

2014

# ImpalaPay Messaging & Transaction System (IMTS) - Airtel Corridor

A Product of ImpalaPay Ltd



*ImpalaPay*

[www.impalapay.com](http://www.impalapay.com)  
<http://airtel.impalapay.com>  
[info@impalapay.com](mailto:info@impalapay.com)  
May 2014



INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH IMPALAPAY'S PRODUCTS. NO INTELLECTUAL PROPERTY RIGHTS, LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, IS GRANTED BY THIS DOCUMENT, EXCEPT AS PROVIDED IN IMPALAPAY'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS. IMPALAPAY ASSUMES NO LIABILITY WHATSOEVER, AND IMPALAPAY DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF IMPALAPAY'S PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

The Programs (which include both the software and documentation) contain proprietary information; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent, and other intellectual and industrial property laws. Reverse engineering, disassembly, or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and we disclaim liability for any damages caused by such use of the Programs.

ImpalaPay may make changes to specifications and product descriptions at any time, without notice.

No part of this publication may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language or computer language, in any form or by any means, electronic mechanical, magnetic, optical, chemical, manual, or otherwise, without the prior written permission of ImpalaPay Ltd.

Copyright © 2014 by ImpalaPay Ltd. All rights reserved.

All other trademarks are property of their respective owners.

Impala Pay Ltd.  
2<sup>nd</sup> Floor, Twiga Towers  
Murang'a Road, Nairobi CBD  
P.O. Box 10971 – 00100  
Nairobi, Kenya

# Table of Contents

1. Document Information	3
1.1.DocumentHistory	3
1.2.Document Conventions	3
1.3.Related Documents	3
1.4. Glossary	3
2. Introduction	4
3. Portal and API Design Overview	5
4. Web Services Interaction	7
4.1. Client Initiated	7
4.1.1. Request Session Id	7
4.1.2. Remit Money	9
4.1.3. Query Balance	12
4.1.4. Query Client Reference Number	13
4.1.5. Query Foreign Exchange Rate	15
5. Message Status Codes	17
6. Appendices	19
A. Airtel Africa Coverage	19
B. Sample Code	21
B1. Java	21
B2. PHP	25
C. References	26

## 1. Document Information

### 1.1.DocumentHistory

Release	Date	Author	Comments
Revision 1	May 2014	Michael Wakahe	1 <sup>st</sup> working version.
Revision 2	Feb 2015	Michael Wakahe	2 <sup>nd</sup> working version.

### 1.2.Document Conventions

N/A

### 1.3.Related Documents

None.

### 1.4. Glossary

API .....	Application Programmer Interface
CAK .....	Communications Authority of Kenya
HTTP .....	Hyper Text Transfer Protocol
MNO .....	Mobile Network Operator e.g. Safaricom
MO .....	Mobile Originated SMS
MSISDN .....	A number uniquely identifying a subscription in a GSM mobile network
MT .....	Mobile Terminated SMS
PRSP .....	Premium Rate Services Provider
SLA .....	Service Level Agreement

---

## 2. Introduction

---

ImpalaPay Ltd is an independent Unified Payment and Messaging platform Provider offering mobile money transfer, mobile banking and related electronic commerce services. Impala Pay has been offering electronic money transfer services since 2004, well before the current market acceptability. Impala Pay is a limited liability companies incorporated and based in Kenya with subsidiaries in Dubai and representations in 6 other countries.

ImpalaPay is a mobile remittance and airtime top up hub and an aggregate partner to the Airtel Africa network. Airtel is present in 18 countries in Africa. The *Airtel Remittance Platform* (recommend suitable product brand) is software that allows for remittance to Airtel Africa subscribers through a web based interface (API). This document serves to describe to software integrators the various commands used to interact with the platform.

