Ubequee

API
Documentation

Version 1.4

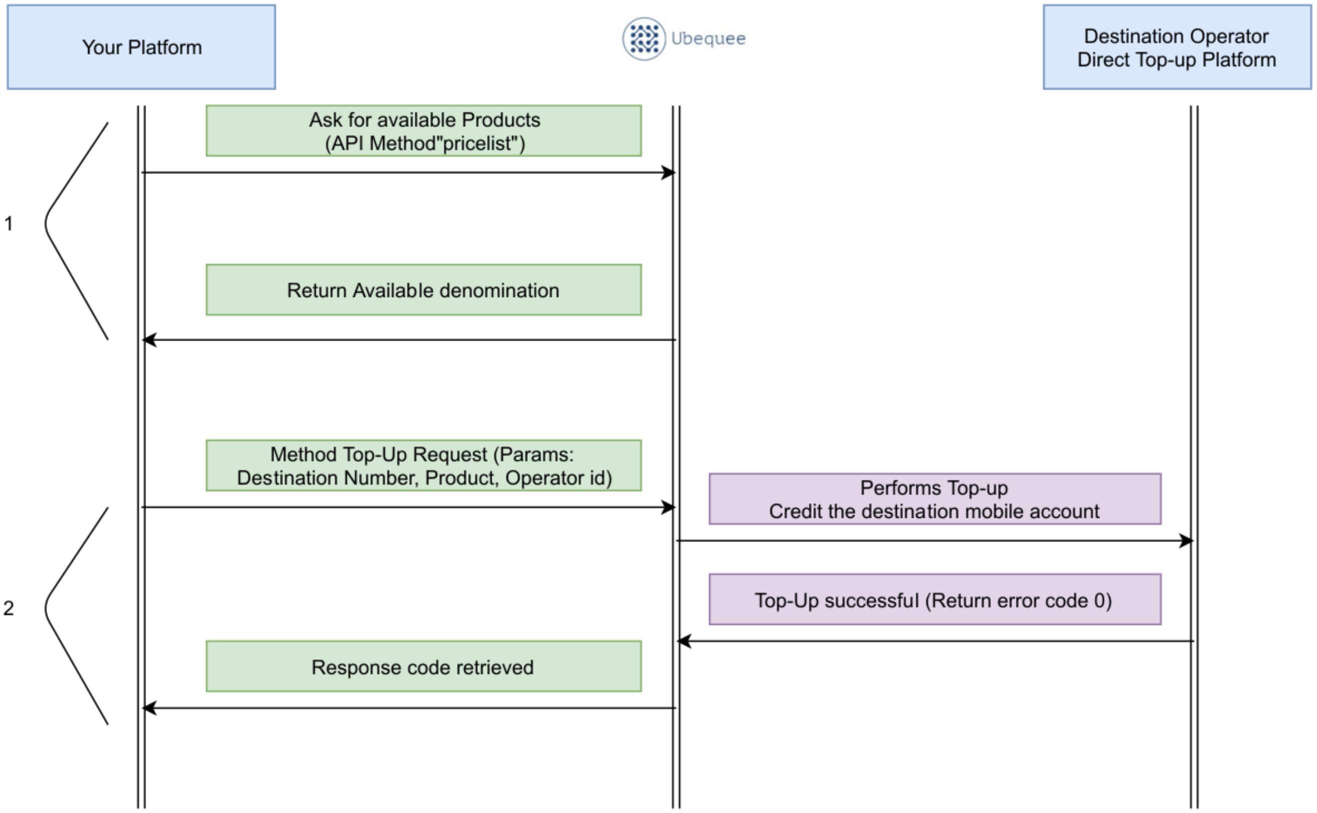
www.api-topup.com
www.ubequee.com

1 Overview

This document describes the Ubequee API protocol and its features.

1.1 TopUp Process Flow

Different functions of the API can be used to conduct a topup (see below):


Your Platform
Ubequee
Destination Operator
Direct Top-up Platform
Ask for available Products
(API Method"pricelist")
1
Return Available denomination
Method Top-Up Request (Params:
Destination Number, Product, Operator id)
Performs Top-up
Credit the destination mobile account
2
Top-Up successful (Return error code 0)
Response code retrieved

You can avoid calling the **pricelist** method if you store the id operator.
To do this, you’ll have to store on your system an exact copy of the products configured in your account. In addition, your system must be able to identify the correct destination operator based on the recipient’s mobile number. Note that calling this function returns the list of available products in less than a few seconds.

2 TCP/IP Protocol

Ubequee 's application is reachable using **HTTP over TCP/IP**.

RFC of the HTTP protocol may be found here http://www.faqs.org/rfcs/rfc1945.html.
Note: many very smart libraries and modules are available for many languages to send HTTP requests.
Most transactions are processed in less than **30 seconds**. However, as a few transactions may be delayed at the destination operator's end, your timeout between topup request and response must be set to 600 seconds. Ubequee's platform guarantees that transactions not processed within **600 seconds** will not be charged, whatever the final status (successful or not).
Our system is in **real-time**, therefore the response to a **topup** request is final. Once the TCP connection established for the previously sent request is closed, no confirmation can be received afterwards, except if you call the **trans_info** method.

3 How to send requests

Ubequee's platform is reachable via HTTPS and a specific URL. Arguments must be provided using **HTTPS POST** methods (both are valid and recognized by our system).

The URL to reach Ubequee's platform is https://api-topup.com/api/easytopup

Protocol also supports XML POST requests (the system will answer in XML if you submitted your request in XML).

3.1 Response text

You will always receive an answer in XML format.

3.2 XML POST Requests

As mentioned above, Ubequee platform is also reachable using a more common format, XML.

