

# SWAZIMOBILE – TOPUPSERVICE API User Manual Seamless Distribution Systems - ERS 360°



## Document History

Version	Date	Changes	Made By
Version 2	12-04-2018	Initial Version	Nanaba Michael Gyamfi
Version3	30-07-2020	Revised version	Queen Latifa Abdulai

## Contents

<b>1 Overview.....</b>	<b>4</b>
1.1 Purpose.....	4
1.2 Intended Audience .....	4
<b>2 Server Information.....</b>	<b>5</b>
<b>3 General Concepts.....</b>	<b>6</b>
3.1 Client context .....	6
3.1.1 Client Context Field Properties .....	6
3.1.2 clientId.....	7
3.1.3 clientId.....	7
3.2 Client transaction data.....	7
3.2.1 clientReference.....	8
3.2.2 clientTag .....	8
3.2.3 clientComment .....	8
<b>4 General Services .....</b>	<b>9</b>
4.1 Overview .....	9
4.1 Topup Service .....	9
4.1.1 RequestTransfer API .....	9
4.1.2 Sample RequestTransfer webservice request.....	9
4.1.3 RequestTransfer Request Field Breakdown.....	10
4.1.4 Sample RequestTransfer Response .....	11
4.1.5 Transfer Response Field breakdown .....	13
<b>5 Result Codes.....</b>	<b>14</b>
5.1 Sample unsuccessful web service response: .....	14
5.2 Result Codes: .....	14
5.3 PRODUCTS.....	16

## 1 Overview

ERS Web Services act as an intermediary service to serve requests from external clients seeking services from ERS. ERS Web Services enable operators or third-party applications to communicate with ERS System. This service will take the client requests and call ERS after mapping the request. After execution of this service, it will convert ERS response to client's desired response and send it back to client application.

### 1.1 Purpose

ERS Web Services APIs enable operators or third-party applications to communicate with ERS System. The purpose of this service is to enable web services through our system. The system takes the client's web services requests and calls ERS after mapping the request. After execution of this service, it will convert ERS response to client's desired response and send it back to client application.

### 1.2 Intended Audience

This document serves as a guide to enable Third Party Platforms (3PPs) and integrate those platforms with ERS to enable web services. The document provides information to the technical personnel of third-party service providers who wish to get information about the ERS system's third-party service enablers and the steps involved in carrying out requests and responses when a third-party application communicates with the ERS system to carry out a web service. ERS system information at the service provider's end is assumed.

## 2 Server Information

**PROD URL:**

**<https://10.1.24.169:8913/topupservice/service>**

**<https://10.1.24.169:8913/topupservice/service?wsdl>**

**STAGING:**

**<https://10.1.24.173:8913/topupservice/service>**

**<https://10.1.24.173:8913/topupservice/service?wsdl>**

## 3 General Concepts

This section will explain some common concepts and structures.

### 3.1 Client context

This is a structure used in all requests to identify and authorize the client performing the transaction. Since all production transactions take place over a secure channel (typically HTTPS), this is sent in clear text.

#### 3.1.1 Client Context Field Properties

SN	Field Name	Description
1	Channel	This is the channel through which client is performing the transaction. Eg. webservice
2	clientId	This is the unique id of the client that is performing the transaction.
3	clientRequestTimeout	The client side timeout for the request. If the server takes longer time than this to process the request it will attempt to abort or rollback the request.
4	initiatorPrincipalId	The authentication details of the VTU reseller that is performing the transaction. This usually consists of an id, type and userId .e.g.  <pre>&lt;initiatorPrincipalId&gt;   &lt;id&gt;\$reseller1&lt;/id&gt;   &lt;type&gt;RESELLERUSER&lt;/type&gt;   &lt;userId&gt;\$username&lt;/userId&gt; &lt;/initiatorPrincipalId&gt;</pre>
5	Password	The password associated with the userId field in initiatorPrincipalId field
6	clientReference	A client reference for the transaction. Should be unique and the maximum allowed field length is 32 characters.
7	clientTag	Custom tag(s) that the client wants to associate with this transaction, expressed as a single string. This will typically (if used) not be unique per transaction and can be used to label transactions and used is useful for report filtering The maximum allowed field length is 64 characters.
8	clientComment	Comment that the client wants to assign to this transaction. The maximum allowed field length is 64 characters.