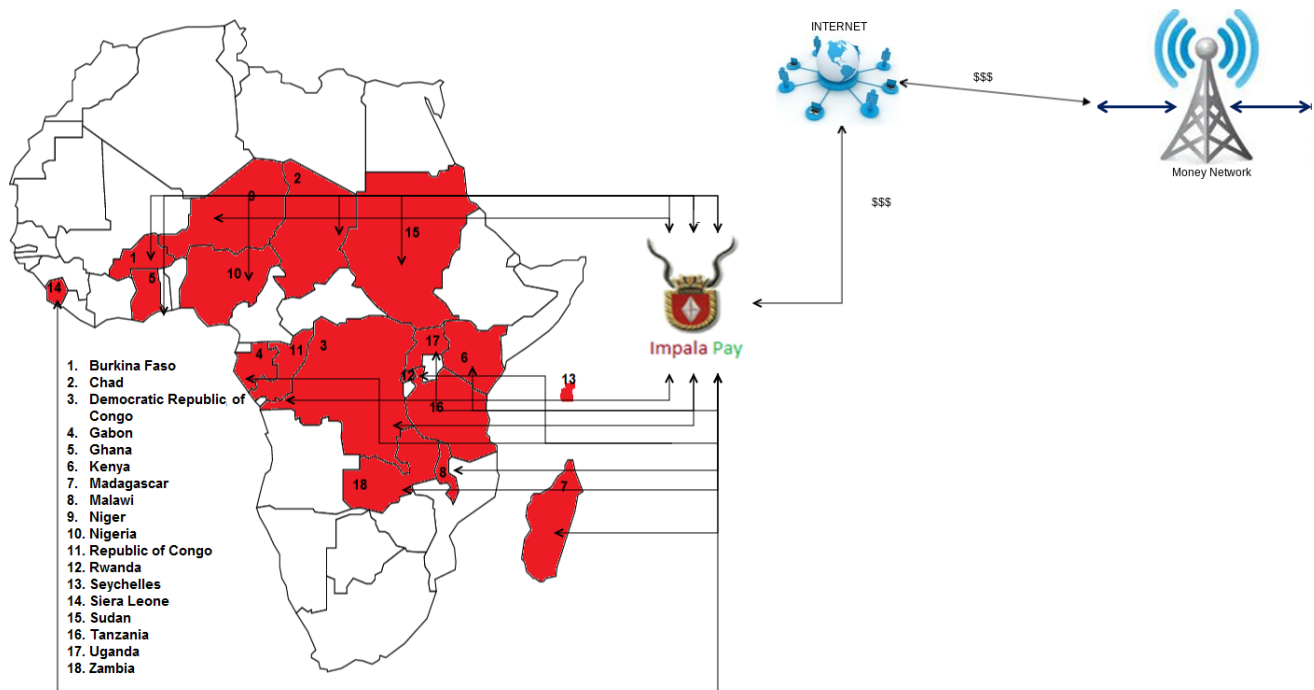


Figure 2.1: Airtel Remittance Platform Overview

---

### 3. Portal and API Design Overview

---

This section serves to give an overview of the design considerations in the platform. The platform portal is accessible through the URL <https://.....>

The communication between client and server is synchronous. There are a variety of conditions to be determined before a remittance transaction is completed. For example, the recipient must be a valid mobile number and the total amount held by the recipient after remittance must not exceed the network allowed amount. The server attempts to finish all of these within a socket connection.

Communication between client and server always utilizes HTTPs. This is regardless of whether the client or the server initiated the communication. HTTPs is widely adopted for providing a secure tunnel for client-server communication over the Internet.

Internally, the platform has many features to guarantee that it works correctly, including a continuous best effort approach to remit into Airtel accounts and security to prevent suspicious activity. It allows businesses and other institutions to remotely monitor their individual accounts.

The web service API utilizes a RESTful approach. Specifically JSON is the means for data interchange between client and server.

Session Ids are generated by the server in order for the client to transact. The server authenticates the client using a challenge mechanism that utilizes the client public certificate. A Session Id is a randomly generated text that is presented to the server in order for a client to execute or query. It expires after roughly double the time it takes to perform the longest type of instruction on the server.

The Session Ids kept securely on the Impala Pay end and are never repeated. We advise the client to keep them securely on his or her end to avoid a replay attack while the session is still valid. One means can be to cache them in memory instead of writing them to RDBMS or file where they are human readable.

A unique Transaction Ids is generated for each request. Transaction Ids graduate through various states until success or failure occurs. Failure can be due to various reasons including a non-existent phone number, the recipient already having funds and unable to receive further funds or due to suspicious remittance activity.

There is a best effort approach to the delivery for remittance requests. The platform will continue trying within an expiry window. After the duration of an expiry window, the server will inform the client that remittance could not occur during that time.

The web portal allows for integrators to view their account activity. Details include account balance and activity. For security reasons, transaction records can neither be created nor altered from the web portal.

---

## 4. Web Services Interaction

---

The following section details the commands used to interact with the system.

### 4.1. Client Initiated

The following are commands that are initiated from the client side. There is always a response for every command sent to the system. This is regardless of whether or not the command was successful. In this manner, you can determine whether or not the command has been successfully processed by the platform. The *Status Code* parameter is used to indicate this.

#### 4.1.1. Request Session Id

A Session Id is a text that is used for identification. All other client requests must be accompanied by a Session Id in order for the server to successfully process them. It is temporal in nature, that is it expires after a predetermined amount of time. For subsequent client requests, a new one must be requested for.