When sending a request, content must be formatted into XML before being posted to the URL.
NOTE: don't forget to set the content type to "text/xml".

The response will be in XML as well. All field names used to create the request and received in the response remain the same
as those used in "text/plain" format.

4 Authentication and Security

Ubequee provides login and portal password at the creation of the merchant account. You may create sub accounts via the web portal https://api-topup.com/. You are required to retrieve your API Token via the web portal.
For each request, the client must generate a key, which must be an integer and unique.

For each request, the following fields must be populated for authentication purposes:

Parameter	Type	Description
login	Alphanumeric String	Login provided by Ubequee
key	Integer	Key generated by the client. The transaction will be refused if the key has already been used before.
md5	Alphanumeric String	MD5 sum of the hex encoded concatenation of login, token, key

NOTE: We strongly advise that you generate and manage the key on your side. A good key would be a time stamp with milliseconds if you are using a unique server.
If you are using multiple servers, the easiest solution may be to use a common sequence shared via a database to ensure that all the servers are synchronized.
You can also set up several topup accounts via the web interface allowing you to have distinct sequences (one per server for instance).

Example

Login and token retrieved from Ubequee portal (fake login/token below, ask for your own account credentials):
login: **client**
token: **pass99**

Key generated by Client side for each transaction.
Calculation of the MD5 hash:

md5 = md5_hex(login.token.key);
Key = 1, MD5 = 2f922ef79253f93d0fc05d0ac0b0d100
Key = 2, MD5 = 43dd325a272140a9f47a3be015eb8129
Key = 3, MD5 = cc98cf7d902047101be5635f6af3877e
Key = 4, MD5 = b4a9d3a95ac0efb73d840bd0dbbe1d1c
Key = 5, MD5 = a43cff6e5d057f84b1ae99e64245bff8

As we can see in the samples, only the key is modified each time from 1 to 5 but the MD5 result is really different.

4.1 Retrieve Your Api Token

To activate access to your API account, you need to **Generate Token** and will also need to **whitelist** the server **IP** address that will access the API.

Instruction to enable your API Token:

1. Click on your username at the top of the menu
2. Select the tab “**Settings**”
3. Generate a new token or copy the existing one
4. Press Save button

Instruction to add|remove whitelist IP address:

1. Click on your username at the top of the menu
2. Select the tab “**Settings**”
3. Add or remove IP address from the whitelist
4. Press Save button

5 Testing Account and URL

Ubequee platform will create an account and you will be able to retrieve your token via the portal with the login name provided.
(Please email us to request an account if this has not already been done)

The URL to reach our topup platform is: https://api-topup.com/api/easytopup
To test how your system reacts to the most common errors returned by our system in the topup method response, “fake” phone numbers in Indonesia have been set up with available products of IDR 5000, 10000, 20000, 50000 and 100000:

- 628123456710 will return the error 0 for PIN topup (successful transaction)
- 628123456770 will return the error 0 for PINless topup (successful transaction)
- 628123456780 will return the error 204 (destination number is not a valid prepaid phone number)
- 628123456781 will return the error 301 (input value out of range or invalid product)
- 628123456790 will return the error 214 (transaction refused by the operator)
- 628123456798 will return the error 998 (system not available, please retry later)
- 628123456799 will return the error 999 (unknown error, please contact support)

6 Methods overview

Ubequee’s API currently supports 8 different methods identified by a specific value of the parameter “action”. Each request must be submitted with at least all the mandatory arguments of the associated action or the system will reject

the request. Optional arguments can be added depending on the action.

Method (value parameter “action”)	Description
ping	This method is used to ping the api.
topup	This method is used to conduct a topup.
check_wallet	This method is used to retrieve the balance in your wallet. It is recommend to use this function not more than 24 times per day.
trans_info	This method is used to retrieve all available information on a specific transaction.
pricelist	This method is used to retrieve related pricelist configured in your account.
simulation	This method is used to simulate a top-up. It takes the same arguments as topup action. It does not perform a real top-up to the destination number.

7 Methods Details

Here are all the methods available

7.1 Method Ping

This method can be used when you want to test the connection and your account credentials.

Request Details

Field	Type	Required	Notes
action	Alphanumeric String	Y	Must be set to “ping”
login	Alphanumeric String	Y	Your account login name. Refer to section 4 for more details on authentication mechanism.
md5	Alphanumeric String	Y	MD5 sum of the hex encoded concatenation of login, token and key. Refer to section 4 for more details on authentication mechanism.
key	Integer	Y	Key generated by the client. The transaction will be refused if the key has been used before. We advise to use an auto increment function. Refer to section 4 for more details on authentication mechanism.

Response Details

Field	Type	Notes
error_code	Integer	Defines the error code of a transaction. Please refer to section 8 for the list of all error codes returned by Ubequee’s API.
error_txt	Alphanumeric String	Returns the human readable value of “error_code”.
info_txt	Alphanumeric String	Should contain “**pong**” if the transaction is successful.
authentication_key	Integer	Authentication key as it has been given by the client in the request.

Example Request Send

```
<xml>
      <login>test</login>
      <key>1327027869</key>
      <md5>md5</md5>
      <action>ping</action>
</xml>
```

Example Response Received

```
<xml>
      <info_txt>pong</info_txt>
      <authentication_key>1327027869</authentication_key>
      <error_code>0</error_code>
      <error_txt>Transaction successful</error_txt>
</xml>
```

7.2 Method Check Wallet

This function is used to retrieve the credit balance in your Ubequee's account.

Request Details

Field	Type	Required	Notes
action	Alphanumeric String	Y	Must be set to "check_wallet"
login	Alphanumeric String	Y	Your account login name.
md5	Alphanumeric String	Y	MD5 sum of the hex encoded concatenation of login, token and key. Refer to section 4 for more details on authentication mechanism.
key	Integer	Y	Key generated by the client. The transaction will be refused if the key has been used before. We advise to use an auto increment function. Refer to section 4 for more details on authentication mechanism.