### 3.1.2 clientId

ClientId will contain type of client and unique id of the client. Client Type is appended before the client unique id and separated with “:”

#### Example clientId

RESELLERMSISDN: 26879212607

clientId Type		
Name	Description	clientUserId (Mandatory)
TERMINALID	This type should be used when reseller will use his terminal id	No
ADMINUSER	This type should be used when an admin user is using the service	No
RESELLERMSISDN	This type should be used when reseller will use his MSISDN	No
RESELLERUSER	This type should be used when reseller will use one of his user account	Yes

### 3.1.3 clientUserId

clientUserId is a specific user of a reseller (client). clientUserId can be mandatory based on the clientId Type.

## 3.2 Client transaction data

This is a structure used in most requests that are not only reading data. It can be used for the client to send data to identify the transaction afterwards using references and tags that the client system recognizes.

The client transaction data has the following fields:

### **3.2.1 clientReference**

The client reference for the transaction should be unique at least for the specific client id to be usable.

The maximum length allowed for this field is 32 characters.

### **3.2.2 clientTag**

Tag(s) that the client wants to associate with this transaction, expressed as a single string. This will typically (if used) not be unique per transaction and can be used to label transaction that can then be reported as part of transaction clearing, reports and/or invoicing. If multiple tags are desired, it is up to the client to partition this string with delimiters, fixed lengths etc.

The maximum allowed length field is 128 characters.

### **3.2.3 clientComment**

Comment that the client wants to assign to this transaction. The maximum allowed length field is 256 characters.



## 4 General Services

Serial No.	Function Name	Description
1.	requestTransfer	Transfer STOCK from a reseller to another reseller

### 4.1 Overview

VTU 3rd party API interface provides a SOAP/XML based web service which acts as an intermediary to serve requests from external client seeking services from VTU system. The proxy takes the client request and calls VU after mapping the request. After request execution, the proxy will convert VTU response to client's desired response and send it back to client application/system.

Below chapters contain detailed description of each of these services including the request and response details to carry out these services.

### 4.1 Transfer Service

Through this service, a reseller buys stock from another reseller. The webservice method used for this service is **requestTransfer**.

#### 4.1.1 RequestTransfer API

The requestTransfer function allows a reseller to receiver stock from another reseller.

#### 4.1.2 Sample RequestTransfer webservice request

```
<soapenv:Envelope xmlns:ext="http://external.interfaces.ers.seamless.com/"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
    <ext:requestTransfer>
      <context>
        <channel>WebService</channel>
        <clientId>${CLIENT_NAME}</clientId>

        <clientReference>${Unique_Request_Reference}</clientReference>
        <initiatorPrincipalId>
```

```

        <id>$resellerid</id>
        <type>RESELLERUSER</type>
        <userId>$username</userId>
    </initiatorPrincipalId>
    <password>$PWD</password>
</context>
<senderPrincipalId>
    <id>$resellerid</id>
    <type>RESELLERID</type>
</senderPrincipalId>
<receiverPrincipalId>
    <id>$resellerid2</id>
    <type>RESELLERID</type>
</receiverPrincipalId>
<senderAccountSpecifier>
    <accountTypeId>RESELLER</accountTypeId>
</senderAccountSpecifier>
<receiverAccountSpecifier>
    <accountTypeId>RESELLER</accountTypeId>
</receiverAccountSpecifier>
<productId>TRANSFER</productId>
<amount>
    <currency>SZL</currency>
    <value>10</value>
</amount>
</ext:requestTransfer>
</soapenv:Body>
</soapenv:Envelope>

```

### 4.1.3 RequestTransfer Request Field Breakdown

Request field for **requestTransfer** is given below:

Name	Description
context	The client context
senderPrincipalId	<p>The id of the sender reseller. This has to be prefixed with the type of the id (which is typically RESELLERUSER) and UserId defines which user made the transfer.</p> <p>Example:</p> <pre> &lt;senderPrincipalId&gt;     &lt;id&gt;\$resellerid&lt;/id&gt;     &lt;type&gt;RESELLERID&lt;/type&gt; &lt;/senderPrincipalId&gt; </pre>

