wallet limits to Razer and Free Fire	
1.4	31 st March 2021	Jio Saavn Top Up	7
		Top Up Test Accounts	10
1.5	10 st June 2021	Net Dragon Top Up	8
		Top Up Test Accounts	10

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

Mintroute will provide the following two keys to each vendor. 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:

- **HTTP Verb:**
Common HTTP verbs for web services.

Example:

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

- **Url-encoded Request Data:**
Convert JSON request data into an associated array and then generate a URL-encoded string from that associative array.

Example:

Original JSON data:

```
{"username":"testvendor","password":"testvendor","data":{"denomination_id":3,"account_id":"10000251"}}
```

URL encoded data:

```
username=testvendor&password=testvendor&data%5Bdenomination_id%5D=3&data%5Baccount_id%5D=10000251
```

- **Signature Time Stamp:**

Coordinated Universal Time (UTC) time stamp represented in **ISO 8601** format, without **seconds and milli seconds**.

Example:

```
YYYYMMDDThh:mm for 20200106T1253 20201124T1155
```

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&password=testvendor&data%5Bdenomination_id%5D=3&data%5Baccount_id%5D=1000025120201210T0959
```

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

- **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

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

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 encrypted using the Vendor Private Token "Vendor Pre-Shared Key".

Below is a list and description of the API functions:

API Request	Description
api/account_validation	The API request is used to validate user account to which the top up is going to be provided
api/account_topup	The API request is used to top up validated user account

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 top up in the Sandbox environment, you will specify the following URL when executing an HTTPs POST:

https://sandbox.mintroute.com/top_up/api/

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.

Important Note:

Our denomination_id 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 denomination_id 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_ids that are configured in your sandbox account.

Important Note: Your production account denomination_id will vary from the sandbox denomination_id. Be sure to update the denomination_id's in the production system before going live.

4 Free Fire Top Up

Important Note: An hourly user limit of USD 250 and a daily user limit of USD 1,000 applies on each wallet.

4.1 api/account_validation

Below is the list of parameters required by the end point for free fire account validation

Required Parameters:

Parameter	Type	Required	Description
username	String	Yes	Unique Username provided to the Vendor
password	String	Yes	Password provided to the Vendor
data	Object	Yes	Data object
denomination_id	Integer	Yes	Unique Denomination ID of the product you would like to top up from the Mintroute Systems
account_id	String	Yes	Free fire player ID for which you would like to have top up on, e.g. 10000335

Request Example:

For free fire top up account validation.

```
{"username":"testvendor","password":"testvendor","data":{"denomination_id":2633,"account_id":"10000335"}}
```

Returned Parameters:

L	Parameter	Type	Description
1	status	Boolean	Will return “true” if the provided account has been validated successfully otherwise an error message will be returned.
1	message	String	Status message of the top up validation if it is successful or not
1	account_details	Array/Object	Information regarding the user account.
2	account_id	String	Free fire player ID which has been validated for the top up.
2	roles	Array/Object	Free fire user account roles
3	role	String	User role name. (nickname of the user in game)
3	packed_role_id	Integer	ID of the provided role

Response Examples:

JSON Format

```
{
  "status": true,
  "message": "Account has been validated successfully",
  "account_details": {
    "account_id": "10000335",
    "roles": [{
      "server_id": 0,
      "server": "",
      "role_id": 0,
      "client_type": 0,
      "role": "JROMAN",
      "packed_role_id": 0
    }]
  }
}
```

If an error occurs during top up validation process

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

OR

```
{"status":false,"error":"Invalid denomination","error_code":"1155"}
```

OR

```
{"status":false,"error":"Invalid brand type","error_code":"1157"}
```

OR

```
{"status":false,"error":"Unable to find account identifier","error_code":"1158"}
```

OR

```
{"status":false,"error":"Invalid account","error_code":"1159"}
```

OR

```
{"status":false,"error":"You do not have a valid contract for the selected denomination","error_code":"1160"}
```

OR