Response Details

Field	Type	Notes
error_code	Integer	Defines the error code of a transaction. Please refer to section 8 for the list of all error codes returned by Ubequee's API.
error_txt	Alphanumeric String	Returns the human readable value of "error_code".
wallet	Numeric	If you are a Merchant, Balance reflects the remaining amount in your account. If you are a Retailer, Balance reflects the remaining limit for the day(Retail Price is deducted from limit).
login	Alphanumeric String	Your account login name.
type	Alphanumeric String	Defines your account type (Prepaid or Postpaid).
currency	Alphanumeric String	Your account currency
authentication_key	Integer	Authentication key as it has been given by the client in the request.

Error Codes

Field	Description
0	Transaction successful
919	All needed argument not received
921	Wrong MD5 encoding
923	Key or MD5 value cannot be empty
925	Key already used or invalid key value
926	Account not active. Please contact your Account Manager
995	Account not found. Please enter and use a correct login

Example Request Send

```
<xml>
      <login>test</login>
      <key>123456789</key>
      <md5>md5</md5>
      <action>check_wallet</action>
</xml>
```

Example Response Prepaid

```
<xml>
    <type>Prepaid</type>
    <login>test1</login>
    <currency>EUR</currency>
    <wallet>1.49</wallet>
    <authentication_key>123456789</authentication_key>
    <error_code>0</error_code>
    <error_txt>Transaction successful</error_txt>
</xml>
```

Example Response Postpaid

```
<xml>
    <type>Postpaid</type>
    <login>test2</login>
    <currency>EUR</currency>
    <wallet>149.98</wallet>
    <authentication_key>123456789</authentication_key>
    <error_code>0</error_code>
    <error_txt>Transaction successful</error_txt>
</xml>
```

7.3 Simulation

This method is used to simulate a top-up. It takes the same arguments as topup action. It does not perform a real top-up to the destination number.

This action will test if the destination number is in our numbering plan, and if the product specified is valid.

- It DOES NOT verify if the destination number has expired for instance as the request is not submitted to the operator. Thus, a simulation can be successful, while a real top-up (request sent to the operators to perform both debit and top-up) can fail.
- API will respond according to product type for the destination for Pin-less and Open Range only.
- Response for PIN based product is the same as Pin-less
- Operator ID should be passed along with the simulation request to get back the correct simulation response.

*Note that this action will provide you with some information (product available in your account) but is not part of the top-up cycle. You should perform top-up (via "**topup" methods**) without making any simulations before.*

7.4 Top Up

This method is used to recharge a destination number with a specified denomination (“product” field). This is the API’s most important action as it is required when sending a topup to a prepaid account phone number in a live environment.

The operator ID (<operatorid>) is obligatory to be passed in order to enforce the system to choose the correct operator or product selected.

Request Details

Field	Type	Required	Notes
action	Alphanumeric String	Y	Must be set to “topup”
login	Alphanumeric String	Y	Your account login name.
md5	Alphanumeric String	Y	MD5 sum of the hex encoded concatenation of login, token and key. Refer to section 4 for more details on authentication mechanism.
key	Integer	Y	Key generated by the client. The transaction will be refused if the key has been used before. We advise to use an auto increment function. Refer to section 4 for more details on authentication mechanism.
msisdn	Alphanumeric String	Y	The international phone number or name of the user requesting to credit a phone number. The format must contain the country code, and will be valid with or without the ‘+’ or ‘00’ placed before it. For example: “6012345678” or “+6012345678” or “006012345678” (Malaysia) or “John” are all valid. This field must not be left empty.
destination_msisdn	Numeric String	Y	This is the destination phone number that will be credited with the amount transferred. Format is similar to “msisdn” and restricted to international phone number only.
product	Alphanumeric String	Y	This field is used to define the remote product (often, the same as the amount in destination currency) to use in the request.
operatorid	Integer	Y	It defines the operator id of the destination MSISDN that must be used when treating the request.

Response Details

Our system can return two possible response formats depending on the type of recharges performed: Pin-Less or Pin-Based. Please refer to prices listed on your account to know the different types of requested products.