The Session Id is always unique. For security reasons it is recommended that the Client application safeguards the Session Ids it receives as they can be used to carry out legitimate transactions.

In order to receive a Session Id, the client application requests the server through the instruction described in this section. Upon successful authentication, the Session Id is sent to a particular URL that is preconfigured. This URL can only be altered with the assistance of the server system administrator. The following sequence diagrams illustrate the process of acquiring a Session Id.



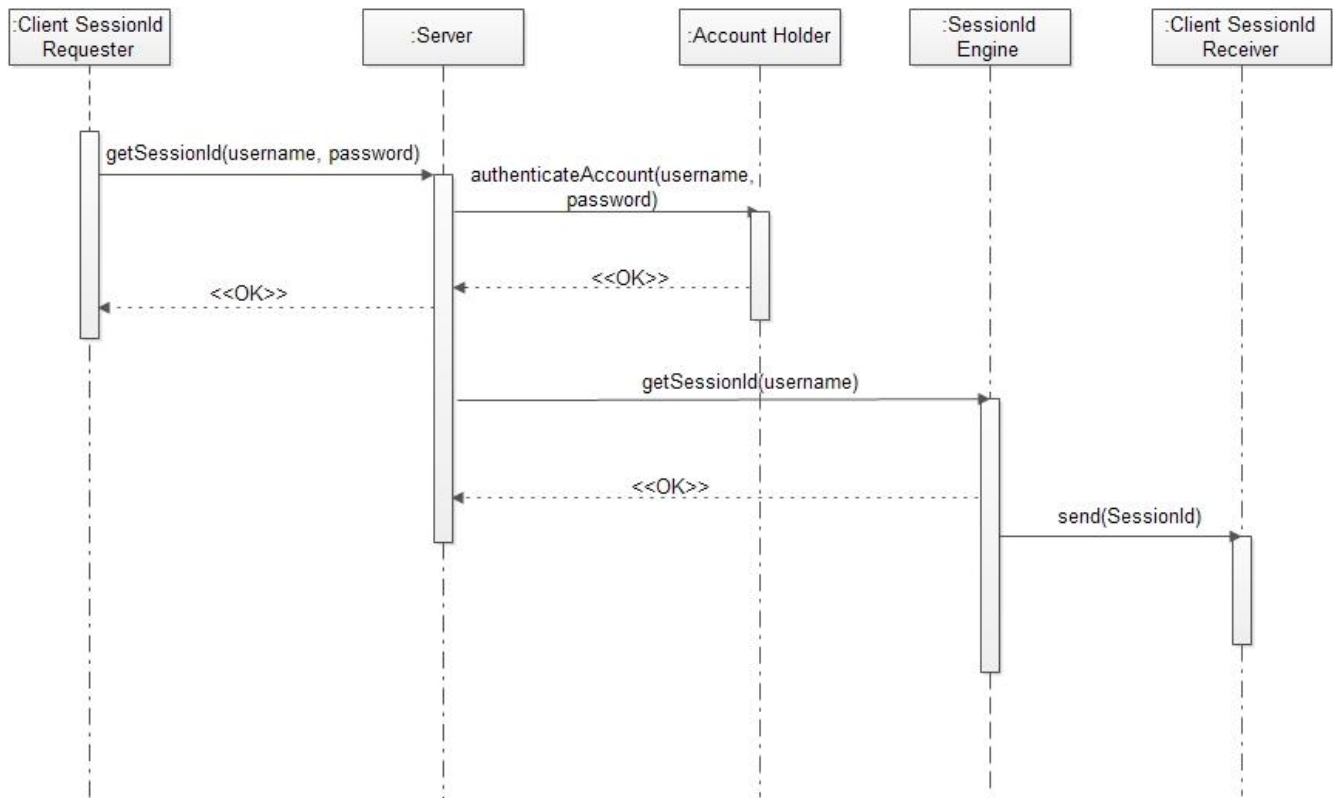


Figure 4.1: Session Id Sequence Flow Diagram

Web Service Endpoint: <https://...../sessionid>

#### Client Request Parameters

Parameter	Value	Compulsory	Description
api_username	a username	Yes	A valid username that is permitted to remit money. Should be URL encoded using the UTF-8 scheme.
api_password	a password	Yes	A valid password that is permitted to remit money. Should be URL encoded using the UTF-8 scheme.