```
{"status":false,"error":"You do not have enough balance to complete this purchase","error_code":"1161"}
```

4.2 api/account_topup

Below is the list of parameters required by the end point for free fire top up

Required Parameters:

Parameter	Type	Required	Description
username	String	Yes	Unique Username provided to the Vendor
password	String	Yes	Password provided to the Vendor
data	Object	Yes	Data object
denomination_id	Integer	Yes	Unique Denomination ID of the product you would like to top up from the Mintroute Systems
account_id	String	Yes	Free fire player ID for which you would like to have top up on, e.g. 10000335, 10000251
packed_role_id	Integer	Yes	Select role id from provided account role ids on top up validation response, e.g. 0, 1
order_id	String	No	An alpha numeric value generated and supplied from your side to uniquely identify a topup order.

Request Example:

For free fire top up,

```
{"username":"testvendor","password":"testvendor","data":{"denomination_id":2633,"account_id":"10000335","packed_role_id":0,"order_id":"OID1000"}}
```

Returned Parameters:

L	Parameter	Type	Description
1	status	Boolean	Will return "true" if the top has been delivered successfully otherwise an error message will be returned.
1	message	String	Status message of the top up transaction if it is successful or not
1	account_details	Array/Object	Information regarding the user account.
2	account_id	String	Free fire player ID for which the top up has been successful
2	transaction_id	String	Top up transaction ID. e.g. TXN1607591339-9763709-1520

Response Examples:

JSON Format

```
{
  "status": true,
  "message": "Account has been topped up successfully",
  "account_details": {
    "account_id": "10000335",
    "transaction_id": "TXN1607591339-9763709-1520"
  }
}
```

If an error occurs during top up process

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

OR

```
{"status":false,"error":"Invalid denomination","error_code":"1155"}
```

OR

```
{"status":false,"error":"Invalid brand type","error_code":"1157"}
```

OR

```
{"status":false,"error":"Unable to find account identifier","error_code":"1158"}
```

OR

```
{"status":false,"error":"Invalid account","error_code":"1159"}
```

OR

```
{"status":false,"error":"You do not have a valid contract for the selected denomination","error_code":"1160"}
```

OR


```
{"status":false,"error":"You do not have enough balance to complete this purchase","error_code":"1161"}
```

OR

```
{"status":false,"error":"Unable to find packed role ID":"1162"}
```

OR

```
{"status":false,"error":"Invalid packed role ID":"1163"}
```

5 Mobile Legends Top Up

5.1 api/account_validation

Below is the list of parameters required by the end point for mobile legends account validation

Required Parameters:

Parameter	Type	Required	Description
username	String	Yes	Unique Username provided to the Vendor
password	String	Yes	Password provided to the Vendor
data	Object	Yes	Data object
denomination_id	Integer	Yes	Unique Denomination ID of the product you would like to top up from the Mintroute Systems
account_id	String	Yes	Mobile legends account ID for which you would like to have top up on, e.g. 24760,
zone_id	String	Yes	Zone ID of your mobile legends account, e.g. 22001

Request Example:

For mobile legends top up account validation,

```
{"username":"testvendor","password":"testvendor","data":{"denomination_id":2683,"account_id":"27295","zone_id":"7000"}}
```

Returned Parameters:

L	Parameter	Type	Description
1	status	Boolean	Will return "true" if the provided account has been validated successfully otherwise an error message will be returned.
1	message	String	Status message of the top up validation if it is successful or not
1	account_details	Array/Object	Information regarding the user account.
2	username	String	Username (nick name) of the user in the game
2	account_id	String	Mobile legends ID which has been validated for the top up.
2	customer_id	String	Customer ID of the validated account
2	flow_id	String	Flow ID of the validated account

Response Examples:

JSON Format

```
{
  "status": true,
  "message": "Account has been validated successfully",
  "account_details": {
    "username": "Test user",
    "account_id": "27295",
    "customer_id":
"cEAPMTA0NTExOTQwOTc4Mjk3AfnMAwKpz9UoQxZkaWFtb35kX21pbnRyb3V0
ZVVTRF8xBJyqtvvgFgA==",
    "flow_id": "204511800978287"
  }
}
```

If an error occurs during top up validation process

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

OR

