



Smartcall Web Service V3 (RESTful)

Technical Interface Specification

Version 1.1.4

Document Change History

Issue	Revision	Date	Author	Reason for Change
1	1.0.0	2017-07-27	Derek Scotney	Initial release
2	1.0.1	2017-08-04	Derek Scotney	Security information
3	1.0.2	2017-08-07	Derek Scotney	Product offering update checks using Etag headers implemented
4	1.0.3	2017-09-04	Derek Scotney	Added a "Getting Started" section
5	1.0.4	2017-09-08	Derek Scotney	Added "FAQ" and "API Operations" sections
6	1.0.5	2017-11-15	Derek Scotney	Added "Cashup" functionality to the Smartload operations and "Token" management functionality to the Authentication operations
7	1.0.6	2018-01-10	Derek Scotney	<ul style="list-style-type: none"> - Added information regarding the automatic token invalidation after 4 hours of non-usage - Added error code and response code information
8	1.0.7	2018-05-23	Derek Scotney	Added "SIM Network" query
9	1.0.8	2018-06-21	Derek Scotney	Added "Query SIM RICA status"
10	1.0.9	2018-08-01	Derek Scotney	- Added reminder to save Smartload reference
11	1.1.0	2019-01-22	Derek Scotney	<ul style="list-style-type: none"> - Added synchronous recharging - Added mobile network connectivity status check (up/down) - Removed recharge cancellation functionality
12	1.1.1	2019-02-22	Derek Scotney	- Added more information to the synchronous recharging endpoint
13	1.1.2	2019-05-07	Derek Scotney	- Updated the information related to the synchronous recharge response code
14	1.1.3	2019-05-21	Derek Scotney	- Updated the information relating to the smsProviderIdentifier field in the synchronous recharge request
15	1.1.4	2019-05-29	Derek Scotney	- Added a new endpoint under RICA whereby an MSISDN can be queried to see if it is a) a Smartcall RICA agent, b) if the agent is enabled, and c) if the agent falls under the clients master dealer

Contents

Document Change History	2
Abbreviation List.....	5
1. Introduction	6
2. Audience	6
3. RESTful alternative to V2.....	6
4. Documentation & Swagger UI.....	7
5. Authentication	7
6. Getting started.....	9
7. Etag headers for product offering update checking	9
8. API Smartload Operations	10
a. Balance Check.....	10
b. Get product information.....	10
c. Get single network products.....	10
d. Get all network products	10
e. Dealer registration check	10
f. Recharge request	10
g. Synchronous recharge request	10
h. Batch recharge request.....	11
i. Recharge Prevend request	11
j. Funds transfer.....	11
k. Order batch vouchers.....	11
l. Retrieve batch order file	11
m. Recreate previous order file	11
n. Transaction enquiry	12
o. Cashup report (daily).....	12
p. Cashup report (period).....	12
9. API SmartRica Operations	13
a. Change of ownership	13
b. Submit registration	13
c. Query SIM RICA status.....	13
d. Agent RICA query	13
10. API Utilities Operation	13
a. SIM Network	13
b. Mobile network status	13
11. API Authentication Operations	13
a. Authentication	13
b. Token invalidation	13
c. Token flush	14
d. Token query.....	14
12. F.A.Q.	15

1. Q: When I get a list of products, each product ID seems to be duplicated with only the “smsIndicator” field being different.	15
2. Q: When I send a prevend request I get a failure informing me that the product ID is invalid, but it is in the product list.....	15
3. Q: What is the difference between a “Prevend” and a “Recharge”?.....	15
4. Q: Why do some recharges fail even though I get a “SUCCESSFUL” response to my recharge request?	15
5. Q: When I send an “authorization request”, I get a response with HTTP Code 429.....	15
6. Q: Where do I enter my Smartload PIN in the various requests?.....	15
7. Q: The balance on my Smartload QA account is zero, how can I top it up?	16
8. Q: I can't read the Smartcall certificate. What do I need to do?	16
9. Q: Is there a charge for using the Smartcall web service API?	16
10. Q: What are the possible responseCode values for recharges?	16
11. Q: What are the possible error code values for recharges?	17