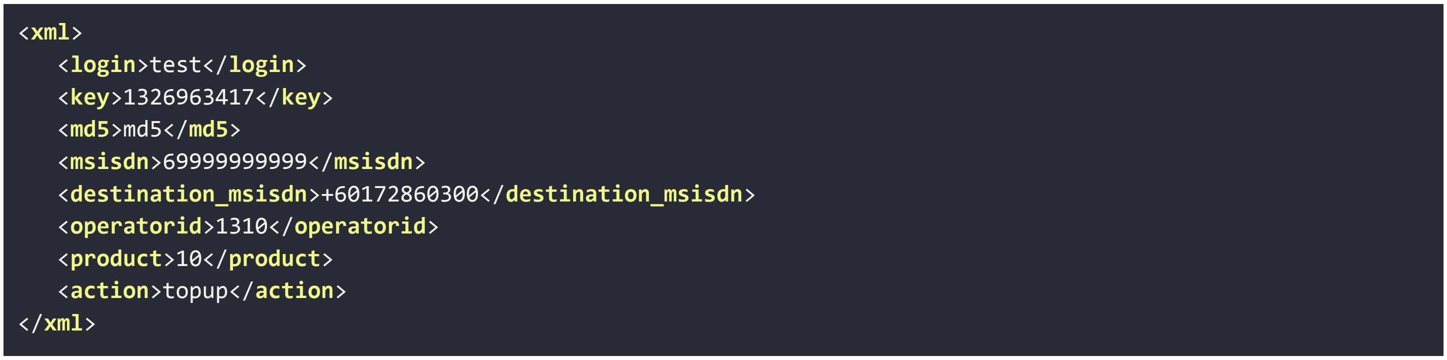
1. Pin-Less

Field	Type	Notes
error_code	Integer	Defines the error code of a transaction. Please refer to section 8 for the list of all error codes returned by Ubequee's API.
error_txt	Alphanumeric String	Returns the human readable value of "error_code".
msisdn	Alphanumeric String	MSISDN (phone number) of the sender (the person who requested the transfer)..
destination_msisdn	Numeric String	Destination MSISDN of the recipient
originating currency	Alphanumeric String	Currency used in the account that performed the transaction request.
destination_currency	Alphanumeric String	Currency used in the destination country of the destination number (where the transfer is going to).
product_requested	Float	Returns the value in the "product" tag the client requested.
actual_product_sent	Float	Returns the real sent amount, this information will be the same as "output_value" in case of successful transfer in most cases. It can be different in case of internal product mapping on our side (India mainly). This is the product that we requested to the operator.
authentication_key	Integer	Authentication key as it has been given by the client in the request. As this key has to be different each time, it may be used as a unique ID on the client side.
wholesale_price	Numeric	Return the price of this product sold to client in the account's currency
retail_price	Numeric	Return the retail price of this product the client is configured to sell
service_fee	Numeric	Return the service fee of this product the client is configured to sell
transactionid	Integer	Defines the ID of the transaction.
operator	Alphanumeric String	Destination operator name
operatorid	Numeric	Destination operator ID
country	Alphanumeric String	Destination country
countryid	Numeric	Destination country ID
return_timestamp	Date	Timestamp recorded for this transaction e.g "**2013-03-18 00:00:00**"

2. Pin-Based

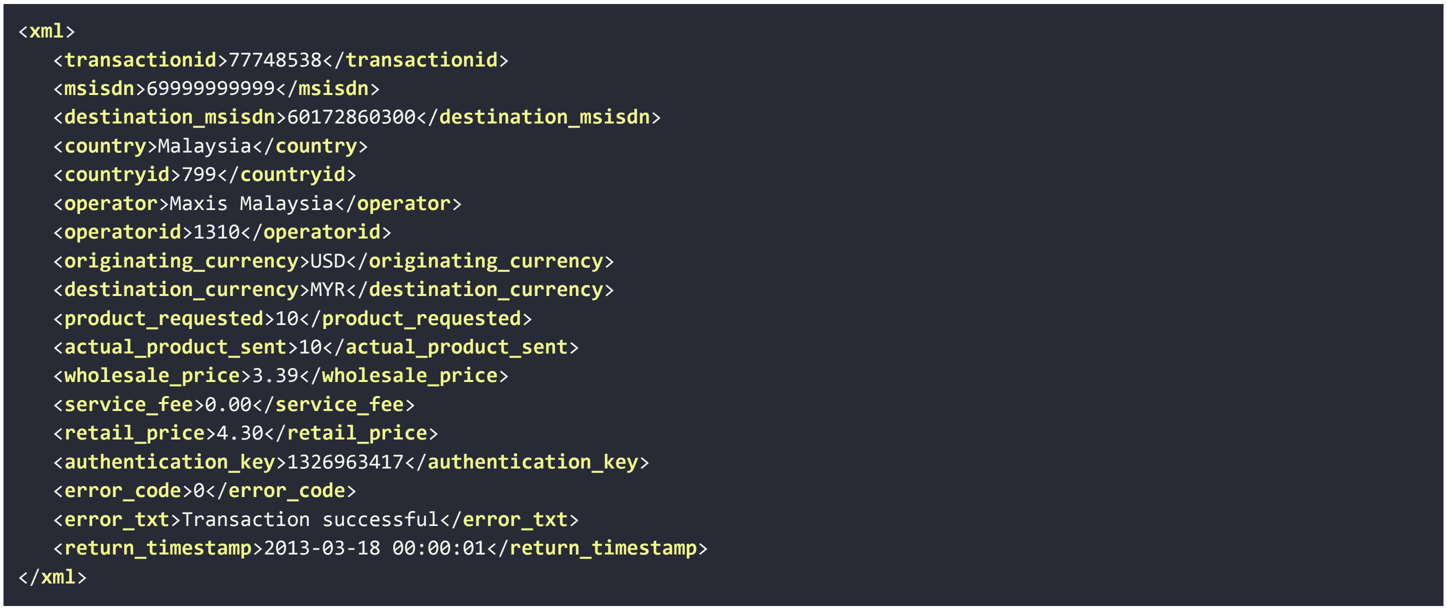
Field	Type	Notes
error_code	Integer	Defines the error code of a transaction. Please refer to section 8 for the list of all error codes returned by Ubequee's API.
error_txt	Alphanumeric String	Returns the human readable value of "error_code".
msisdn	Alphanumeric String	MSISDN (phone number) of the sender (the person who requested the transfer)..
destination_msisdn	Numeric String	Destination MSISDN of the recipient
originating currency	Alphanumeric String	Currency used in the account that performed the transaction request.
destination_currency	Alphanumeric String	Currency used in the destination country of the destination number (where the transfer is going to).
product_requested	Float	Returns the value in the "product" tag the client requested.
actual_product_sent	Float	Returns the real sent amount, this information will be the same as "output_value" in case of successful transfer in most cases. It can be different in case of internal product mapping on our side (India mainly). This is the product that we requested to the operator.
authentication_key	Integer	Authentication key as it has been given by the client in the request. As this key has to be different each time, it may be used as a unique ID on the client side.
wholesale_price	Numeric	Return the price of this product sold to client in the account's currency
retail_price	Numeric	Return the retail price of this product the client is configured to sell
service_fee	Numeric	Return the service fee of this product the client is configured to sell
transactionid	Integer	Defines the ID of the transaction.
operator	Alphanumeric String	Destination operator name
operatorid	Numeric	Destination operator ID
country	Alphanumeric String	Destination country
countryid	Numeric	Destination country ID
pin_based	String	Type of product returned ("**Yes**", default "**No**" if not set). Appears only in case of Pin-based transactions.
pin_validity	String	Validity information of the PIN
pin_code	Alphanumeric String	Code of the PIN
pin_ivr	Integer	IVR number of this PIN
pin_serial	Alphanumeric String	Serial number of the PIN
pin_value	Integer	Value of the PIN returned
return_timestamp	Date	Timestamp recorded for this transaction e.g "**2013-03-18 00:00:00**"

