ngun



![](data:None;base64,)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.

2nd Floor, Twiga Towers

Murang’a Road, Nairobi CBD

P.O. Box 10971 – 00100

Nairobi, Kenya

Table of Contents

[1. Document Information 2](#__RefHeading__8416_726830005)

[1.1. Document History 2](#__RefHeading__8418_726830005)

[1.2. Document Conventions 2](#__RefHeading__8420_726830005)

[1.3. Related Documents 3](#__RefHeading__8422_726830005)

[1.4. Glossary 3](#__RefHeading__8424_726830005)

[2. Introduction 3](#__RefHeading__8426_726830005)

[3. Portal and API Design Overview 4](#__RefHeading__8428_726830005)

[4. Web Services Interaction 5](#__RefHeading__8430_726830005)

[4.1. Client Initiated 5](#__RefHeading__8432_726830005)

[4.1.1. Request Session Id 5](#__RefHeading__8434_726830005)

[4.1.2. Remit Money 7](#__RefHeading__8436_726830005)

[4.1.3. Query Balance 8](#__RefHeading__8438_726830005)

[4.1.4. Query Client Reference Number 9](#__RefHeading__8440_726830005)

[4.1.5. Query Foreign Exchange Rate 10](#__RefHeading__8442_726830005)

[5. Status Codes 12](#__RefHeading__8444_726830005)

[6. Appendices 14](#__RefHeading__8446_726830005)

[A. Airtel Africa Coverage 14](#__RefHeading__8448_726830005)

[B. Remittance Limits 15](#__RefHeading__8450_726830005)

[C. Sample Code 16](#__RefHeading__8452_726830005)

[C1. JAVA 16](#__RefHeading__8454_726830005)

[C2. PHP 27](#__RefHeading__8456_726830005)

[D. References 30](#__RefHeading__8458_726830005)

# 1. Document Information

## 1.1. Document History

|  |  |  |  |
| --- | --- | --- | --- |
| **Release** | **Date** | **Author** | **Comments** |
| Revision 1 | May 2014 | Michael Wakahe | 1st working version. |
| Revision 2 | Nov 2014 | Michael Wakahe | Provide balance for each country in local  currency |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## 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.

![](data:None;base64,)

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://airtel.impalapay.com](https://airtel.impalapay.com/).

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.

![](data:None;base64,)

Figure 4.1: Session Id Sequence Flow Diagram

Web Service Endpoint: <http://airtel.impalapay.com/>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. |
| session\_challenge | plain text | Yes | This is the decrypted version of the text sent by the server. The client has decrypted using her/his private key. |

Example JSON Request Body: {“api\_username”:“demo”, “api\_password”:“password”, “session\_challenge”:“plain text” }

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 |

Example JSON 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.

![](data:None;base64,)

Fig 4.2.2 Money Remittance

Web Service Endpoint: <http://airtel.impalapay.com/>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 | A 2 letter country code following the [ISO 3166-2[[1]](#footnote-2)](http://en.wikipedia.org/wiki/ISO_3166-2) standards. Examples can be seen in Appendix A: *Airtel Africa Coverage*. |
| sender\_name | name | Yes | The full name of the sender, 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 [ISO 4217[[2]](#footnote-3)](http://en.wikipedia.org/wiki/ISO_4217) standard. Currency codes for the Airtel Africa countries can be found in Appendix A: *Airtel Africa Coverage*. |
| recipient\_country\_code | country code | Yes | A 2 letter country code following the [ISO 3166-2](http://en.wikipedia.org/wiki/ISO_3166-2) 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. |
| sender\_token | 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 [ISO 8601[[3]](#footnote-4)](http://en.wikipedia.org/wiki/ISO_8601) recommendations. |

Example JSON Request Body: {“session\_id”:“54321”, “source\_country\_code”:“AU”, “sender\_name”:“Joe Smith”, “recipient\_mobile”:“254733123456”, “amount”:“120.25”, “recipient\_currency\_code”:“KES”, “recipient\_country\_code”:“KE”, “reference\_number”:“AU321”, “sender\_token”:“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: <http://airtel.impalapay.com/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: {“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: <http://airtel.impalapay.com/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: {“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: <http://airtel.impalapay.com/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 [ISO 4217[[4]](#footnote-5)](http://en.wikipedia.org/wiki/ISO_4217) standard. Currency codes for the Airtel Africa countries can be found in Appendix A: *Airtel Africa Coverage*. |

Example JSON Request Body: {“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:  2014-06-11T13:54:27+03:00  This follows the [ISO 8601[[5]](#footnote-6)](http://en.wikipedia.org/wiki/ISO_8601) recommendations. |
| status | a status | Yes | A status indicating whether the server accepted the request or not. |

Example JSON Response Body:

Valid Request: “api\_username”:“demo”,, “amout”:“87.541238”, “client\_datetime”:“2014-06-1T13:54:27+03:00”, “status”: “Ok”}