<p>receiverPrincipalId</p>	<p>The id of the receiving reseller</p> <p>Example:</p> <pre> &lt;receiverPrincipalId&gt;   &lt;id&gt;\$resellerid2&lt;/id&gt;   &lt;type&gt;RESELLERID&lt;/type&gt; &lt;/receiverPrincipalId&gt; </pre>
<p>senderAccountSpecifier</p>	<p>Account type of sender reseller</p> <p>Example:</p> <pre> &lt;senderAccountSpecifier&gt;   &lt;accountTypeId&gt;RESELLER&lt;/accountTypeId&gt; &lt;/senderAccountSpecifier&gt; </pre>
<p>receiverAccountSpecifier</p>	<p>Account type of receiver reseller</p> <p>Example:</p> <pre> &lt;receiverAccountSpecifier&gt;   &lt;accountTypeId&gt;RESELLER&lt;/accountTypeId&gt; &lt;/receiverAccountSpecifier&gt; </pre>
<p>productId</p>	<p>ID of the product, which is typically Transfer</p> <p>Example:</p> <pre> &lt;productId&gt;TRANSFER&lt;/productId&gt; </pre>
<p>amount</p>	<p>The amount to be transferred</p> <pre> &lt;amount&gt;   &lt;currency&gt;SZL&lt;/currency&gt;   &lt;value&gt;2&lt;/value&gt; &lt;/amount&gt; </pre>

#### 4.1.4 Sample RequestTransfer Response

```

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns2:requestTransferResponse xmlns:ns2="http://external.interfaces.ers.seamless.com/">
      <return>
        <ersReference>2017102412125845102001292</ersReference>
        <resultCode>0</resultCode>
        <resultDescription>SUCCESS</resultDescription>
        <receivedAmount>
          <currency>SZL</currency>
          <value>50.00</value>
        </receivedAmount>
        <senderPrincipal>
          <principalId>
            <id>$reselelrid</id>
            <type>RESELLERID</type>
          </principalId>
          <principalName>$username</principalName>
          <accounts>
            <account>
              <accountDescription>Reseller account for reseller</accountDescription>
              <accountSpecifier>
                <accountId>$resellerid</accountId>
                <accountTypeId>RESELLER</accountTypeId>
              </accountSpecifier>
              <balance>
                <currency>SZL</currency>
                <value>250.00</value>
              </balance>
              <creditLimit>
                <currency>SZL</currency>
                <value>0.00000</value>
              </creditLimit>
            </account>
          </accounts>
          <status>Active</status>
          <msisdn/>
        </senderPrincipal>
        <receiverAccountSpecifier>
          <accountId>$resellerid2</accountId>
          <accountTypeId>RESELLER</accountTypeId>
        </receiverAccountSpecifier>
        <transferAmount>
          <currency>SZL</currency>
          <value>50.00</value>
        </transferAmount>
        <receiverPrincipal>
          <principalId>
            <id>$resellerid2</id>
            <type>RESELLER</type>
          </principalId>
          <principalName/>
          <accounts>
            <account>
              <accountSpecifier>
                <accountId>$resellerid2</accountId>
                <accountTypeId>RESELLER</accountTypeId>
              </accountSpecifier>
            </account>
          </accounts>
        </receiverPrincipal>
      </return>
    </ns2:requestTransferResponse>
  </soap:Body>
</soap:Envelope>

```

#### 4.1.5 Transfer Response Field breakdown

The response to transfer request is **transferResponse**. Breakdown of this return field is explained below:

Name	Description
resultCode	A numerical value indicating the end result of the request that was last processed. See the <u><a href="#">Result Codes</a></u> section
ersReference	Unique ERS transaction reference which is used for transfer
resultDescription	Result message of the transaction. This usually elaborates the result code
requestedTransferAmount	Requested amount to be transferred
senderPrincipal	Sender account information with account details from which the amount is deducted for transfer
receiverAccountSpecifier	The account to which the transfer amount is credited
transferAmount	Amount which is transferred
recieverPrincipal	Transfer receiver principal with details of the account to which the transfer amount is credited

## 5 Result Codes

Every web service response contains a resultCode and corresponding resultDescription fields. A resultCode of '0' indicates the request was successful. The description for each resultCode value is provided in the corresponding resultDescription field.

### 5.1 Sample unsuccessful web service response:

```
<:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns2:requestTransferResponse xmlns:ns2="http://external.interfaces.ers.seamless.com/">
      <return>
        <ersReference>2018020120320802901000016</ersReference>
        <resultCode>32</resultCode>
        <resultDescription>INVALID_TOPUP_PRINCIPAL_ID</resultDescription>
        <requestedTransferAmount>
          <currency>SZL</currency>
          <value>2</value>
        </requestedTransferAmount>
        <senderPrincipal>
          <accounts/>
        </senderPrincipal>
        <receiverPrincipal>
          <accounts/>
        </receiverPrincipal>
      </return>
    </ns2:requestTransferResponse>
  </soap:Body>
</soap:Envelope>
```

### 5.2 Result Codes:

Result Code	Result Description
0	SUCCESS
1	PENDING_APPROVAL
10	REJECTED_BUSINESS_LOGIC
11	REJECTED_AMOUNT
12	REJECTED_PAYMENT
13	REJECTED_TOPUP
20	AUTHENTICATION_FAILED
21	ACCESS_DENIED