ExampleJSON Request Body: {"api\_username":"demo", "api\_password":"password" }

### Server Response Parameters

Parameter	Value	Compulsory	Description
api_username	a username	Yes	A valid username that is permitted to remit money. Will be URL encoded using the UTF-8 scheme.
session_id	session id text	Yes	Randomly generated text that will be used by the Client for identification
status	a status	Yes	A status indicating whether the server accepted the request or not

ExampleJSON Response Body: {"api\_username":"demo", "session\_id":"1234567", "status":"Ok"}

The following are extra notes concerning a Session Id:

- A Client application can only have one valid Session Id at a time. If a Client initiates a request for a new Session Id before a previously held one expires, then the old one will be expired.

#### **4.1.2.Remit Money**

The following sequence diagram illustrates sending money to an Airtel wallet.

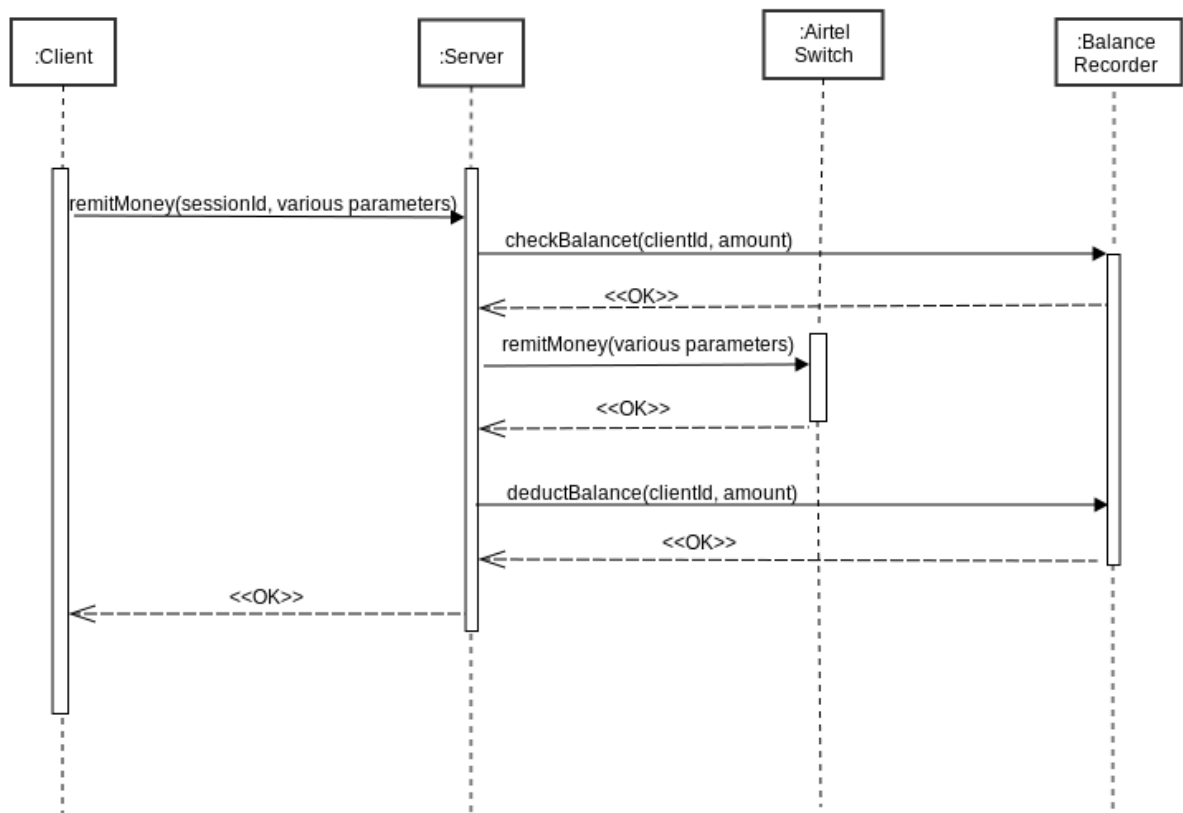


Fig 4.2.2 Money Remittance

Web Service Endpoint: <https://...../remit>

#### Client Request Parameters

Parameter	Value	Compulsory	Description
api_username	a username	Yes	A valid username that is permitted to remit money. Will be URL encoded using the UTF-8 scheme.
session_id	session id text	Yes	Randomly generated text that is used by the Client for identification
source_country_code	country code	Yes	A2 letter country code following the <a href="http://en.wikipedia.org/wiki/ISO_3166-2">ISO 3166-2</a> <sup>1</sup> standards. Examples can be seen in Appendix A: <i>Airtel Africa Coverage</i> .
sendername	name	Yes	The full name of the sender,