1. Example Pin-Less



```xml
<xml>
   <login>test</login>
   <key>1326963417</key>
   <md5>md5</md5>
   <msisdn>69999999999</msisdn>
   <destination_msisdn>+60172860300</destination_msisdn>
   <operatorid>1310</operatorid>
   <product>10</product>
   <action>topup</action>
</xml>
```

Response received Pin-Less



```xml
<xml>
   <transactionid>77748538</transactionid>
   <msisdn>69999999999</msisdn>
   <destination_msisdn>60172860300</destination_msisdn>
   <country>Malaysia</country>
   <countryid>799</countryid>
   <operator>Maxis Malaysia</operator>
   <operatorid>1310</operatorid>
   <originating_currency>USD</originating_currency>
   <destination_currency>MYR</destination_currency>
   <product_requested>10</product_requested>
   <actual_product_sent>10</actual_product_sent>
   <wholesale_price>3.39</wholesale_price>
   <service_fee>0.00</service_fee>
   <retail_price>4.30</retail_price>
   <authentication_key>1326963417</authentication_key>
   <error_code>0</error_code>
   <error_txt>Transaction successful</error_txt>
   <return_timestamp>2013-03-18 00:00:01</return_timestamp>
</xml>
```

2. Example Pin-Based

```
<xml>
   <login>test</login>
   <key>1326964272</key>
   <md5>md5</md5>
   <msisdn>69999999999</msisdn>
   <destination_msisdn>00009779742045269</destination_msisdn>
   <operatorid>1310</operatorid>
   <product>200</product>
   <delivered_amount_info>1</delivered_amount_info>
   <action>topup</action>
</xml>
```

Response received Pin-Based

```
<xml>
   <transactionid>77749654</transactionid>
   <msisdn>69999999999</msisdn>
   <destination_msisdn>9779742045269</destination_msisdn>
   <country>Nepal</country>
   <countryid>819</countryid>
   <operator>NTC CDMA Nepal</operator>
   <operatorid>1310</operatorid>
   <originating_currency>USD</originating_currency>
   <destination_currency>NPR</destination_currency>
   <product_requested>200</product_requested>
   <actual_product_sent>200</actual_product_sent>
   <wholesale_price>2.58</wholesale_price>
   <service_fee>0.00</service_fee>
   <retail_price>3.30</retail_price>
   <pin_based>yes</pin_based>
   <pin_value>200</pin_value>
   <pin_code>12345678901234</pin_code>
   <pin_ivr>1415</pin_ivr>
   <pin_serial>123456789123456789</pin_serial>
   <pin_validity>2014-01-18T00:00:00+05:45</pin_validity>
   <authentication_key>1326964272</authentication_key>
   <error_code>0</error_code>
   <error_txt>Transaction successful</error_txt>
</xml>
```

7.5 Trans info

This method can be used to retrieve available information on a specific transaction.
Please note that values of “input_value” and “debit_amount_validated” are rounded to 2 digits after the comma but are the same as the values returned in the fields “input_value” and “validated_input_value” of the “topup” method response.

Request Details

Field	Type	Required	Notes
action	Alphanumeric String	Y	Must be set to “topup”
login	Alphanumeric String	Y	Your account login name.
md5	Alphanumeric String	Y	MD5 sum of the hex encoded concatenation of login, token and key. Refer to section 4 for more details on authentication mechanism.
key	Integer	Y	Key generated by the client. The transaction will be refused if the key has been used before. We advise to use an auto increment function. Refer to section 4 for more details on authentication mechanism.
transactionid	Integer	Y	ID of the transaction you want to get information on.

Response Details

Field	Type	Notes
error_code	Integer	Defines the error code of a transaction. Please refer to section 8 for the list of all error codes returned by Ubequee's API.
error_txt	Alphanumeric String	Returns the human readable value of "error_code".
msisdn	Alphanumeric String	MSISDN (phone number) of the sender (the person who requested the transfer).
destination_msisdn	Numeric String	Destination MSISDN of the recipient
originating currency	Alphanumeric String	Currency used in the account that performed the transaction request.
destination_currency	Alphanumeric String	Currency used in the destination country of the destination number (where the transfer is going to).
product_requested	Float	Returns the value in the "product" tag the client requested.
actual_product_sent	Float	Returns the real sent amount, this information will be the same as "output_value" in case of successful transfer in most cases. It can be different in case of internal product mapping on our side (India mainly). This is the product that we requested to the operator.
authentication_key	Integer	Authentication key as it has been given by the client in the request. As this key has to be different each time, it may be used as a unique ID on the client side.
transaction_error_code	Integer	Defines the error code of the transaction ID requested.
transaction_error_txt	Alphanumeric String	Defines the human readable value of the error code of the transaction ID requested.
transaction_authentication_key	Integer	Authentication key used for this transaction
date	String	Date when the transaction requested was performed (GMT).
wholesale_price	Numeric	Return the price of this product sold to client in the account's currency
retail_price	Numeric	Return the retail price of this product the client is configured to sell
service_fee	Numeric	Return the service fee of this product the client is configured to sell
operator	Alphanumeric String	Destination operator name
operatorid	Numeric	Destination operator ID
country	Alphanumeric String	Destination country
countryid	Numeric	Destination country ID
pin_based	String	Type of product returned ("**Yes**", default "**No**" if not set). Appears only in case of Pin-based transactions.
pin_validity	String	Validity information of the PIN
pin_code	Alphanumeric String	Code of the PIN
pin_ivr	Integer	IVR number of this PIN
pin_serial	Alphanumeric String	Serial number of the PIN
pin_value	Integer	Value of the PIN returned
return_timestamp	Date	Timestamp recorded for this transaction e.g "**2021-10-28 10:10:05**"