```
{"status":false,"error":"Invalid  
denomination","error_code":"1155"}
```

OR

```
{"status":false,"error":"Invalid brand type  
","error_code":"1157"}
```

OR

```
{"status":false,"error":"Unable to find account  
identifier","error_code":"1158"}
```

OR

```
{"status":false,"error":"Invalid  
account","error_code":"1159"}
```

OR

```
{"status":false,"error":"You do not have a valid contract  
for the selected denomination","error_code":"1160"}
```

OR

```
{"status":false,"error":"You do not have enough balance to  
complete this purchase","error_code":"1161"}
```

OR

```
{"status":false,"error":"Unable to find zone ID":"1164"}
```

OR

```
{"status":false,"error":"Invalid zone ID":"1165"}
```

5.2 api/account_topup

Below is the list of parameters required by the end point for mobile legends top up

Required Parameters:

Parameter	Type	Required	Description
username	String	Yes	Unique Username provided to the Vendor
password	String	Yes	Password provided to the Vendor
data	Object	Yes	Data object
denomination_id	Integer	Yes	Unique Denomination ID of the product you would like to top up from the Mintroute Systems
account_id	String	Yes	Mobile legends account ID for which you would like to have top up on, e.g. 24760
customer_id	String	Yes	Provided customer ID on top up validation response, e.g. cEAPMTA0NTExOTQwOTc4Mjk3AfnMAwKpz9UoQxZkaWFtb35kX21pbnRyb3V0ZVVTRF8xBJyqtvGfgA==
flow_id	String	Yes	Provided flow ID on top up validation response, e.g. 204511800978287, 334511780978290
order_id	String	No	An alpha numeric value generated and supplied from your side to uniquely identify a topup order.

Request Example:

For mobile legends top up,

```
{
  "username": "testvendor",
  "password": "testvendor",
  "data": {
    "denomination_id": 2683,
    "account_id": "27295",
    "customer_id": "cEAPMTA0NTExOTQwOTc4Mjk3AfnMAwKpz9UoQxZkaWFtb35kX21pbnRyb3V0ZVVTRF8xBJyqtvGfgA==",
    "flow_id": "204511800978287"
  }
}
```

Returned Parameters:

L	Parameter	Type	Description
1	status	Boolean	Will return "true" if the top has been delivered successfully otherwise an error message will be returned.
1	message	String	Status message of the top up transaction if it is successful or not
1	account_details	Array/Object	Information regarding the user account.

2	account_id	String	Account ID for which the top up has been successful.
2	transaction_id	String	Top up transaction ID. e.g. ML-24760-459c-400c-85d6-156eeda94b68

Response Examples:

JSON Format

```
{
  "status": true,
  "message": "Account has been topped up successfully",
  "account_details": {
    "account_id": "27295",
    "transaction_id": "ML-24760-459c-400c-85d6-156eeda94b68"
  }
}
```

If an error occurs during top up process

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

OR

```
{"status":false,"error":"Invalid denomination","error_code":"1155"}
```

OR

```
{"status":false,"error":"Invalid brand type", "error_code":"1157"}
```

OR

```
{"status":false,"error":"Unable to find account identifier","error_code":"1158"}
```

OR

```
{"status":false,"error":"Invalid account","error_code":"1159"}
```

OR

```
{"status":false,"error":" You do not have a valid contract for the selected denomination","error_code":"1160"}
```

OR

```
{"status":false,"error":" You do not have enough balance to complete this purchase","error_code":"1161"}
```

OR

```
{"status":false,"error":"Unable to find customer ID":"1166"}
```

OR

```
{"status":false,"error":"Unable to find flow ID":"1168"}
```

6 Razer Gold Top Up

Important Note: Account limit for Razer Gold basic is USD 300 and can be expanded by the user to USD 500 after upgrading to premium account.

6.1 api/account_validation

Below is the list of parameters required by the end point for razer gold account validation

Required Parameters:

Parameter	Type	Required	Description
username	String	Yes	Unique Username provided to the Vendor
password	String	Yes	Password provided to the Vendor
data	Object	Yes	Data object
denomination_id	Integer	Yes	Unique Denomination ID of the product you would like to top up from the Mintroute Systems
account_id	String	Yes	Razer gold account email for which you would like to have top up on, e.g. razergoldsandbox+gl3@gmail.com

Request Example:

For razer gold top up account validation,