# 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.

1. Command Status Codes

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

|  |  |
| --- | --- |
| **Value** | **Interpretation** |
| OK |  |
| INVALID\_SESSIONID |  |
| INVALID\_PARAMETES |  |
| UNKNOWN\_USERNAME | Generated when the session id generated by the system has expired. The user has to login again. |
| CURRENCY\_UNAVAILABLE | Generated when the currency code entered does not match any currency codes in the system. |
| COUNTRYCODE\_UNAVAILABLE | Generated when the provided country code is incorrect. |
| PROVIDED\_RECIPIENT\_CURRENCY\_AND\_RECIPIENTCOUNTRY\_MISMATCH | Generated when the provided currency code and the country of the recipient do not match. |
| FAILED\_TRANSACTION | The transaction failed. |
| INVALID\_PASSWORD | Provided password is incorrect. |
| INACTIVE\_ACCOUNT | The account has been deactivated |
| INVALID\_REFERENCE\_NUMBER | There is no transaction with the provided reference number |

1. Money Transfer Status Codes

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

|  |  |
| --- | --- |
| **Value** | **Interpretation** |
| SUCCESS |  |
| INSUFFICIENT\_FUNDS |  |
| UNKNOWN\_DESTINATION |  |
| CURRENCY\_UNAVAILABLE |  |
| COUNTRY\_UNAVAILABLE |  |
| WALLET\_AMOUNT\_EXCEEDED |  |
| RECEIVE\_FREQUENCY\_EXCEEDED |  |
| LOW\_AMOUNT |  |
| HIGH\_AMOUNT |  |

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.

|  |  |
| --- | --- |
| **Value** | **Description** |
| ACCEPTED\_FOR\_DELIVERY | The top-up request has been accepted by the gateway for delivery. |
| SUBMITTED\_TO\_OPERATOR | The message has been submitted to the operator airtime server. |
| TOPUP\_SUCESS | The top-up attempt is successful. |
| TOPUP\_FAILURE | The top-up attempt is not successful. |
| UNKNOWN\_USERNAME | The username or email used to authenticate is not registered with the gateway. |
| WRONG\_PASSWORD | The password supplied does not match the username. |
| ACCOUNT\_SUSPENDED | The account from which the top-up attempt has been made is not active, it may be in a suspended or deleted state. |
| INVALID\_PARAMETERS | Not enough or invalid parameters have been provided when submitting a request to send a message. This also includes invalid amounts for top-up. |
| INSUFFICIENT\_FUNDS | The user has exhausted the airtime allowed for top-up. |
| SERVER\_ERROR | The gateway has experienced an internal error. |

# 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 messages to the API and receiving response.

### C1. JAVA

1.Q**uery Forex**

**import** com.google.gson.Gson;

**import** com.impalapay.airtel.servlet.api.APIConstants;

**import** java.io.BufferedReader;

**import** java.io.IOException;

**import** java.io.InputStreamReader;

**import** java.io.OutputStreamWriter;

**import** java.net.MalformedURLException;

**import** java.net.URL;

**import** java.net.URLEncoder;

**import** java.security.KeyManagementException;

**import** java.security.NoSuchAlgorithmException;

**import** java.security.SecureRandom;

**import** java.security.cert.Certificate;

**import** java.security.cert.X509Certificate;

**import** java.util.HashMap;

**import** java.util.LinkedHashMap;

**import** java.util.Map;

**import** javax.net.ssl.HostnameVerifier;

**import** javax.net.ssl.HttpsURLConnection;

**import** javax.net.ssl.SSLContext;

**import** javax.net.ssl.SSLPeerUnverifiedException;

**import** javax.net.ssl.SSLSession;

**import** javax.net.ssl.TrustManager;

**import** javax.net.ssl.X509TrustManager;

**import** **static** org.junit.Assert.\*;

**import** org.joda.time.LocalDateTime;

**import** org.junit.Test;

/\*\*

\* Tests the {@link QueryForex}

\* <p>

\* Copyright (c) ImpalaPay Ltd., Sep 31, 2014

\*

\* **@author** <a href="mailto:eugene@impalapay.com">Eugene Chimita</a>

\*

\*/

**public** **class** TestQueryForex {

final String CGI\_URL = "https://airtel.tawi.mobi/xchangeRate";

/\*\*

\* Test method for

\* .

\*/

@Test

**public** **void** testDoPostHttpServletRequestHttpServletResponse() {

// Test by calling the URL without all required parameters

Map<String, String> expected = **new** HashMap<>();

expected.put("command\_status",APIConstants.***COMMANDSTATUS\_INVALID\_PARAMETERS***);

Map<String, String> user = **new** HashMap<>();

user.put("api\_username", " ");

user.put("session\_id", " ");

user.put("currency\_code", "");

Gson g = **new** Gson();

String jsonData = g.toJson(user);

String jsonResult = g.toJson(expected);

System.***out***.println(getResponse(CGI\_URL, jsonData));

}

/\*\*

\* **@param** httpsUrl

\* **@param** args

\*/

**private** String getResponse(String httpsUrl, String args) {

URL url;

String response = "";

**try** {

// Create a context that doesn't check certificates.