Error Code

Field	Description	Notes
0	Transaction successful	
401	Transaction ID not found or was not made by your account	

Example Request Sent

```
<xml>
   <login>your_login</login>
   <key>1326967613</key>
   <md5>md5</md5>
   <transactionid>77748538</transactionid>
   <action>trans_info</action>
</xml>
```

Response received

```
<xml>
   <transactionid>77748538</transactionid>
   <msisdn>69999999999</msisdn>
   <destination_msisdn>60172860300</destination_msisdn>
   <transaction_authentication_key>1326963417</transaction_authentication_key>
   <transaction_error_code>0</transaction_error_code>
   <transaction_error_txt>Transaction successful</transaction_error_txt>
   <country>Malaysia</country>
   <countryid>799</countryid>
   <operator>Maxis Malaysia</operator>
   <operatorid>1311</operatorid>
   <product_requested>10</product_requested>
   <actual_product_sent>10</actual_product_sent>
   <wholesale_price>3.39</wholesale_price>
   <retail_price>4.30</retail_price>
   <service_fee> 1.2 </service_fee>
   <date>2021-01-19 08:57:21</date>
   <originating_currency>USD</originating_currency>
   <destination_currency>MYR</destination_currency>
   <authentication_key>1326967613</authentication_key>
   <error_code>0</error_code>
   <error_txt>Transaction successful</error_txt>
</xml>
```

7.6 Price List

This method is used to retrieve coverage and pricelist offered to you.

Request Details

Field	Type	Required	Notes
action	Alphanumeric String	Y	Must be set to “topup”
login	Alphanumeric String	Y	Your account login name.
md5	Alphanumeric String	Y	MD5 sum of the hex encoded concatenation of login, token and key. Refer to section 4 for more details on authentication mechanism.
key	Integer	Y	Key generated by the client. The transaction will be refused if the key has been used before. We advise to use an auto increment function. Refer to section 4 for more details on authentication mechanism.
info_type	Alphanumeric String	Y	1. “**countries**” : Returns a list of all countries offered to you 2. “**country**” : Returns a list of operators in the country 3. “**operator**” : Returns a list of wholesale and retail price for the operator
content	Integer	N	1. Not used if info_type = “**countries**” 2. countryid of the requested country if info_type = “**country**” 3. operatorid of the requested operator if info_type = “**operator**”

Response Details

Field	Type	Notes
error_code	Integer	Defines the error code of a transaction. 0 or 919
error_txt	Alphanumeric String	Returns the human readable value of “error_code”.
authentication_key	Integer	Authentication key as it has been given by the client in the request. As this key has to be different each time, it may be used as a unique ID on the client side.
country	Alphanumeric String	Country of the destination operator
countryid	Numeric	Unique ID of the destination country
destination_currency	Alphanumeric String	Currency in the country of destination
payment_currency	Alphanumeric String	The payment currency
operator	Alphanumeric String	Operator of the destination MSISDN
operatorid	Integer	Unique ID of the destination operator
product_list	String	Returns the list of all available products configured for the current account. Products are separated by a comma.
retail_price_list	String	Returns the list of all final retail prices configured for the current topup account. Prices are separated by a comma and in the currency of the topup account.
wholesale_price_list	String	Returns the wholesale price list of all available products configured for the current account. Prices are separated by a comma and in the currency of the topup account.
service_fee_list	Date	Returns the list of all fee prices configured for the current topup account

Example Request Sent

info_type = countries

```
<xml>
   <login>your_login</login>
   <key>1326965217</key>
   <md5>md5</md5>
   <info_type>countries</info_type>
   <action>pricelist</action>
</xml>
```