```
{"username":"testvendor","password":"testvendor","data":{"denomination_id":2636,"account_id":"razergoldsandbox+gl3@gmail.com"}}
```

Returned Parameters:

L	Parameter	Type	Description
1	status	Boolean	Will return "true" if the provided account has been validated successfully otherwise an error message will be returned.
1	message	String	Status message of the top up validation if it is successful or not
1	account_details	Array/Object	Information regarding the user account.
2	account_id	String	Razer gold account email which has been validated for the top up.
2	validated_token	String	Account validation token
2	reference_id	String	Account validation reference ID
2	wallet_balance	String	Balance of the user wallet. e.g. 270
2	wallet_currency_code	String	Currency code of the user wallet. e.g. USD

Response Examples:

JSON Format

```
{
  "status": true,
  "message": "Account has been validated successfully",
  "account_details": {
    "account_id": "razergoldsandbox+gl3@gmail.com",
    "validated_token":
"7b809499f8e046c3adfb6a95bb094132",
    "reference_id": "RG-36e9-4580-b151-0725da7f6b70-
5f7b054f93642",
    "wallet_balance": "270",
    "wallet_currency_code": "USD"
  }
}
```

If an error occurs during top up validation process

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

OR

```
{"status":false,"error":"Invalid denomination","error_code":"1155"}
```

OR

```
{"status":false,"error":"Invalid brand type","error_code":"1157"}
```

OR

```
{"status":false,"error":"Unable to find account identifier","error_code":"1158"}
```

OR

```
{"status":false,"error":"Invalid account","error_code":"1159"}
```

OR

```
{"status":false,"error":"You do not have a valid contract for the selected denomination","error_code":"1160"}
```

OR

```
{"status":false,"error":"You do not have enough balance to complete this purchase","error_code":"1161"}
```

6.2 api/account_topup

Below is the list of parameters required by the end point for razer gold top up

Required Parameters:

Parameter	Type	Required	Description
username	String	Yes	Unique Username provided to the Vendor
password	String	Yes	Password provided to the Vendor
data	Object	Yes	Data object
denomination_id	Integer	Yes	Unique Denomination ID of the product you would like to top up from the Mintroute Systems
account_id	String	Yes	Razer gold account email for which you would like to have top up on, e.g. razergoldsandbox+gl3@gmail.com
validated_token	String	Yes	Provided validated token on top up validation, e.g. 7b809499f8e046c3adfb6a95bb094132
reference_id	String	Yes	Provided reference ID on top up validation, e.g. RG-36e9-4580-b151-0725da7f6b70-5f7b054f93642
order_id	String	No	An alpha numeric value generated and supplied from your side to uniquely identify a topup order.

Request Example:

For razer gold top up,

```
{
  "username": "testvendor",
  "password": "testvendor",
  "data": {
    "denomination_id": "2636",
    "account_email": "razergoldsandbox+gl3@gmail.com",
    "validated_token": "7b809499f8e046c3adfb6a95bb094132",
    "reference_id": "RG-36e9-4580-b151-0725da7f6b70-5f7b054f93642"
  }
}
```

Returned Parameters:

L	Parameter	Type	Description
1	status	Boolean	Will return "true" if the top has been delivered successfully otherwise an error message will be returned.
1	message	String	Status message of the top up transaction if it is successful or not

1	account_details	Array/Object	Information regarding the user account.
2	account_id	String	Razer gold account ID which has been validated for the top up.
2	transaction_id	String	Top up transaction ID. e.g. RG-439d-4209-b46a-08440e32c149-5fd1d8fce2eb3

Response Examples:

JSON Format

```
{
  "status": true,
  "message": "Account has been topped up successfully",
  "account_details": {
    "account_id": "razergoldsandbox+gl3@gmail.com",
    "transaction_id": "RG-439d-4209-b46a-08440e32c149-5fd1d8fce2eb3"
  }
}
```

If an error occurs during top up process

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

OR