<sup>1</sup>[http://en.wikipedia.org/wiki/ISO\\_3166-2](http://en.wikipedia.org/wiki/ISO_3166-2)

			for example “Joe Smith”.
recipient_mobile	mobile phone number	Yes	The mobile phone number of the recipient, including the country dialing code. For example, a Kenyan phone number would be 254733123456.
amount	a number	Yes	A floating point number, with a maximum of 2 decimal places. For example 60.35
recipient_currency_code	currency code	Yes	A three letter currency code, using the <a href="http://en.wikipedia.org/wiki/ISO_4217">ISO 4217</a> <sup>2</sup> standard. Currency codes for the Airtel Africa countries can be found in Appendix A: <i>Airtel Africa Coverage</i> .
recipient_country_code	country code	Yes	A2 letter country code following the <a href="http://en.wikipedia.org/wiki/ISO_3166-2">ISO 3166-2</a> standards.
reference_number	reference number	Yes	A reference number generated by the client application. This can be used to identify a financial transaction from the client side. Note that the client application should strive to generate a unique reference number for each remittance transaction.
sendertoken	sender identification token	Yes	This is a unique identifier of the sender. Examples are a mobile phone number, Permanent Account Number (PAN) or a bank card number.
client_datetime	a date and time	Yes	A date-time of when the client sent the request. It should be in the form: YYYY-MM-DDThh:mm:ss±hh:mm For example: 2014-06-11T13:54:27+03:00 This follows the <a href="http://en.wikipedia.org/wiki/ISO_8601">ISO 8601</a> <sup>3</sup> recommendations.

<sup>2</sup>[http://en.wikipedia.org/wiki/ISO\\_4217](http://en.wikipedia.org/wiki/ISO_4217)

<sup>3</sup>[http://en.wikipedia.org/wiki/ISO\\_8601](http://en.wikipedia.org/wiki/ISO_8601)

Example JSON Request Body: {"api\_username": "demo", "session\_id": "54321", "source\_country\_code": "AU", "sendername": "Joe Smith", "recipient\_mobile": "254733123456", "amount": "120.25", "recipient\_currency\_code": "KES", "recipient\_country\_code": "KE", "reference\_number": "AU321", "sendertoken": "1514324567", "client\_datetime": "2014-06-11T13:54:27+03:00"}

#### Server Response Parameters

Parameter	Value	Compulsory	Description
api_username	a username	Yes	A valid username that is permitted to remit money. Will be URL encoded using the UTF-8 scheme.
transaction_id	text	Yes	A Transaction Id is a unique identifier for a transaction. It is generated by the server and can be referenced in future.
status	a status	Yes	A status indicating whether the server accepted the request or not.

#### Example JSON Response Body:

Valid Request: {"api\_username": "demo", "transaction\_id": "47123e90-f1a6-11e3-ac10-0800200c9a66", "command\_status": "Ok", "remit\_status": "SUCCESS"}

#### **4.1.3. Query Balance**

This API functionality allows for one to query the balance of money. The float value is in US Dollars and the country value is in the local currency.

Web Service Endpoint: <https://...../balance>

#### Client Request Parameters

Name	Value	Compulsory	Description
api_username	a username	Yes	A valid username that is

			permitted to remit money. Will be URL encoded using the UTF-8 scheme.
session_id	session id text	Yes	Randomly generated text that is used by the Client for identification

Example JSON Request Body: {"api\_username": "demo" "session\_id": "54321"}

#### Server Response Parameters

Parameter	Value	Compulsory	Description
api_username	a username	Yes	A valid username that is permitted to remit money. Will be URL encoded using the UTF-8 scheme.
float	a number	Yes	The client float held in an escrow account in US Dollars.
balance	String	Yes	A list of all Airtel Africa country codes and corresponding balance. The balance cannot go below zero value in any country.
status	a status	Yes	A status indicating whether the server accepted the request or not.

#### Example JSON Response Body:

Valid Request: {"api\_username": "demo", "float": "10500.50",  
"balance": "BF-234,TD-6780,TD-0,CD-900,GA-890.23,GH-12.5,KE-7002,MG-23,MW-452.32,NE-345,NG-456,CG-980,RW-390,SC-932,SL-452.92,TZ-953.29,UG-0,ZM-8", "status": "Ok"}

#### **4.1.4. Query Client Reference Number**

This functionality is particularly useful in case the Client application does not receive a response when attempting to remit money (Section 4.1.2) owing to an HTTP timeout or other Internet connectivity disruption.

Web Service Endpoint: <https://...../refNumStatus>

### Client Request Parameters

Name	Value	Compulsory	Description
api_username	a username	Yes	A valid username that is permitted to remit money. Will be URL encoded using the UTF-8 scheme.
session_id	session id text	Yes	Randomly generated text that is used by the Client for identification
reference_number	reference number	Yes	A reference number generated by the client application. This can be used to identify a financial transaction from the client side.