Response received

```
<xml>
   <country>Russia,Senegal,India,Bolivia,United States,Morocco,Sri Lanka,El Salvador,United
States,Indonesia,Poland,Burundi,Grenada,India,India,Pakistan,Turks and Caicos,United States,Egypt,Egypt,Fiji,Brazil,St
Lucia,Antigua and Barbuda,Pakistan,Montserrat,Dominican
Republic,Guatemala,Mexico,Peru,India,Ghana,Venezuela,India,India,India,Indonesia,Russia,Afghanistan,India,Russia,Russia,
India,Bolivia,India,Bangladesh,Guatemala,Russia,Tunisia,Brazil,Russia,Honduras,Honduras,Russia,Ukraine,Colombia,St
Lucia,India,India,Poland,Laos,St Kitts and Nevis,Dominican Republic,Nepal,Poland,Ghana,Equatorial
Guinea,Afghanistan,Anguilla,India,Afghanistan,Vietnam,Jamaica,Honduras,St Kitts and
Nevis,Jordan,Vietnam,Kazakhstan,Indonesia,Malaysia,Ghana,Pakistan,Russia,Peru,Afghanistan,Venezuela,Kazakhstan,Haiti,
Palestine,Bangladesh,Russia,Yemen,Barbados,India,Honduras,Sri Lanka,Turkey,Philippines,Russia,Russia,Sri
Lanka,India,Russia,Mexico,Benin,Tunisia,Mexico,Senegal,Nepal,India,Russia,Pakistan,Sri
Lanka,Afghanistan,Chile,India,China,Grenada,Central African
Republic,India,Philippines,India,Cambodia,Mali,Thailand,Poland,Cyprus,Russia,United States,Congo,Indonesia,Antigua and
Barbuda,Barbados,India,Nicaragua,Vietnam,Vietnam,United States,Jamaica,Dominican
Republic,Philippines,Nigeria,Anguilla,Guinea
Bissau,Romania,Montserrat,Indonesia,India,Colombia,Pakistan,India,Vietnam,Tanzania,India,Ivory
Coast,India,Portugal,India,India,Colombia,India,Sudan,Nicaragua,Guyana,Cameroon,Indonesia,India,Sri
Lanka,Madagascar,Honduras,Dominican
Republic,Portugal,India,India,Haiti,Bangladesh,Malaysia,Nigeria,Ecuador,Nigeria,Cambodia,El
Salvador,Nigeria,Mexico,Ghana,India,Pakistan,Indonesia,Indonesia,India,United States,Dominican Republic,St Vincent
Grenadines,Fiji,Ghana,Panama,Tanzania,Jamaica,Malaysia,Haiti,Trinidad and Tobago,Mexico,Costa
Rica,Russia,Somalia,Nigeria,Guatemala,Portugal,India,Mexico,Ukraine,Russia,Poland,Kenya,India,Dominica,Ukraine,Guyana,
India,Vietnam,Russia,Russia,Liberia,Ivory Coast,Central African Republic,Guatemala,China,Nigeria,Russia,United
States,Yemen,St Lucia,Rwanda,Kenya,Russia,Cambodia,India,Russia,Poland,Guatemala,Guatemala,Niger,Mexico,El
Salvador,Jamaica,Bolivia,Nepal,United States,El Salvador,Russia,Indonesia,Honduras,Russia,Democratic Republic of the
Congo,Russia,Haiti,United States,Venezuela,India,India,India,Pakistan,Pakistan,St Vincent Grenadines,United
States,Mexico,Kazakhstan,Pakistan,Iraq,Dominican Republic,Afghanistan,India,Venezuela,Russia,Dominican
Republic,Poland,Ukraine,Uganda,Cuba,Bangladesh,Bolivia,Russia,Peru,Nicaragua,Russia,Trinidad and
Tobago,India,Guinea,Ghana,Russia,India,Poland,Russia,Zimbabwe,Russia,Ghana,Russia,India,Nicaragua,Vietnam,Russia,Cameroo
n,Cayman Islands,Dominica,Brazil,Peru,Sri Lanka,Romania,Indonesia,Indonesia,India</country>
   <countryid>848,859,766,686,899,813,871,730,899,767,841,698,753,766,766,832,894,899,729,729,737,691,872,668,832,917,7
25,756,808,839,766,748,907,766,766,766,767,848,661,766,848,848,766,686,766,678,756,848,891,691,848,762,762,848,901,
710,872,766,766,841,785,918,725,819,841,748,731,661,916,766,661,908,774,762,918,776,908,778,767,799,748,832,848,839,661
,907,778,761,834,678,848,911,679,766,762,871,892,840,848,848,871,766,848,808,683,891,808,859,819,766,848,832,871,661,
707,766,708,753,705,766,840,766,699,801,884,841,718,848,899,913,767,668,679,766,824,908,908,899,774,725,840,826,916,
758,847,917,767,766,710,832,766,908,882,766,773,766,843,766,766,710,766,874,824,760,700,767,766,871,797,762,725,843,7
66,766,761,678,799,826,727,826,699,730,826,808,748,766,832,767,767,766,899,725,873,737,748,836,882,774,799,761,889,8
08,715,848,867,826,756,843,766,808,901,848,841,779,766,724,901,760,766,908,848,848,789,773,705,756,708,826,848,899,
911,872,849,779,848,699,766,848,841,756,756,825,808,730,774,686,819,899,730,848,767,762,848,712,848,761,899,907,766,7
66,766,832,832,873,899,808,778,832,769,725,661,766,907,848,725,841,901,900,717,678,686,848,839,824,848,889,766,757,7
48,848,766,841,848,915,848,748,848,766,824,908,848,700,704,724,691,839,871,847,767,767,766</countryid>
   <authentication_key>1337662098</authentication_key>
   <error_code>0</error_code>
   <error_txt>Transaction successful</error_txt>
</xml>
```

Example Request Sent

info_type = country

```
<xml>
   <login>your_login</login>
   <key>1326965217</key>
   <md5>md5</md5>
   <info_type>country</info_type>
   <content>767</content>
   <action>pricelist</action>
</xml>
```

Response received

```
<xml>
   <country>Indonesia</country>
   <countryid>767</countryid>
   <operator>Indosat Starone Indonesia,Telkom Flexi Indonesia,Indosat IM3 Indonesia,AAA-TESTING Indonesia,Esia Bakrie Telecom Indonesia,Indosat Mentari Indonesia,Telkomsel Simpati Indonesia,Three Telecom Indonesia,Telkomsel KartuAS Indonesia,Excelcom Indonesia,Axis Indonesia</operator>
   <operatorid>1320,1322,1313,1310,1326,1312,1324,1327,1325,215,1411</operatorid>
   <authentication_key>1337662386</authentication_key>
   <error_code>0</error_code>
   <error_txt>Transaction successful</error_txt>
</xml>
```