```
{"status":false,"error":"Invalid denomination","error_code":"1155"}
```

OR

```
{"status":false,"error":"Invalid brand type","error_code":"1157"}
```

OR

```
{"status":false,"error":"Unable to find account identifier","error_code":"1158"}
```

OR

```
{"status":false,"error":"Invalid  
account","error_code":"1159"}
```

OR

```
{"status":false,"error":"You do not have a valid contract  
for the selected denomination","error_code":"1160"}
```

OR

```
{"status":false,"error":"You do not have enough balance to  
complete this purchase","error_code":"1161"}
```

OR

```
{"status":false,"error":"Unable to find validated  
token":"1170"}
```

OR

```
{"status":false,"error":"Unable to find reference ID":"1172"}
```

7 Jio Saavn Top Up

7.1 api/account_validation

Below is the list of parameters required by the end point for jio saavn account validation.

Required Parameters:

Parameter	Type	Required	Description
username	String	Yes	Unique Username provided to the Vendor
password	String	Yes	Password provided to the Vendor
data	Object	Yes	Data object
denomination_id	Integer	Yes	Unique Denomination ID of the product you would like to top up from the Mintroute Systems
account_id	String	Yes	Jio Saavn account id that you want to top-up e.g. +971521234567, +971847575638 (valid mobile number with country code)

Request Example:

Jio Saavn top up account validation.

```
{"username":"testvendor","password":"testvendor","data":{"denomination_id":2722,"account_id":"+971521234567"}}
```

Returned Parameters:

L	Parameter	Type	Description
1	status	Boolean	Will return "true" if the provided account has been validated successfully otherwise an error message will be returned.
1	message	String	Status message of the top up validation if it is successful or not
1	account_details	Array/Object	Information regarding the user account.
2	account_id	String	Jio Saavn account id which has been validated for the top up.

Response Examples:

JSON Format

```
{
  "status": true,
  "message": "Account has been validated successfully",
  "account_details": {
    "account_id": "+971521234567"
  }
}
```

If an error occurs during the account validation process the structure of errors is shown in the following examples

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

OR

```
{"status":false,"error":"Invalid denomination","error_code":"1155"}
```

OR

```
{"status":false,"error":"Invalid brand type", "error_code":"1157"}
```

OR

```
{"status":false,"error":"Unable to find account identifier","error_code":"1158"}
```

OR

```
{"status":false,"error":"Invalid account","error_code":"1159"}
```

OR

```
{"status":false,"error":"You do not have a valid contract for the selected denomination","error_code":"1160"}
```

OR

```
{"status":false,"error":" You do not have enough balance to complete this purchase","error_code":"1161"}
```

7.2 api/account_topup

Below is the list of parameters required by the end point for jio saavn top up

Required Parameters:

Parameter	Type	Required	Description
username	String	Yes	Unique Username provided to the Vendor
password	String	Yes	Password provided to the Vendor
data	Object	Yes	Data object
denomination_id	Integer	Yes	Unique Denomination ID of the product you would like to top up from the Mintroute Systems
account_id	String	Yes	Jio Saavn account id that you want to top-up e.g. +971521234567, +971847575638 (valid mobile number with country code)
order_id	String	No	A unique alpha numeric value generated and supplied from your side to uniquely identify a top up order.

Request Example:

Jio Saavn top up request

```
{"username":"testvendor","password":"testvendor","data":{"denomination_id":2722,"account_id":"+971521234567","order_id":"OID1001"}}
```

Returned Parameters:

L	Parameter	Type	Description
1	status	Boolean	Will return "true" if the top has been delivered successfully otherwise an error message will be returned.
1	message	String	Status message of the top up transaction if it is successful or not
1	account_details	Array/Object	Information regarding the user account.

2	account_id	String	Jio Saavn account id for which the top up has been successful
2	transaction_id	String	Top up transaction ID. e.g. TXN1617175898-2101-4090-9050-0a05b0d3be18-8765

Response Examples:

JSON Format