Figures

Figure 1: Authorization Message Flows	8
Figure 2: Response Codes	16

Tables

Table 1: Recharge Response Code.....	16
Table 2: ERROR Codes	17

Abbreviation List

FTP	File Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
ICCID	Integrated Circuit Card ID 19 or 20-digit serial number of the SIM card
MSISDN	Mobile Subscriber Integrated Services Digital Network Number (or cell phone number)
OAuth 2.0	OAuth 2.0 is an authorization framework that enables applications to obtain limited access to user accounts on an HTTP service
REST	Representational State Transfer
SIM	Subscriber Identity Module (Card)
SOAP	Simple Object Access Protocol
SSL	Secure Socket Layer
USSD	Unstructured Supplementary Service Data
WS	Web Service

1. Introduction

This document describes Smartcall's RESTful Smartload web service which provides the opportunity for dealers to directly access Smartload and SmartRica by creating their own client interface.

The purpose of this document is to provide a clear technical guideline on how to connect to the webservice and how to call the individual functions as provided by the web service. This would cover amongst others, calls to perform and query individual airtime recharge requests.

Smartload specific details and the associated rules are covered in the complementary Business Specification document. The Business Specification document should be read in conjunction with this document when implementing your own client solution.

2. Audience

This document is for developers wishing to create a secure client to connect to Smartcall's web service using the new RESTful interface, to access both Smartload and SmartRica functionality.

3. RESTful alternative to V2

Smartcall has released this RESTful version of its V2 web service to both provide a RESTful web service as well as changing its web security model from "WS-Security" to the "OAUTH 2.0" model. The new web service (V3) runs in conjunction with V1 and V2, and existing users need not do anything.

New users should preferably use the new service unless there is a specific requirement to use V2.

4. Documentation & Swagger UI

The RESTful webservice interface is described in a swagger file which can be accessed (**for the test service**) at:

<https://www.smartcallesb.co.za:8101/webservice/swagger.json>

The content of this file describes each endpoint, the HTTP method (GET/POST/DELETE) with which to access it, and the data models of all the JSON messaging objects. A web browser based interface to this file is provided (Swagger UI) at:

<https://www.smartcallesb.co.za:8101/webservice/api>

This web interface also provides the functionality to test each endpoint.

* Launching the web page from the link above may result in the “https” being removed in the browser and the page failing to load. In this case, simply add the “https” back into the url.

** The URL is case sensitive

5. Authentication

As mentioned above, the RESTful web service security will be using the (2-Step) OAUTH 2.0 model over HTTPS. Before any web service calls (other than the ping test) can take place successfully, the user must first be authenticated and then use the security token returned in all subsequent calls.

Note: Unlike V1 & V2 when the Smartload MSISDN/PIN was used for authentication, V3 uses a user defined username and password combination. One or more Smartload accounts can then be linked to those user credentials.

The Authentication message flow is described below.

Step 1: A login/authentication is done by submitting an HTTP POST request to the endpoint “.../webservice/auth” with the “Authorization” field in the HTTP header populated with “Basic” and the “username:password” string Base64 encoded.

Step 2: The login credentials are validated by the Smartcall server and if successful, a response object containing a time-based token is returned.

Step 3: All endpoints are now accessible with the HTTP header field “Authorization” populated with “Bearer” and the supplied token.

Step 4: Although the security token will **automatically expire after 24 hours (or 4 hours of no usage)**, functionality is provided to invalidate the token at the end of a session if a user so wishes. This is accomplished by submitting an HTTP DELETE request to the same endpoint used for authentication. Once again, the HTTP header field “Authorization” populated with “Bearer” and the supplied token is required.