Example JSON Request Body: {"api\_username": "demo", "session\_id": "54321", "reference\_number": "AU321"}

### Server Response Parameters

Parameter	Value	Compulsory	Description
api_username	a username	Yes	A valid username that is permitted to remit money. Will be URL encoded using the UTF-8 scheme.
transaction_id	text	Yes	A Transaction Id is a unique identifier for a transaction. It is generated by the server and can be referenced in future.
transaction_status	text	Yes	The status of this particular transaction.
status	a status	Yes	A status indicating whether the server accepted the request or not.

### Example JSON Response Body:

Valid Request, with transaction successfully completed: {"api\_username": "demo", "transaction\_id": "47123e90-f1a6-11e3-ac10-0800200c9a66", "transaction\_status": "SUCCESS", "status": "Ok"}

Valid Request, with transaction not having been registered by server:

{"api\_username": "demo", "transaction\_id": "47123e90-f1a6-11e3-ac10-080020c9a66", "transaction\_status": "UNKNOWN\_REF\_NUM", "status": "Ok"}

#### 4.1.5. Query Foreign Exchange Rate

The exchange rate of a country is measured against the US Dollar. It is updated approximately once a day. This query allows the Client application to determine what will be used at that instant.

The amount given in the response has the equivalent value of USD \$1 (One Dollar).

Web Service Endpoint: <https://...../xchangeRate>

##### Client Request Parameters

Name	Value	Compulsory	Description
api_username	a username	Yes	A valid username that is permitted to remit money. Will be URL encoded using the UTF-8 scheme.
session_id	session id text	Yes	Randomly generated text that is used by the Client for identification
currency_code	currency code	Yes	A three letter currency code, using the <a href="http://en.wikipedia.org/wiki/ISO_4217">ISO 4217</a> <sup>4</sup> standard. Currency codes for the Airtel Africa countries can be found in Appendix A: <i>Airtel Africa Coverage</i> .

Example JSON Request Body: {"api\_username": "demo", "session\_id": "54321", "currency\_code": "KES"}

##### Server Response Parameters

Parameter	Value	Compulsory	Description
api_username	a username	Yes	A valid username that is permitted to remit money. Will be URL encoded using the UTF-8 scheme.
amount	a number	Yes	The amount that is equivalent to one US Dollar. The accuracy is 6 decimal points.
client_datetime	a date and time	Yes	The current date-time of the server. It will be in the form: YYYY-MM-DDThh:mm:ss±hh:mm For example:

---

<sup>4</sup>[http://en.wikipedia.org/wiki/ISO\\_4217](http://en.wikipedia.org/wiki/ISO_4217)



			2014-06-11T13:54:27+03:00 This follows the <a href="#">ISO 8601</a> <sup>5</sup> recommendations.
status	a status	Yes	A status indicating whether the server accepted the request or not.

#### ExampleJSON Response Body:

Valid Request: {"api\_username":"demo",, "amout":"87.541238","client\_datetime":"2014-06-11T13:54:27+03:00", "status": "Ok"}

---

<sup>5</sup>[http://en.wikipedia.org/wiki/ISO\\_8601](http://en.wikipedia.org/wiki/ISO_8601)

---

## 5. Status Codes

---

Status codes are divided into two categories:

- **Command Status Codes:** these are to give feedback on the Client conformity to the API command structure;
- **Money Transfer Status Codes:** these are to give information on the state of remittance to an Airtel mobile subscriber.

### i. Command Status Codes

These have the keyword “*command\_status*”. Below is a listing with explanations:

Value	Interpretation
OK	
INVALID_SESSIONID	Incorrect session_id
INVALID_REFERENCE_NUMBER	There is no transaction with the provided reference number.
INACTIVE_ACCOUNT	The account has been deactivated.
INVALID_PASSWORD	Provided password is incorrect.
FAILED_TRANSACTION	The transaction failed
PROVIDED_RECIPIENT_CURRENCY_AND_RECIPIENTCOUNTRY_MISMATCH	Generated when the provided currency code and the country of the recipient do not match.
COUNTRYCODE_UNAVAILABLE	Generated when the provided country code is incorrect.
CURRENCY_UNAVAILABLE	Generated when the currency code entered does not match any currency codes in the system.
UNKNOWN_USERNAME	Generated when the user name entered does not match the one in the database.
INVALID_PARAMETERS	This error is generated when the user name or password fields are blank

ii. Money Transfer Status Codes

A status code is generated in response to an attempt to send airtime, query balance or check the status of a transaction. Below is a list of them with their description. These have the keyword “remit\_status”. Below is a listing with explanations:

Status_code	Status_Description
0100019	PAYEE_DAILY_CUMULATIVE_AMT_REACHED
0100007	PAYEE_DAILY_CUMULATIVE_COUNT_REACHED
0100023	PAYEE_MONTHLY_CUMULATIVE_AMT_REACHED
0100011	PAYEE_MONTHLY_CUMULATIVE_COUNT_REACHED
60030	PAYEE_MAXIMUM_BALANCE_EXCEEDED
0100018	PAYER_DAILY_CUMULATIVE_AMT_REACHED
0100006	PAYER_DAILY_CUMULATIVE_COUNT_REACHED
0100022	PAYER_MONTHLY_CUMULATIVE_AMT_REACHED
0100010	PAYER_MONTHLY_CUMULATIVE_COUNT_REACHED
0100004	PAYER_REQUESTED_AMT_GREATER_PER_TXN_AMT
0100002	USER_NOT_ALLOWED_AS_PAYER
00051	INVALID_MSISDN_LENGTH
00126	RECEIVER_PARTY_NOT_ACTIVE
00067	SENDER_PARTY_NOT_ACTIVE
00666	PAYEE_NOT_FOUND
00667	PAYER_NOT_FOUND
01029	AUTHENTICATION_FAILED
00032	UNKNOWN_ERROR
00028	AMOUNT_NOT_NUMERIC
00027	AMOUNT_FIELD_EMPTY
00317	SENDER_PARTY_BARRED
S000	SUCCESS
60019	INSUFFICIENT_FUNDS

---

## 6. Appendices

---

### A. Airtel Africa Coverage

Country	Country Code	Currency	Currency Code	Airtel Network	Calling Code
Burkina Faso	BF	West African CFA franc	XOF	Airtel Burkina Faso	+226
Chad	TD	Central African CFA franc	XAF	Airtel Chad	+235
Democratic Republic of the Congo (DRC)	CD	Central African CFA franc	XAF	Airtel DRC	+243
Gabon	GA	Central African CFA franc	XAF	Airtel Gabon	+241
Ghana	GH	Ghanaian cedi	GHS	Airtel Ghana	+233
Kenya	KE	Kenya shilling	KES	Airtel Kenya	+254
Madagascar	MG	Malagasy ariary	MGA	Airtel Madagascar	+261
Malawi	MW	Malawian kwacha	MWK	Airtel Malawi	+265
Niger	NE	West African CFA franc	XOF	Airtel Niger	+227
Nigeria	NG	Nigerian naira	NGN	Airtel Nigeria	+234
Republic of the Congo	CG	Central African CFA franc	XAF	Airtel Congo B	+242
Rwanda	RW	Rwandan franc	RWF	Airtel Rwanda	+250
Seychelles	SC	Seychelles rupee	SCR	Airtel Seychelles	+248
Sierra Leone	SL	Sierra Leonean leone	SLL	Airtel Sierra Leone	+232
Tanzania	TZ	Tanzanian shilling	TZS	Airtel Tanzania	+255
Uganda	UG	Ugandan shilling	UGX	Airtel Uganda	+256
Zambia	ZM	Zambian kwacha	ZMW	Airtel Zambia	+260

## B. Remittance Limits

The following table lists the limits of transactions per country. This is both in amount and frequency.

Country	Limit
Kenya	A subscriber can receive a maximum of KES xxx in one transaction. (S)he can receive a total number of x remittances within 24 hours. (S)he can send a maximum amount of KES xxx in one transaction.

## C. Sample Code

Below are examples of sending and receiving messages to the gateway.

### C1. Java

The below java class can be used to make requests to the Api and print out the response

```
import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.io.UnsupportedEncodingException;


import org.apache.http.HttpResponse;

import org.apache.http.client.ClientProtocolException;

import org.apache.http.client.methods.HttpPost;

import org.apache.http.entity.StringEntity;

import org.apache.http.impl.client.CloseableHttpClient;

import org.apache.http.impl.client.HttpClients;

import org.apache.http.util.EntityUtils;

import org.apache.commons.lang3.exception.ExceptionUtils;

import org.apache.log4j.Logger;


import com.google.gson.JsonElement;

import com.google.gson.JsonParser;


/**
 * can send POST text data, to HTTP
 * <p>
 * Copyright (c) ImpalapayLtd.,Sep 13, 2014
 *
 * @author <a href="mailto:eugene@impalapay.com">Eugene Chimita</a>
 *
 */
```

```

public class ApiRequest {

    private String url;//the url to the Api endpoint e.g.https://airtel.impalapay.com/refNumStatus
    private String params;// a valid json object containing the requiredparameters

    boolean retry;

    private Logger logger;

    private String dbName = "";

    public ApiRequest() {

        logger = Logger.getLogger(this.getClass());
    }

    public ApiRequest(String url, String params) {

        super();

        this.url = url;

        this.params = params;

    }

    public String doPost() {

        CloseableHttpClient httpclient = HttpClients.createDefault();

        StringEntity entity;

        String out = "";

        try {

            entity = new StringEntity(params);

            HttpPost httppost = new HttpPost(url);

            httppost.setEntity(entity);

            HttpResponse response = httpclient.execute(httppost);

