Interface Specification

Interswitch© Services

Created By:	Taiwo, Olabamiji
Approved By:	Omotayo, Olalekan
Document Ref:	ISA10001I
Version:	1.0
Filename:	ISA10001I Authentication

Authorized distribution: G (specific customer use only)

No part of this document may be reproduced or transmitted
in any form or by any means, electronic or mechanical,
for any purpose, without express written permission of:

Interswitch, Enterprise Architecture & System Integration
InterSwitch Limited
Plot 1648C Oko Awo Street, Victoria Island
Lagos, Nigeria

© 2015 InterSwitch Ltd. All rights reserved.
Products and company names mentioned herein may
be the trademarks of their respective owners.

The information in this publication is subject to change without
prior notice and is provided "as is" without warranty of any kind. While every precaution
has been taken in the preparation of this document, InterSwitch Ltd assumes no
responsibility for errors or omissions, or for damages resulting from the use of
the information contained herein. The entire risk arising out of the use of
this information remains with the recipient.

Scope

This document describes how a third party service provider will be authenticated by Interswitch© Services.

Modification Log

V	Author	Date	Summary of changes
1.0	Taiwo, Olabamiji	2nd Nov, 2015	Initial Draft

References

S/N	Document Ref	V	Title

Definitions and Acronyms

S/N	Terms	Description

Contents

SCOPE **3**

1 INTRODUCTION **6**

1.1 OVERVIEW 6

2 DESIGN **7**

2.1 OVERVIEW 7

3 AUTHENTICATION **8**

3.1 HTTP HEADERS FOR AUTHENTICATION 8

3.2 REALM 9

4 HEADER COMPUTATION **10**

4.1 AUTHORIZATION COMPUTATION 10

4.2 SIGNATURE COMPUTATION 10

5 COMMUNICATION **11**

6 HTTP RESPONSE CODE **11**

1 Introduction

1.1 Overview

This document describes how a third-party will request for an Interswitch Service. All our APIs require users to be authenticated before they can access any of our resources.

2 Design

2.1 Overview

We authenticate every call to our resources using a combination of **HTTP headers**. These HTTP Headers must be sent for every call in other to access InterSwitch resources. Note that all data formats and response definitions are in conformance with the **REST** standard.

3 Authentication

In order to do authentication, you will be required to send the following HTTP headers parameters: Timestamp, Nonce, Signature, and the Signature Method as part of your requests. Also, you will be required to send an Authorization HTTP header parameter.

All fields are required except otherwise stated.

3.1 HTTP Headers for Authentication

The following describes the important headers required to be sent.

S/N	Headers		Description
1.	Timestamp	Mandatory	Request timestamp in epoch (unix). It must be in seconds and NOT milliseconds. e.g. 1440071245
2.	Nonce	Mandatory	Request nonce. A unique generated value for each request. It should not be repeated. Length should NOT be more than 64 characters. e.g. f5ab08912e11d147b65d0789e1ad58472dca64e9
3.	Authorization	Mandatory	Describes the Realm and Identity of the user requesting access to resource. See pseudo-code below
4	Signature	Mandatory	Request signature. Must be represented in base 64. The signature is calculated from a combination defined data elements separated by the '&' character. See pseudo-code below
5	SignatureMethod	Mandatory	The cryptographic hash function used to calculate the Signature e.g. “SHA512”
6	Content-Type	Mandatory	The MIME type of the body of the request e.g. application/json

Sample Authentication Headers


```
Authorization: InterswitchAuth SUtJQUVFMzhDMjRBMzYzRTRGQzAxREVCRkJGRTlGOERDMUY0QkNCMkJDNDg=
Content-Type: application/json
Nonce: 091f90dbbf9748f0b2854a038dbdac44
SignatureMethod: SHA512
Signature: mf73jzzhaVN8U0oZ7iiKcVgEzBY=
Timestamp: 1440071245
```


3.2 Realm

Authorization on Interswitch API can either be done through the OAuth Bearer realm or custom "InterswitchAuth" realm. The section below defines how to use each. Except otherwise stated, the InterswichAuth will be used. In cases where the Bearer realm is required, it will be stated in the API documentation page.

3.2.1.1 InterswitchAuth Realm

The InterswitchAuth authorization realm is a custom realm that allows you to be authenticated with just your ClientID. In this realm, developer is expected to encode client ID in base 64 in the Authorization header. See sample below.


```
InterswitchAuth SUtJQUYzRDNGMkQyMEY1MTM2REVGNUExMjUwMzZDN0U0QjgzMzgwQjBGMjA
```


3.2.1.2 Bearer Realm

The bearer realm is a reuse of OAuth 2.0 authorization framework where developers are expected to send an access token for authorization. Access token are given/issued to developers either by logging in to the developer console and generating one for themselves or by requesting for access token through an API call. See here for how to generate AccessToken.


```
Bearer
eyJhbGciOiJSUzI1NiJ9.eyJhdWQiOlsicGFzc3BvcnQiXSwic2NvcGUiOlsicHJvZmlsZSJdLCJqdGkiOiI3ZGViNjllNS02ZjQzLTQ5OTUtOTdhNC02NTVjO
GFmNWNkMjciLCJjbGllbnRfaWQiOiJBM0MyNkE2RTVDQTc0QjNBQTU4MUNERDdEMUU0NzRFRCJ9.cN0nnOiYu5KqCrhvUvRaUjlOcEN4W2O7
sfSJVfOBrm2mMa1a0ulLD14iwdsyZ8vCBDUz8TO8BHRuQwS1cp9WDGz55kmhPsS9tuo35pqSlpj-MBuACnzzzBzl92Q5KDxyXaC5q0f1-
utSJrYJOqHqvMZh0duq-6vnfNkZWjF95t83wJWXayy-mUBp7IH1m5mbQdYYNj_09oVS-cL03Drh7NWhBAV1LTNZnJq6Hp3G83aUzJNI-
fmddmNeuYyaQRitcfMT3CzJ-8AG1XGe8Njne00fBaT01pDPM5gr2ZxANBd35jlqKrhmV2qM8Bo5S09JrTmeYAq_s4ty7DAETRoD4Q
```

4 Header Computation

Below is the pseudo-code for computing authorization and signature headers:

4.1 Authorization computation

All messages must be authorized using following the standard 'Basic' authorization model of the web. The HTTP header Authorization should be set to the base 64 encoded value of the client_id (Register and obtain your client-id from https://developer.interswitchng.com).