Example Request Sent

info_type = operator

```
<xml>
<login>your_login</login>
<key>1326965217</key>
<md5>md5</md5>
<info_type>operator</info_type>
<content>1310</content>
<return_service_fee>1</return_service_fee>
<action>pricelist</action>
</xml>
```

Response received

```
<xml>
   <country>Indonesia</country>
   <countryid>767</countryid>
   <operator>AAA-TESTING Indonesia</operator>
   <operatorid>1310</operatorid>
   <destination_currency>IDR</destination_currency>
   <payment_currency>EUR</payment_currency>
   <product_list>5000,10000,20000,50000,100000</product_list>
   <retail_price_list>0.52,1.05,2.10,6.30,14.18</retail_price_list>
   <service_fee_list>0,0,0,0,0</service_fee_list>
   <wholesale_price_list>0.50,1.00,2.00,6.00,13.50</wholesale_price_list>
   <distributor_margin>5,5,5,5,5</distributor_margin>
   <distributor_margin_abs>0.02,0.05,0.10,0.30,0.68</distributor_margin_abs>
   <authentication_key>1337662386</authentication_key>
   <error_code>0</error_code>
   <error_txt>Transaction successful</error_txt>
```

```
</xml>
```

8 Error Codes

Exhaustive list of error codes returned by Orange ODM platform

Field	Description	Notes
0	Transaction successful	
101	Destination MSISDN out of range	
104	MSISDN in black list	MSISDN refers to destination MSISDN
105	Not enough credit on your account	
200	Transaction canceled by Customer	
202	Transaction canceled	
203	Transaction incomplete	
204	Destination Account is not prepaid	
207	Transaction amount limit exceeded	Limit refers to Destination Limitation
213	Duplicated transaction	
214	Topup refused	
215	Service to this destination operator is temporarily unavailable	
216	Destination number not activated	
217	Destination number expired	
218	Request timeout	
219	Key does not exist	
221	Fraud suspicion	
224	Invalid length of destination MSISDN	
230	Recipient reached maximum topup number	
231	Recipient reached maximum topup amount	
233	Account reached maximum topup amount	
241	Account reached maximum daily topup amount	
301	input_value out of range or invalid product	
310	Denomination blocked	
320	Requested amount out of range	
321	Requested currency not allowed for this account	

401	transactionid not found	
900	Not enough information to process the topup	
901	Invalid action	
904	Invalid input_value	
907	Invalid transaction ID	
908	Account not configured for this service	
919	All needed argument not received	Check column “**Required**” of the methods
921	Wrong key encoding	
922	Originating IP not allowed	
923	Key already used or invalid key value	Key must be unique and greater than the last key used.
998	System not available. Please retry later	
999	Unknown error. Please contact support	

9 Sample Code in Java

This class generates the MD5 value in JAVA language.

```java
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import java.util.GregorianCalendar;
class HelloWorldApp {
    public static void main(String[] args) {
        String md5_string = getKeyedDigest("your_login", "your_ token", "your_key");
        System.out.println(md5_string);
    }
    public static String getKeyedDigest(String login, String token, String key) {
        try {
            String temp = login + token + key;
            MessageDigest md5 = MessageDigest.getInstance("MD5");
            byte[] bytes = md5.digest(temp.getBytes());
            return byteToHex(bytes);
        } catch (NoSuchAlgorithmExceptione) {
            System.out.println("oops");
        }
        return null;
    }
    public static String byteToHex(byte[] bits) {
        if (bits == null) {
            return null;
        }
        StringBuffer hex = new StringBuffer(bits.length * 2);
        for (int i = 0; i < bits.length; i++) {
            if (((int) bits[i] & 0xff) < 0x10) {
                hex.append("0");
            }
            hex.append(Integer.toString((int) bits[i] & 0xff, 16));
        }
        return hex.toString();
    }
}
```

10 Sample Code in PHP

```
<?php
//Sample code to send request to API
// Orange ODM account credentials
$login = "your_login";
$token = "your_ token";
// MD5 calculation
$key = time();
$md5 = md5($login . $token . $key);
// Orange ODM Platform
$url = "https://credittransferdistributor.orange.com/cgi-bin/topup";
//Test data
$destination_msisdn = "+628123456770";
?>
Check Wallet:
<form name="wallet" method="post" action="<?php echo $url; ?>" >
<input type="text" name="login" value="<?php echo $login ?>" >
<input type="hidden" name="key" value="<?php echo $key ?>" >
<input type="hidden" name="md5" value="<?php echo $md5 ?>" >
<input type="hidden" name="action" value="check_wallet" >
<input type="submit" value="submit" />
</form>
Check a destination number or get products for a number:
<form name="recipient" method="post" action="<?php echo $url ?>" >
<input type="hidden" name="login" value="<?php echo $login ?>" >
<input type="hidden" name="key" value="<?php echo $key ?>" >
<input type="hidden" name="md5" value="<?php echo $md5 ?>" >
<input type="text" name="destination_msisdn" value="<?php echo $destination_msisdn ?>" >
<input type="hidden" name="action" value="msisdn_info" >
<input type="submit" value="submit" />
</form>
Send a Topup:
<form name="topup" method="post" action="<?php echo $url ?>" >
<input type="hidden" name="login" value="<?php echo $login ?>" >
<input type="hidden" name="key" value="<?php echo $key ?>" >
<input type="hidden" name="md5" value="<?php echo $md5 ?>" >
<input type="hidden" name="delivered_amount_info" value="1" >
<input type="text" name="destination_msisdn" value="<?php echo $destination_msisdn ?>" >
<input type="text" name="msisdn" value="<?php echo $msisdn ?>" >
<input type="text" name="product" value="<?php echo $product ?>" >
<input type="hidden" name="action" value="topup" >
<input type="submit" value="submit" />
</form>
```