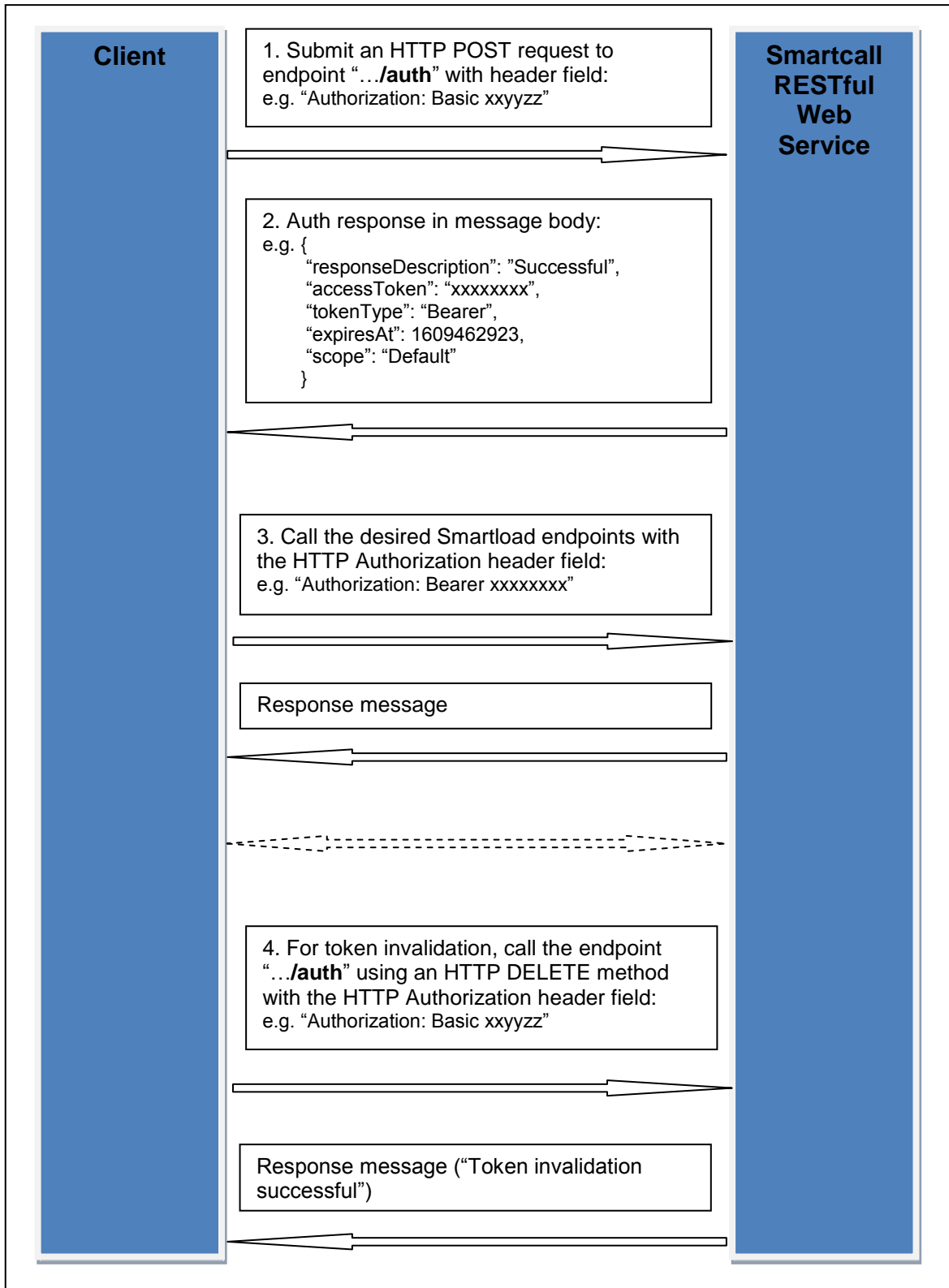


Figure 1: Authorization Message Flows

6. Getting started

To enable you to get started, you will need a username and password. These can be chosen by you and provided to the relevant developer/sales person you are in contact with. Please also provide 1 or more Smartload account numbers (MSISDNs) in order to be able to access the API business functions (balance/recharge etc).