```
{
  "status": true,
  "message": "Account has been topped up successfully",
  "account_details": {
    "account_id": "+971521234567",
    "transaction_id": "TXN1617175898-2101-4090-9050-0a05b0d3be18-8765"
  }
}
```

If an error occurs during account top up process the structure of errors is shown in the following examples

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

OR

```
{"status":false,"error":"Invalid denomination","error_code":"1155"}
```

OR

```
{"status":false,"error":"Invalid brand type","error_code":"1157"}
```

OR

```
{"status":false,"error":"Unable to find account identifier","error_code":"1158"}
```

OR

```
{"status":false,"error":"Invalid account","error_code":"1159"}
```

OR

```
{"status":false,"error":"You do not have a valid contract for the selected denomination","error_code":"1160"}
```

OR

```
{"status":false,"error":"You do not have enough balance to complete this purchase","error_code":"1161"}
```

OR

```
{"status":false,"error":"Invalid packed role ID":"1163"}
```

8 Net Dragon (Eudemons Online, Conquer Online & Heroes Evolved) Top Up

8.1 api/account_initialization

Below is the list of parameters required by the end point for net dragon to get game server list.

Required Parameters:

Parameter	Type	Required	Description
username	String	Yes	Unique Username provided to the Vendor
password	String	Yes	Password provided to the Vendor
data	Object	Yes	Data object
denomination_id	Integer	Yes	Unique Denomination ID of the product you would like to top up from the Mintroute Systems

Request Example:

Net Dragon top up account initialization.

```
{"username":"testvendor","password":"testvendor","data":{"denomination_id":2824}}
```

Returned Parameters:

L	Parameter	Type	Description
1	status	Boolean	Will return “true” if the provided account has been validated successfully otherwise an error message will be returned.
1	message	String	Status message of the top up validation if it is successful or not
1	data	Array/Object	Server list consist of all available servers for the requested game.

Response Examples:

JSON Format

```
{
  "status": true,
  "message": "Server List fetched successfully",
  "data": {
    "serverList": [
      {
        "America": [
          {
            "serverName": "NewYork/Toronto",
            "serverId": "2",
            "isForTest": 0
          },
          {
            "serverName": "LA/LasVegas",
            "serverId": "3",
            "isForTest": 0
          },
          {
            "serverName": "Aries",
            "serverId": "10",
            "isForTest": 0
          },
          {
            "serverName": "Cronus",
            "serverId": "18",
            "isForTest": 0
          },
          {
            "serverName": "Uranus",
            "serverId": "26",
            "isForTest": 0
          },
          {
            "serverName": "SkyPark",
            "serverId": "27",
            "isForTest": 0
          },
          {
            "serverName": "Delphinus",
            "serverId": "33",
            "isForTest": 0
          }
        ]
      }
    ]
  }
}
```

```

    }, {
      "serverName": "Hyperion",
      "serverId": "35",
      "isForTest": 0
    }],
    "Europe": [{
      "serverName": "London",
      "serverId": "5",
      "isForTest": 0
    }, {
      "serverName": "DarkMarsh",
      "serverId": "29",
      "isForTest": 0
    }, {
      "serverName": "RoseHall",
      "serverId": "31",
      "isForTest": 0
    }, {
      "serverName": "Lyra",
      "serverId": "32",
      "isForTest": 0
    }],
    "Continent": [],
    "Asia": [{
      "serverName": "Jin",
      "serverId": "21",
      "isForTest": 0
    }, {
      "serverName": "ElvenCity",
      "serverId": "25",
      "isForTest": 0
    }, {
      "serverName": "Jupiter",
      "serverId": "28",
      "isForTest": 0
    }, {
      "serverName": "Draconis",
      "serverId": "34",
      "isForTest": 0
    }
  ]
}
}
}

```

If an error occurs during the account initialization process the structure of errors is shown in the following examples

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

OR

```
{"status":false,"error":"Invalid denomination","error_code":"1155"}
```

OR

```
{"status":false,"error":"Invalid brand type","error_code":"1157"}
```

OR

```
{"status":false,"error":"You do not have a valid contract for the selected denomination","error_code":"1160"}
```

OR