SSLContext sslContext = SSLContext.*getInstance*("TLS");

TrustManager[] trustMgr = getTrustManager();

sslContext.init(**null**, // key manager

trustMgr, // trust manager

**new** SecureRandom()); // random number generator

HttpsURLConnection.*setDefaultSSLSocketFactory*(sslContext

.getSocketFactory());

url = **new** URL(httpsUrl);

HttpsURLConnection con = (HttpsURLConnection) url.openConnection();

con.setRequestMethod("POST");

con.setDoOutput(**true**);

// Guard against "bad hostname" errors during handshake.

con.setHostnameVerifier(**new** HostnameVerifier() {

**public** **boolean** verify(String host, SSLSession sess) {

**if** (host.equals("airtel.tawi.mobi")) {

**return** **true**;

} **else** {

**return** **false**;

}

}

});

// Send data to the output

sendData(con, args);

// Dump all cert info

// printHttpsCert(con);

// Dump all the content

response = getContent(con);

} **catch** (MalformedURLException e) {

System.***err***.println("MalformedURLException");

e.printStackTrace();

} **catch** (IOException e) {

System.***err***.println("IOException");

e.printStackTrace();

} **catch** (NoSuchAlgorithmException e) {

System.***err***.println("NoSuchAlgorithmException");

e.printStackTrace();

} **catch** (KeyManagementException e) {

System.***err***.println("KeyManagementException");

e.printStackTrace();

}

**return** response;

}

/\*\*

\* Send data to the url

\*

\* **@param** con

\*/

**private** **void** sendData(HttpsURLConnection con, String args) {

**if** (con != **null**) {

**try** {

// send data to output

OutputStreamWriter writer = **new** OutputStreamWriter(

con.getOutputStream());

writer.write(args);

writer.flush();

writer.close();

} **catch** (IOException e) {

System.***err***.println("IOException");

e.printStackTrace();

}

}

}

**private** String getContent(HttpsURLConnection con) {

StringBuffer buff = **new** StringBuffer("");

**if** (con != **null**) {

**try** {

BufferedReader br = **new** BufferedReader(**new** InputStreamReader(

con.getInputStream()));

String input;

**while** ((input = br.readLine()) != **null**) {

buff.append(input + "\n");

}

br.close();

} **catch** (IOException e) {

e.printStackTrace();

}

}// end 'if(con != null)'

**return** buff.toString().trim();

}

/\*\*

\* **@return** {@link TrustManager}

\*/

**private** TrustManager[] getTrustManager() {

TrustManager[] certs = **new** TrustManager[] { **new** X509TrustManager() {

**public** X509Certificate[] getAcceptedIssuers() {

**return** **null**;

}

**public** **void** checkClientTrusted(X509Certificate[] certs, String t) {

}

**public** **void** checkServerTrusted(X509Certificate[] certs, String t) {

}

} };

**return** certs;

}

/\*\*

\* **@param** con

\*/

**private** **void** printHttpsCert(HttpsURLConnection con) {

**if** (con != **null**) {

**try** {

System.***out***.println("Response Code : " + con.getResponseCode());

System.***out***.println("Cipher Suite : " + con.getCipherSuite());

System.***out***.println("\n");

Certificate[] certs = con.getServerCertificates();

**for** (Certificate cert : certs) {

System.***out***.println("Cert Type : " + cert.getType());

System.***out***.println("Cert Hash Code : " + cert.hashCode());

System.***out***.println("Cert Public Key Algorithm : "

+ cert.getPublicKey().getAlgorithm());

System.***out***.println("Cert Public Key Format : "

+ cert.getPublicKey().getFormat());

System.***out***.println("\n");

}

} **catch** (SSLPeerUnverifiedException e) {

System.***err***.println("SSLPeerUnverifiedException");

e.printStackTrace();

} **catch** (IOException e) {

System.***err***.println("IOException");

e.printStackTrace();

}

}// end 'if(con != null)'

}

}

### C2. PHP

1. **Php Session Listener Script**
2. Capture server response on request for session\_id

<?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](http://africa.airtel.com/kenya/)

2. [Communications Commission of Kenya](http://www.cck.go.ke/)

![](data:None;base64,)

1. <http://en.wikipedia.org/wiki/ISO_3166-2> [↑](#footnote-ref-2)
2. <http://en.wikipedia.org/wiki/ISO_4217> [↑](#footnote-ref-3)
3. <http://en.wikipedia.org/wiki/ISO_8601> [↑](#footnote-ref-4)
4. <http://en.wikipedia.org/wiki/ISO_4217> [↑](#footnote-ref-5)
5. <http://en.wikipedia.org/wiki/ISO_8601> [↑](#footnote-ref-6)