



Open API Project

OneAPI v2.0 Common Information Guide

Document Version 1.0

Document Revision History

| Rev # | Date | Description | Author |
|-------|--------------|---|------------------------|
| 1.0 | June 24 2013 | Initial version, based on Aepona OneAPI 2.0 Common Information Guide doc v1.1a. | Aepona/Akagi Kobayashi |
| | | | |
| | | | |
| | | | |
| | | | |

Copyright

2013

© Aepona Limited,

Beacon House,

Clarendon Dock,

Belfast,

BT1 3BG

All rights reserved. This document or any part thereof may not, without the written consent of Aepona Limited, be copied, reprinted or reproduced in any material form including but not limited to photocopying, transcribing, transmitting or storing it in any medium or translating it into any language, in any form or by any means, be it electronic, mechanical, xerographic, optical, magnetic or otherwise.

The information contained in this document is proprietary and confidential and all copyright, trademarks, trade names, patents and other intellectual property rights in the documentation are the exclusive property of Aepona Limited unless otherwise specified. The information (including but not limited to data, drawings, specification, documentation, software listings, source or object code) shall not at any time be disclosed directly or indirectly to any third party without Aepona Limited's prior written consent.

The information contained herein is believed to be accurate and reliable. Aepona Limited accepts no responsibility for its use by any means or in any way whatsoever. Aepona Limited shall not be liable for any expenses, costs by damage that may result from the use of the information contained within this document. The information contained herein is subject to change without notice.

Table of Contents

| | | |
|----------|--|----------|
| 1 | What You Will Need – A Summary | 5 |
| 2 | Developer Access | 5 |
| 2.1 | Accessing the REST API from a Browser..... | 5 |
| 2.2 | Basic Authentication..... | 5 |
| 2.3 | Using the REST APIs from a Browser..... | 6 |
| 2.4 | Using the REST APIs from an Application..... | 6 |
| 3 | Response Codes & Exceptions | 7 |
| 3.1 | Response Codes..... | 7 |
| 3.2 | REST Exceptions..... | 7 |
| 3.2.1 | Service Exceptions..... | 8 |
| 3.2.2 | Policy Exceptions..... | 9 |

1 What You Will Need – A Summary

| | |
|---|---|
| 1 | The service endpoints made available to you when you registered with your service provider. |
| 2 | The API documentation specific to the APIs you wish to use. |
| 3 | Knowledge of an application programming language . Your service provider may provide you with client libraries and bindings. |
| 4 | To receive SMS sent to your application by end users, you will need to obtain a registrationId (e.g. short code or similar identifier) to identify your application to the network, which ensures correct routing. For more information on this, contact your service provider. |

2 Developer Access

This section provides the detailed information required to successfully access the services available to you.

2.1 Accessing the REST API from a Browser

Simply enter the API URL in a browser. At this point you will be prompted to access the required certificate from the server.

2.2 Basic Authentication

In order to use the various services, basic authentication is required. The information below is applicable to any of the APIs.

In all cases, if you try to use the service APIs without basic authentication credentials, you will receive the following HTTP error:

```
HTTP/1.1 401 Unauthorized
```

2.3 Using the REST APIs from a Browser

Once you have entered the URL into the browser, a pop-up window is displayed which allows you to enter your username and password. Please enter your application username and password in order to authenticate.

2.4 Using the REST APIs from an Application

From within a Java application, code similar to the following is required to set up basic authentication:

```
import java.net.HttpURLConnection;

import org.apache.geronimo.mail.util.Base64;

final static String username = "partner";

final static String password = "partnerpassword";

final static String url = "http://your-endpoint-
here";

final static String credentials = username + " : "
+ password;

final static String authHeaderValue = new
String(Base64.encode(credentials.getBytes()));

HttpURLConnection con =
(HttpURLConnection)new
URL(url).openConnection();

con.setRequestProperty ("Authorization", "Basic "
+ authHeaderValue);
```

Replace "username" and "password" with your application username and password.

3 Response Codes & Exceptions

3.1 Response Codes

HTTP response codes are used to indicate:

- **200** – Success!
- **400** – Bad request; check the error message for details
- **401** – Authentication failure, check your authentication details
- **403** – Forbidden; please provide authentication credentials
- **404** – Not found: mistake in the host or path of the service URI
- **405** – Method not supported: for example you mistakenly used a HTTP GET to create an SMS instead of a POST
- **500** – The server encountered an unexpected condition. This could be incorrect authentication details or limited user permission
- **503** – Server busy and service unavailable. Please retry the request.

For more details on these, refer to <http://www.ietf.org/rfc/rfc2616.txt>.

3.2 REST Exceptions

```
HTTP/1.1 400 Bad Request
Content-Type: application/json
Content-Length: 1234
Date: Thu, 04 Jun 2009 02:51:59 GMT
{"requestError": {
  "serviceError": {
    "messageId": "SVC0002",
    "text": " Invalid input value for message part %1",
    "variables": " tel:+016309700000"
  }
}}
```

HTTP 400 indicates either a service exception or a policy exception.