Using these credentials you will then be able to test on the QA system using the URLs provided above. Once you are ready to access the LIVE API, you can provide us with the credentials you would like to use so we can set up your account. The URL remains the same with the exception of the port which changes to 8100.

Note: It is possible to restrict access to a specific account on an IP address basis. If you require this, please also supply the IP address(s) of the machines that will be interacting with the API.

7. Etag headers for product offering update checking

To make it easier to check for changes to the Smartload product offerings, “Etag” and “If-None-Match” HTTP headers have been implemented for the “products” and “networks” endpoints. In the response message of the standard GET method call, a new header “Etag” is returned which contains a hash value of the result data on the server side. In subsequent similar transactions, you can include the header “If-None-Match” with the last obtained Etag value. Should the data (product/network) not have changed, the response from the server will have an HTTP status code of 304 (Not Modified) and the body will be empty. Should the data have been modified, the standard result data will be returned along with the Etag header containing the new hash value.

To see this in a practical example, please use the Swagger UI interface provided.

8. API Smartload Operations

a. Balance Check

Provides the functionality to perform a balance check on any Smartload MSISDN registered to the user.

b. Get product information

Obtains information for a specified product ID.

c. Get single network products

Lists all the products available from a specified network.

d. Get all network products

Returns a list of all the Smartload products available from all the networks.

e. Dealer registration check

Checks whether the specified MSISDN is a registered Smartload dealer.

f. Recharge request

Provides the web service user with the functionality to perform a recharge operation with either a mobile network product or electricity. It should be noted that a successful response only indicates a successful submission to Smartcall and not a final transaction status. This status can be obtained using the “Transaction Query” operation.

NB: Please store the Smartload reference number returned from a successful recharge transaction as you will need it for doing recons with our reports.

g. Synchronous recharge request

Provides the web service user with the functionality to perform a fully synchronous recharge operation for mobile network products. Electricity purchases can not be handled synchronously and will function the same as with a normal recharge request. Unless a “pending” response is received, the response received indicates a final transaction status. In the case of a “pending” response, the final status must be obtained using the “Transaction Query” operation.

Please note that the field “smsProviderIdentifier” *should only be populated* in the recharge request if a branded SMS should be sent.

In the QA environment, recharge requests are handled by a basic “**network simulator**”, and as such, predefined responses for various scenarios are returned. Current features are:

- The simulator drops each network for 5 minutes individually to allow for failure scenarios to be tested. E.g Vodacom down, all up, MTN down, all up...
- Recharges of R11 are interpreted as “Business failures”. This is when the network sends a response indicating that the recharge is invalid.

***NOTE:** In the event of a recharge failure, the “error” field in the response message will be populated. The “statusCode/statusMessage” fields may also be populated. If the “error” field is *null*, and the “statusCode” is 0, then the transaction should be deemed as “SUCCESSFUL”. A *non-zero* or *null* “statusCode” should always be accompanied by a populated “error” field. The “statusCode” field values are:

- 0 – Successful
- 1 – Application/Internal error
- 2 – Business error (e.g. invalid recharge request information)

Where possible, the “statusMessage” field is populated with the [mobile network recharge response](#) information, and where these fields are not populated, the error field provides relevant information.

h. Batch recharge request

This is similar to the “Recharge Request”, but differs in that a batch of up to 100 recharge requests can be submitted within a single operation.

i. Recharge Prevend request

This is not a recharge operation, but a query to ascertain if a recharge operation “would be successful”, and is only available for PINLESS products on the Vodacom and MTN networks. It is a much quicker operation than a recharge as it is submitted directly to the relevant network, and a response code and the unfiltered response is returned. A SUCCESS response code, indicates the recharge will go through. APP_ERROR indicates the recharge will not go through, and SYS_ERROR indicates an infrastructure issue. An actual recharge performed in conjunction with this prevent MUST use the same client reference number.

j. Funds transfer

This provides the facility to transfer funds between two Smartload accounts.