```

```

// for debugging
System.out.println(entity.getContentType());
System.out.println(entity.getContentLength());
System.out.println(EntityUtils.toString(entity));
System.out.println(EntityUtils.toByteArray(entity).length);

// System.out.println("-----");
System.out.println(response.getStatusLine());
System.out.println(url);

BufferedReader rd = new BufferedReader(new InputStreamReader(
    response.getEntity().getContent()));

out = rd.readLine();

JsonElement root = new JsonParser().parse(out);

String specificvalue = root.getAsJsonObject().get("replace with key of the value to retrieve")
    .getString();

System.out.println(specificvalue);

/**
 * String line = ""; while ((line = rd.readLine()) != null) {
 * //System.out.println(line); }
 */

} catch (UnsupportedEncodingException e) {
    logger.error("UnsupportedEncodingException for URL: " + url + "");
    logger.error(ExceptionUtils.getStackTrace(e));

} catch (ClientProtocolException e) {

```



```
        logger.error("ClientProtocolException for URL: " + url + "");  
        logger.error(ExceptionUtils.getStackTrace(e));  
  
    } catch (IOException e) {  
        logger.error("IOException for URL: " + url + "");  
        logger.error(ExceptionUtils.getStackTrace(e));  
    }  
    return out;  
}  
  
}
```

## C2. PHP

Receive session\_id request response.

```
<?php
/*
    Listens to HTTP POST that contains Json Object and prints to file

    @author Eugene Chimita
    20th September 2014
*/

// This is the file that the HTTP POST information will be written to.
// Adjust accordingly.
$logFile = "/tmp/sessionLog.txt";

$jsonData= file_get_contents('php://input');

$phpArray = json_decode($jsonData);

//*****
// No need to change what is below this line
//*****

//fetch each key value pair posted per request
foreach ($phpArray as $key => $value) {

    $final=$key.'='.$value."\r\n".$final;
    $ip=$_SERVER['REMOTE_ADDR'];
    $time= gmdate("l jS \of F Y h:i:s A",$_SERVER['REQUEST_TIME']);
    $url=$_SERVER['REQUEST_URI'];
    $method=$_SERVER['REQUEST_METHOD'];
    $remoteport=$_SERVER['REMOTE_PORT'];
    $protocol=$_SERVER['SERVER_PROTOCOL'];
    $software=$_SERVER['SERVER_SOFTWARE'];
    $port=$_SERVER['SERVER_PORT'];
}

$fileHandle = fopen($logFile, 'a') or die("Unable to open the listenerLog.txt.");
fwrite($fileHandle, "Remote Ip Address:". "\t". $ip. "\r\n");
fwrite($fileHandle, "Time:". "\t". "\t". $time. "\r\n");
fwrite($fileHandle, "Method:". "\t". "\t". $method. "\r\n");
fwrite($fileHandle, "Remote Port:". "\t". $remoteport. "\r\n");
fwrite($fileHandle, "Protocol:". "\t". $protocol. "\r\n");
fwrite($fileHandle, "Local Port:". "\t". $port. "\r\n");
fwrite($fileHandle, "*****end*****". "\r\n". "\r\n". "\r\n");
fwrite($fileHandle, $final);
fclose($fileHandle);

$output = "Thank you";
echo $output;
?>
```

## **D. References**

1. [Airtel Kenya](#)
2. [Communications Commission of Kenya](#)

For more information on the (IMTS) – Airtel Corridor, please visit:

<http://airtel.impalapay.com>.

The IMTS—Airtel Corridor, the Impala Pay logo, and Impala Pay are either registered trademarks or trademarks of Impala Pay Ltd in Kenya and/or other countries.

All other trademarks are the properties of their respective owners.

©2014 Impala Pay Ltd. All rights reserved.

2nd Floor,  
Twiga Towers  
Murang'a Road, CBD  
Nairobi, Kenya

Email:

[info@impalapay.com](mailto:info@impalapay.com)

Tel: +254 (0)20-222-  
0635

Web:

[www.impalapay.com](http://www.impalapay.com)