The **requestError** object contains either a **serviceException** or a **policyException** object.

The **serviceException** describes the reason why the service cannot accept the request; in this example because the phone number was too long.

A **policyException** object means that the request syntax is valid, however an operator policy has been broken, e.g. you are requesting to charge an amount that exceeds a limit that the operator has set.

serviceException and **policyException** share the same body, which includes the following pairs:

- identifier pair – for the exception (**messageId**)
- text pair – to describe it consistently (text)
- a variables pair – to indicate any specific cause of the error (variables). The variables relate to the %1 placeholder(s) in the text.

3.2.1 Service Exceptions

The following are some examples of exceptions which may be thrown when an operation fails:

Table 1: Service Exceptions

| Error | Explanation |
|---|---|
| SVC0001 – Service error occurred | A service-related error has occurred as a result of a client invocation on the service. This category can be used for implementation-specific errors. Contact the support team. |
| SVC0002 – Invalid input value | An input parameter value is not of the expected type. Check the parameter types and re-submit your request. |
| SVC0004 – Invalid address(es) | The request terminal device address does not exist. Update your request and re-submit. |
| SVC0007 – Invalid charging information | The charging information provided is invalid. Update your request and re-submit. |

| | |
|--------------------------------|---|
| SVC0270 – Charge failed | The charge failed. This could be due to, for example, the transaction not being found or payment not allowed. Contact the support team. |
|--------------------------------|---|

3.2.2 Policy Exceptions

A policy exception means that the request syntax is valid, however an operator policy has been broken.

The two types of policy exceptions are as follows:

- **POL0002**: Privacy Error - There was a problem using the Privacy service. Check your method use and re-submit your request
- **POL0001**: Policy error occurred. This exception may be thrown to indicate a fault relating to a policy associated with the service. This category can be used for implementation-specific errors such as:

Table 2: Policy Error Codes

| Error | Explanation |
|---|---|
| POL-003 : TPA profile not found | The third party application cannot be found. |
| POL-006 : TPA exceeded its maximum allowed rate of transactions | The maximum rate of transactions is exceeded. Ensure that the rate of your requests is within the limits set up in your SLA, e.g. 10 TPS (Transactions Per Second). |
| POL-011 : Charging not supported | Inline charging is not supported by this operator. Re-submit your request without charging information. |
| POL-014 : White List is enforced, and address is not in White List | A white list is enforced and the number is not in the white list. Check your SLA details. |
| POL-015 : Black List is enforced, and address is in Black List | A black list is enforced and the number is in the black list. Check you SLA details. |
| POL-016 : Max Requests is enforced, and max requests has been exceeded | The maximum number of requests for this service is exceeded. Contact the support team. |
| POL-017 : Operation is not allowed | The method/operation is not supported in your current SLA. Check your SLA and use a method that is supported. |

| Error | Explanation |
|---|---|
| POL-018: All targets were rejected for MDN access and authorization failure | <p>This indicates that none of the destination numbers can be retrieved by the internal address resolver such as LDAP or Lookup.</p> <p>It includes white/black list rejection when the destination number cannot be found in either list that is enforced. In this case, check your policy contract and request the number to be added to/removed from the appropriate list.</p> |
| POL-020: Max Message Length is enforced, and max message length has been exceeded | A maximum message length policy is in place and you have exceeded this. Check you SLA for the maximum message length, update your message and re-submit your request. |
| POL-021: Min Message Length is enforced, and message length is less than min allowed | A minimum message length policy is in place and your message length is less than this minimum. Check you SLA for the maximum message length, update your message and re-submit your request. |
| POL-022: Receipting is enforced, and receipting has not been enabled | A receipt has been requested but it is not enabled for this service. Remove the receipt request and re-submit you request. |
| POL-038: Max Charge Amount is enforced and maximum charge amount has been exceeded | A maximum charge amount is enforced and has been exceeded. Check your SLA for this limit and re-submit your request with the correct amount. |
| POL-039: Min Charge Amount is enforced and charge amount is less than minimum value | A minimum charge amount is enforced and a value less than this has been used. Check your SLA for this limit and re-submit your request with the correct amount. |
| POL-040: Max Destination Addresses is enforced and maximum destination addresses has been exceeded | A maximum destination address limit is enforced and it has been exceeded. Check your SLA for the limit and re-submit your request. |
| POL-042: The requested accuracy is less than the allowable value | The accuracy requested is too high. Re-submit the request with a lower accuracy, i.e. a value of 5000m or more. |

| Error | Explanation |
|---|---|
| POL-049: SPID Black List is enforced and address SPID is in the SPID Black List. | Applicable in multiple carrier deployments, Black List is enforced and the carrier identified by the Service Provider ID is in the black list. Therefore all the addresses from the carrier are rejected. |