The **Authorization** value is as follows:

```
InterswitchAuth Base64(client_id)
```

4.2 Signature computation

The signature method should be SHA-512 calculated against the data elements above in the order in which they are listed.

The pseudo-code below shows how this can be done with any programming language of choice:

```
String baseStringToBeSigned = http_verb + "&" +  percent_encode(url) + "&" + timestamp + "&" + nonce + "&" + client_id +
"&" + shared_secret_key;

String parameterStringToBeSigned = percent_encode(value1) + "&" + percent_encode(value2);

String stringToBeSigned = baseStringToBeSigned + "&" + parameterStringToBeSigned;

String signature = Base64(Hash(stringToBeSigned));
```

See sample code in Appendix B below

5 Communication

Requests will be sent over the HTTPS protocol.

6 HTTP Response Code

Please refer to the standard HTTP response codes. Where anything starting with 2XX signifies approved, 4XX means client error and 5XX indicates server error. When the response codes start with 4XX or 5XX, an error object will be returned to explain further the reason for this failure. See Appendix A for error code descriptions.

APPENDIX A (HTTP Error Codes)

Response Code	Description
200	OK
400	BAD REQUEST
401	UNAUTHORIZED
403	FORBIDDEN
404	NOT FOUND
500	INTERNAL ERROR
501	NOT IMPLEMENTED

APPENDIX B (Response Code)

Sample response Codes obtainable are below. Note that this list is not exhaustive.

S/N	Response Code	Response description
1	E10	The service field tag has not been populated. Please ask Interswitch Support for more help.
2	E11	This requests expects a transactionType field to be set and it is not. Please ask Interswitch Support for more help
3	E12	SecurityToken header is missing.
4	E13	Username or password is wrong.
5	E14	Missing content type header.
6	E15	The transaction you are trying to do is not supported
7	E16	The service you have specified is not available
8	E17	The service provider you have specified has not been set up on the platform.
9	E18	The service is unreachable at the moment
10	E19	A bad response was received from service provider
11	E20	Your request has timed out
12	E21	An unknown error has occurred. Please ask Interswitch Support for more help.
13	E22	Access key has not been supplied. Ensure the Authorization header has been set appropriately.
14	E23	Nonce has not been supplied. Ensure the Nonce header has been set appropriately.

15	E24	Please synchronize your time with the server and ensure your time is not outside of 30mins of the Server local time. The standard time zone used is GMT
16	E25	Nonce has already been used
17	E26	Signature failed for this request. Please ask Interswitch Support for more help.
18	E27	An unknown error occured during Authentication. Please ask Interswitch Support for more help.
19	E28	Your credentials do not have access to this resource
20	E29	Timestamp is missing in your header
21	E30	SecurityToken header is bad.
22	E31	XML data is badly formatted
23	E32	JSON data is badly formatted
24	E33	Keystore is not accessible. Contact Interswitch.
25	E34	The system is busy at the moment, please try again later.
26	E35	Bad request to service provider. Contact Interswitch.
26	E36	Bad request to service processor. Contact Interswitch.
27	E37	Bad request from client. Kindly check your message.
28	E38	Bad response from service provider. Contact Interswitch.
29	E39	Bad response from service processor. Contact Interswitch.
30	E40	Bad response to client. Contact Interswitch.
31	E41	Bad request from client. Kindly check your message.

APPENDIX C (Sample Code)

```
// Timestamp

TimeZone lagosTimeZone = TimeZone.getTimeZone("Africa/Lagos");

Calendar calendar = Calendar.getInstance(lagosTimeZone);

long timestamp = calendar.getTimeInMillis() / 1000;


// Nonce

UUID uuid = UUID.randomUUID();

String nonce = uuid.toString().replaceAll("-", "");


// Signature Method

String signatureMethod = "SHA1";


// Signature

String httpMethod = "POST"; // HTTP Method of the resource that is being called

String encodedResourceUrl = "RESOURCE_URL";  // put the resource URL here

String clientId = "CLIENT_ID";  // put your client Id here

String clientSecretKey = "CLIENT_SECRET_KEY";  // put your client secret here

String signatureCipher = httpMethod + "&" + encodedResourceUrl + "&" + timestamp + "&" + nonce + "&" + clientId + "&" +
clientSecretKey;

MessageDigest messageDigest = MessageDigest.getInstance(signatureMethod);

byte[] signatureBytes = messageDigest.digest(signatureCipher.getBytes());

String signature = new String(Base64.encodeBase64(signatureBytes));


// Setting HTTP Header values

String resourceUrl = "https://sandbox.interswitchng.com/passport/oauth/token";

HttpPost post = new HttpPost(resourceUrl);

post.setHeader("TIMESTAMP", timestamp);

post.setHeader("NONCE", nonce);

post.setHeader("SIGNATURE_METHOD", signatureMethod);

post.setHeader("SIGNATURE", signature)
```