```
{"status":false,"error":"Unable to fetch game server list. Please try again or contact techsupport@mintroute.com","error_code":"1193"}
```

8.2 api/account_validation

Below is the list of parameters required by the end point for net dragon account validation.

Required Parameters:

Parameter	Type	Required	Description
username	String	Yes	Unique Username provided to the Vendor
password	String	Yes	Password provided to the Vendor
data	Object	Yes	Data object
denomination_id	Integer	Yes	Unique Denomination ID of the product you would like to top up from the Mintroute Systems
account_id	String	Yes	Net Dragon game account id that you want to top-up e.g. tqgm001 (registered game username is the account id)
server_id	String	Yes	Net Dragon server id that you want to top-up for e.g. 5, 32. (selected game server of the netdragon game account)

Request Example:

Net Dragon top up account validation.

```
{ "username": "testvendor", "password": "testvendor", "data": { "denomination_id": 2824, "account_id": "tqgm001", "server_id": "5" } }
```

Returned Parameters:

L	Parameter	Type	Description
1	status	Boolean	Will return "true" if the provided account has been validated successfully otherwise an error message will be returned.
1	message	String	Status message of the top up validation if it is successful or not
1	account_details	Array/Object	Information regarding the user account.
2	account_id	String	Net dragon game account id which has been validated for the top up.

Response Examples:

JSON Format

```
{
  "status": true,
  "message": "Account has been validated successfully",
  "account_details": {
    "account_id": "tqgm001"
  }
}
```

If an error occurs during the account validation process the structure of errors is shown in the following examples

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

OR

```
{"status":false,"error":"Invalid denomination","error_code":"1155"}
```

OR

```
{"status":false,"error":"Invalid brand type", "error_code":"1157"}
```

OR

```
{"status":false,"error":"Unable to find account identifier","error_code":"1158"}
```

OR

```
{"status":false,"error":"Invalid account","error_code":"1159"}
```

OR

```
{"status":false,"error":"Unable to find server ID","error_code":"1194"}
```

OR

```
{"status":false,"error":" You do not have a valid contract for the selected denomination","error_code":"1160"}
```

OR

```
{"status":false,"error":" You do not have enough balance to complete this purchase","error_code":"1161"}
```

8.3 api/account_topup

Below is the list of parameters required by the end point for net dragon top up

Required Parameters:

Parameter	Type	Required	Description
username	String	Yes	Unique Username provided to the Vendor
password	String	Yes	Password provided to the Vendor
data	Object	Yes	Data object
denomination_id	Integer	Yes	Unique Denomination ID of the product you would like to top up from the Mintroute Systems
account_id	String	Yes	Net Dragon game account id that you want to top-up e.g., tqgm001 (registered game username is the account id)
server_id	String	Yes	Net Dragon server id that you want to top-up for e.g., 5, 32. (selected game server of the netdragon game account)
order_id	String	No	A unique alpha numeric value generated and supplied from your side to uniquely identify a top up order.

Request Example:

Net Dragon top up request

```
{"username":"testvendor","password":"testvendor","data":{"denomination_id":2722,"account_id":"+971521234567","order_id":"OID1001"}}
```

Returned Parameters:

L	Parameter	Type	Description
1	status	Boolean	Will return "true" if the top has been delivered successfully otherwise an error message will be returned.
1	message	String	Status message of the top up transaction if it is successful or not

1	account_details	Array/Object	Information regarding the user account.
2	account_id	String	Net Dragon game account id for which the top up has been successful
2	transaction_id	String	Top up transaction ID. e.g., TXN1623325203-9298-42a6-b817-26d86ac45955-tqg

Response Examples:

JSON Format

```
{
  "status": true,
  "message": "Account has been topped up successfully",
  "account_details": {
    "account_id": "tqgm001",
    "transaction_id": "TXN1623325203-9298-42a6-b817-26d86ac45955-tqg"
  }
}
```

If an error occurs during account top up process the structure of errors is shown in the following examples

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

OR

```
{"status":false,"error":"Invalid denomination","error_code":"1155"}
```

OR

```
{"status":false,"error":"Invalid brand type","error_code":"1157"}
```

OR

```
{"status":false,"error":"Unable to find account identifier","error_code":"1158"}
```

OR