k. Order batch vouchers

This is not a recharge operation, but rather an order for a batch of “PINNED” vouchers. Once the request has been processed, the response message will contain all the information required for retrieving the file containing the voucher PINs.

l. Retrieve batch order file

This is the subsequent call to the call listed above. By providing the relevant information, a password protected “zipped” file containing the voucher PINs is downloaded. The password to the file is provided in the initial request response.

m. Recreate previous order file

In the event that the information returned in the initial order response get lost (e.g. the password), this operation “re-creates” the file with the same content but a new password that is returned in the response.

n. Transaction enquiry

This operation provides the user with the ability to query a specific transaction status. A typical case would be after a recharge has been submitted, one would use this operation to find out if the recharge was successful. The status will be “pending” until a final status is received from the relevant network, in which case the status will move to “success” or “failure”.

o. Cashup report (daily)

This operation provides the user with the ability to get a basic cashup report for the specified Smartload account for the current day.

p. Cashup report (period)

This operation provides the user with the ability to get a basic cashup report for the specified Smartload account for the period specified.

9. API SmartRica Operations

a. Change of ownership

Provides the functionality to change the RICA registered owner of a SIM.

b. Submit registration

Provides the functionality to submit a new RICA registration

c. Query SIM RICA status

Provides the functionality to query the status of a submitted RICA request

d. Agent RICA query

Queries the RICA status of a provided MSISDN to determine if the MSISDN is:

- A Smartcall RICA agent
- In an *enabled* state
- If the agent falls under the clients master dealer

10. API Utilities Operation

a. SIM Network

Provides the facility to check the current mobile network of a SIM by passing in the MSISDN.

b. Mobile network status

This operation provides the user with the ability to get the current connectivity status between Smartcall and the various mobile networks.

11. API Authentication Operations

a. Authentication

Provides the functionality to authenticate a user using Basic authentication, which on success returns a JWT token (valid for 24 hours) for use in subsequent transactions with the API.

b. Token invalidation

Allows a user to invalidate a token once they have finished transacting instead of waiting until the token expires. **Tokens should be reused as much as possible.**

c. Token flush

Invalidates ALL the current users tokens. In the event the user is accessing this endpoint using a token for authentication, the current token will also be invalidated. This endpoint can also be accessed using the Basic authentication option used for "Authentication".

d. Token query

Provides the user with the ability to query how many tokens of the initial maximum of 20 allowed can still be requested (additional authentications). This endpoint can be accessed using both the Basic and Bearer authentication options.

12. F.A.Q.

1. **Q: When I get a list of products, each product ID seems to be duplicated with only the “smsIndicator” field being different.**

A: The reason for the “duplication”, is that in the future, when an sms notification is requested to be sent, the discount may be lower. It should be noted though that the relevant mobile network will generally send a recharge notification.

2. **Q: When I send a prevent request I get a failure informing me that the product ID is invalid, but it is in the product list.**

A: Prevent requests are only allowed for “pinless” products, and only available for Vodacom and MTN.

3. **Q: What is the difference between a “Prevent” and a “Recharge”?**

A: A “recharge” is immediately logged into Smartcall’s recharging system and submitted to the relevant network “asynchronously”. Due to the queueing process and possible load issues (for example), the actual submission to the network might not my immediate. Due to the asynchronous process, a “transaction query” is required to find out if the recharge transaction was ultimately successful. In a very, very small percentage of cases, there are failures returned by the networks for various reasons. A prevent takes a slightly different route though our system in that it is submitted directly (synchronously) to the relevant network (only **Vodacom** and **MTN** provide this feature) and we get back an immediate indication as to whether the transaction is expected to be successful or not. If the prevent is successful, a recharge can then be done and **MUST** use the same reference number as used for the prevent.

4. **Q: Why do some recharges fail even though I get a “SUCCESSFUL” response to my recharge request?**

A: A “successful” recharge response is only an indication of a successful recharge submission to the Smartcall web service. Since it is only submitted to the relevant mobile network asynchronously after submission, the network can still fail the transaction for a variety of reasons.