22	INVALID_NEW_PASSWORD
29	INVALID_INITIATOR_PRINCIPAL_ID
30	INVALID_RECEIVER_PRINCIPAL_ID
31	INVALID_SENDER_PRINCIPAL_ID
32	INVALID_TRANSFER_PRINCIPAL_ID
33	INVALID_INITIATOR_PRINCIPAL_STATE
34	INVALID_RECEIVER_PRINCIPAL_STATE
35	INVALID_SENDER_PRINCIPAL_STATE
36	INVALID_TRANSFER_PRINCIPAL_STATE
37	INITIATOR_PRINCIPAL_NOT_FOUND
38	RECEIVER_PRINCIPAL_NOT_FOUND
39	SENDER_PRINCIPAL_NOT_FOUND
40	TOPUP_PRINCIPAL_NOT_FOUND
41	INVALID_PRODUCT
42	INVALID_RECEIVER_ACCOUNT_TYPE
43	INVALID_SENDER_ACCOUNT_TYPE
44	INVALID_TRANSFER_ACCOUNT_TYPE
45	RECEIVER_ACCOUNT_NOT_FOUND
46	SENDER_ACCOUNT_NOT_FOUND
47	TOPUP_ACCOUNT_NOT_FOUND
90	SYSTEM_ERROR
91	UNSUPPORTED_OPERATION
92	LICENSE_REJECTION
93	SYSTEM_BUSY
94	SERVICE_UNAVAILABLE

## 5.3 PRODUCTS

Below is a list of SKUs and their PRICES currently configured in VTU.

NOTE: This table is subject to change as and when new PRODUCTS are added or removed or PRICES REVIEW.

SKU	Price(E)	DESCRIPTION	CATEGORY
<b>DATABUNDLE</b>			
DB_5MB	2	5MB Anytime and 2.5MB Nighttime	DATA_BUNDLE
DB_15MB	5	15MB Anytime and 7.5MB Nighttime	DATA_BUNDLE
DB_25MB	10	25MB Anytime and 12.5MB Nighttime	DATA_BUNDLE
DB_30MB	20	30MB Anytime and 15MB Nighttime	DATA_BUNDLE
DB_50MB	25	50MB Anytime and 25MB Nighttime	DATA_BUNDLE
DB_75MB	30	75MB Anytime and 37.5MB Nighttime	DATA_BUNDLE
DB_100MB	35	100MB Anytime and 50MB Nighttime	DATA_BUNDLE
DB_150MB	50	150MB Anytime and 75MB Nighttime	DATA_BUNDLE
DB_300MB	90	300MB Anytime and 150MB Nighttime	DATA_BUNDLE
DB_500MB	130	500MB Anytime and 250MB Nighttime	DATA_BUNDLE
DB_700MB	150	700MB Anytime and 350MB Nighttime	DATA_BUNDLE
DB_1GB	230	1GB Anytime and 500MB Nighttime	DATA_BUNDLE
DB_2GB	280	2GB Anytime and 1GB Nighttime	DATA_BUNDLE
DB_3GB	400	3GB Anytime and 1.5GB Nighttime	DATA_BUNDLE
DB_5GB	600	5GB Anytime and 2.5GB Nighttime	DATA_BUNDLE
DB_7GB	820	7GB Anytime and 3.5GB Nighttime	DATA_BUNDLE
DB_10GB	1100	10GB Anytime and 5GB Nighttime	DATA_BUNDLE
DB_15GB	1300	15GB Anytime and 7.5GB Nighttime	DATA_BUNDLE
DB_25GB	2300	25GB Anytime and 12.5GB Nighttime	DATA_BUNDLE
<b>PACKAGES</b>			
DB_PKG1	5	Package 1	PACKAGES
DB_PKG2	10	Package 2	PACKAGES
DB_PKG3	50	Package 3	PACKAGES
DB_PKG4	195	Package 4	PACKAGES
DB_PKG5	295	Package 5	PACKAGES
DB_PKG6	495	Package 6	PACKAGES
DB_SOCIAL	240	Social Plan	SOCIAL PLAN
<b>VOUCHER</b>			
LOG10001	10	10 Voucher	VOUCHER



LOG20001	20	20 Voucher	VOUCHER
LOG50001	50	50 Voucher	VOUCHER
LOG100001	100	100 Voucher	VOUCHER
LOG200001	200	200 Voucher	VOUCHER