```
{"status":false,"error":"Invalid account","error_code":"1159"}
```

OR

```
{"status":false,"error":"Unable to find server ID","error_code":"1194"}
```

OR

```
{"status":false,"error":"You do not have a valid contract for the selected denomination","error_code":"1160"}
```

OR

```
{"status":false,"error":"You do not have enough balance to complete this purchase","error_code":"1161"}
```

OR

```
{"status":false,"error":"Invalid packed role ID":"1163"}
```

9 Error Handling

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

CODE	DESCRIPTION
1033	Duplicate Order ID Provided
1139	Request fails authentication due to invalid header format
1140	Invalid date parameter
1145	No such vendor exists, or vendor is not active
1146	Required data is missing
1147	Input data format was not as required, please provide correct data format
1141	Signature expired
1149	Request fails authentication

1150	Request contains invalid strings
1151	Username is missing in the request
1153	No such user exists, or user is not active
1154	Unable to find denomination identifier
1155	Invalid denomination
1157	Invalid brand type
1158	Unable to find account identifier
1159	Invalid account
1160	You do not have a valid contract for the selected denomination
1039	You do not have enough balance to complete this purchase
1162	Unable to find packed role ID
1163	Invalid packed role ID
1164	Unable to find zone ID
1165	Invalid zone ID
1166	Unable to find customer ID
1168	Unable to find flow ID
1170	Unable to find validated token
1172	Unable to find reference ID
1173	Server temporarily not available
1174	Role not found
1175	Unable to get the response, please contact techsupport
1176	An error occurred, please contact techsupport
1177	Internal system error, please contact techsupport
1178	Invalid data, please contact techsupport
1179	Topup region blocked
1180	Invalid user wallet or wallet currency.
1181	User Wallet reached top up limit
1182	Invalid product details
1183	We are undergoing scheduled maintenance. Please try again after an hour or contact techsupport@mintroute.com
1184	Hourly limit exceeded
1185	Unable to handle the request, please contact techsupport
1186	Daily limit exceeded
1187	Could not create a subscription for auto renewal users

1188	Unable to create a user with the Phone Number
1189	This product can be bought only once
1190	This product can be bought only once per month
1191	This product can be bought only once per quarter
1192	Request has not completed and in middle of processing
1192	Unable to fetch game server list. Please try again or contact techsupport@mintroute.com
1194	Unable to find server ID
1196	User is not eligible to purchase item

10 Handling Timeouts

Your API request to Mintroute can time out because of a network glitch or a server error. We recommend that you implement a wait of at-least 60 seconds to hear back the Mintroute response. We would also recommend using 'order_id' parameter in your top calls. When implemented it can allow you to resubmit your timed out requests without accidentally causing double topups. Mintroute always looks for a unique order_id when it is provided in the request. If your first request was accepted by us and you send another request with same order_id then Mintroute will respond back Duplicate Order ID error. This will confirm that your previous request has already been successful.

Please wait for two minutes before retrying timed out requests to accommodate for network and server delays.

11 Top Up Test Accounts

The following test accounts can be used in the integration process. These accounts are only for testing purpose & are not valid on production server.

MERCHANT	TEST ACCOUNTS
Free Fire	Account Id: 10000335
Mobile Legends	Account Id: 27295 & Zone Id: 7000
Razer Gold	razergoldsandbox+gl001@gmail.com razergoldsandbox+gl002@gmail.com razergoldsandbox+gl003@gmail.com razergoldsandbox+gl004@gmail.com razergoldsandbox+gl005@gmail.com

	up to ... razergoldsandbox+gl015@gmail.com
Jio Saavn	Any valid mobile number with country code e.g: +971521234567
Net Dragon (Eudemons Online)	Account ID: tqgm001 Server ID: 5
Net Dragon (Conquer Online)	Account ID: tqgm001 Server ID: 5
Net Dragon (Heroes Evolved)	Account ID: 5e84385ba0eeb84e5f0e0f24#620000 Server ID: 132