5. **Q: When I send an “authorization request”, I get a response with HTTP Code 429.**

A: The cause of this issue is generally because you have performed multiple authorization requests (and received security tokens), without invalidating the tokens you are finished with. The web service allows for a maximum of 20 concurrent sessions per user account.

6. **Q: Where do I enter my Smartload PIN in the various requests?**

A: With the new webservice interface, we have upgraded the security model to use a username and password instead of the Smartload account number (MSISDN) and 4-digit PIN. Your Smartload account is linked to your web service user account, and no further validation is required.

7. **Q: The balance on my Smartload QA account is zero, how can I top it up?**

A: Please send a topup request email to developers@smartcall.co.za and we will add funds to your QA account. Please remember to supply the Smartload account number that you would like topped-up.

8. **Q: I can't read the Smartcall certificate. What do I need to do?**

A: In some cases you may need to have the [thawte_SSL_CA_G2.cer](#) certificate installed.

9. **Q: Is there a charge for using the Smartcall web service API?**

A: There are **no** charges for using the API.

10. **Q: What are the possible responseCode values for recharges?**

A: See below

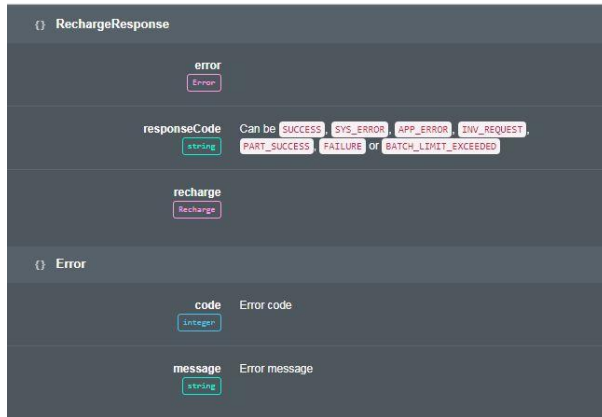


Figure 2: Response Codes

Response Code	Description
SUCCESS	Successful recharge submission to Smartcall. In the case of Pinless recharges, it does not mean the the recharge will be successful when sent to the network. Final recharge status can be queried using the "Transaction Query" API call.
SYS_ERROR	Typically a fault on the interface between Smartcall and the relevant mobile network
APP_ERROR	Typically a problem with the content of the recharge request (<i>error codes below</i>)
INV_REQUEST	This code is specific to Prevend requests and means there was a problem with the content of the prevend request message
PART_SUCCESS	This is used for Batch Recharges where not all the recharge requests contained in the batch are successful
FAILURE	This is used for Batch Recharges where ALL the recharges in the batch are unsuccessful
BATCH_LIMIT_EXCEEDED	The maximum number of recharges in the Batch Recharge Request has been exceeded (max 100)

Table 1: Recharge Response Code

11. Q: What are the possible error code values for recharges?

A: See below.

ERROR Code	Description
2	General System Error, recharged failed
3	Invalid owner Cell No provided
5	Invalid recharge network
6	Invalid recharge Cell No provided
8	A meter number is compulsory for an electricity recharge
9	Invalid Offering ID (product code)
10	Pinless Indicator Error: Smartload only distributes vouchers on the selected network
12	The Owner Cell No is no longer an active SMARTLOAD dealer
15	Insufficient funds in the SmartLoad wallet - Current balance: R...
16	A recharge with this meter no. has been received. Please wait for it to complete before retrying
17	The prevend has expired
18	Error (No Stock)
19	The quantity must be greater than 0
1001	Duplicate recharge (usually when customer recharge ID is re-used)
1002	Recharge still being processed
1003	Generic recharge failure
1004	Recharge response timed out, query to validate if recharge was submitted
1005	ClientReference does not exist
1006	Recharge Timeout, Please query recharge for status
1007	Recharge parameter validation failure
1008	Recharge does not exist
2001	Batch order does not exist

Table 2: ERROR Codes