Oracle® Communications Services Gatekeeper

OneAPI Application Developer's Guide

Release 5.1

E37522-01

June 2013

Oracle Communications Services Gatekeeper OneAPI Application Developer's Guide, Release 5.1

E37522-01

Copyright © 2011, 2013, Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Contents

3 OneAPI Short Messaging

4 OneAPI Multimedia Messaging

5 OneAPI Terminal Location

6 OneAPI Payment

7 Wholesale Applications Community Enhancements

Preface

This document describes the OneAPI RESTful Web Services offered by Oracle Communications Services Gatekeeper. It includes a high-level overview of the interfaces and operations.

Audience

This book is intended for software developers who wish to integrate functionality provided by telecom networks into their programs by using the OneAPI RESTful Web Services interface in Services Gatekeeper.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc`.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info` or visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs` if you are hearing impaired.

Related Documents

For more information, see the following documents in the Oracle Communications Services Gatekeeper set:

- *Oracle Communications Services Gatekeeper Accounts and SLAs Guide*
- *Oracle Communications Services Gatekeeper Application Developer's Guide*
- *Oracle Communications Services Gatekeeper Communication Service Guide*
- *Oracle Communications Services Gatekeeper Concepts Guid*e
- *Oracle Communications Services Gatekeeper Installation Guide*
- *Oracle Communications Services Gatekeeper Partner Relationship Management Guide*
- *Oracle Communications Services Gatekeeper Platform Developer's Studio Guide*
- *Oracle Communications Services Gatekeeper Platform Test Environment Guide*
- *Oracle Communications Services Gatekeeper SDK User's Guide*
- *Oracle Communications Services Gatekeeper System Administrator's Guide*

1

Services Gatekeeper OneAPI Overview

This chapter presents an overview of Oracle Communications Services Gatekeeper OneAPI server functionality (OneAPI) and the ways that application developers can use this functionality to interact with Services Gatekeeper.

About the Services Gatekeeper OneAPI Interface

This document provides an overview of the basic concepts you need to know in order to create applications that successfully interact with Services Gatekeeper using the OneAPI specifications published for Global System for Mobile Communications (GSM) operators.

Communication Services

OneAPI application request data flows through Services Gatekeeper through communication services. A communication service consists of a service type (such as Multimedia Messaging and Terminal Location and so on), an application-facing interface (also called a “north” interface), and a network-facing interface (also called a “south” interface).

OneAPI operations are processed by the same south interfaces to network components used by the proprietary Services Gatekeeper RESTful facade for supported services.

For more information on communication services, see *Oracle Communications Services Gatekeeper Communication Service Guide*.

The Services Gatekeeper OneAPI interface complies with Open Mobile Alliance (OMA) specifications. For more information on OMA OneAPI specifications, see:

`http://technical.openmobilealliance.org/Technical/Comments.aspx`

Supported OneAPI Services

Services Gatekeeper supports the following OneAPI services:

- SMS
- MMS
- Terminal Location
- Payment

There are a number of communication services for which OneAPI does not provide a standard interface. Services Gatekeeper supports these services using the proprietary

RESTful interface. A complete list of supported services is available in *Oracle Communications Services Gatekeeper RESTful Application Developer's Guide*.

The GSMA provides application developers OneAPI specifications for using these services when interacting with a OneAPI server. Services Gatekeeper, acting as a OneAPI server, provides response messages to operations as specified by the OneAPI specification.

SMS

Applications use the OneAPI RESTful Short Messaging interface over SMPP (`wlng_nt_sms_px21.ear)` to send and receive SMS messages, to fetch SMSs and delivery status reports, and to start and stop a notification.

When the request body for an SMS operation contains a request for a delivery receipt, the application provides a correlator for the message being sent and includes an endpoint address for returning the delivery notification.

For a complete description of the OneAPI RESTful SMS interface, including example operations, see Chapter 3, "OneAPI Short Messaging."

MMS

Applications use the OneAPI RESTful Multimedia Messaging interface over MM7 (`wlng_nt_multimedia_messaging_px21.ear`) to send a multimedia message (MMS) and to fetch information on MMSs that have been received for the applications and stored by Services Gatekeeper.

Applications use the interface to fetch those messages, get delivery status on sent messages, and start and stop a notification.

For a complete description of the OneAPI RESTful MMS interface, including example operations, see Chapter 4, "OneAPI Multimedia Messaging."

Terminal Location

Applications use the OneAPI RESTful Location interface over MLP (`wlng_nt_terminal_location_px21.ear`) to get a location for an individual terminal or a group of terminals.

For a complete description of the OneAPI RESTful Location interface, including example operations, see Chapter 5, "OneAPI Terminal Location."

Payment

Applications use the OneAPI RESTful Payment interface over Diameter Ro (`wlng_nt_sms_px21.ear`) to charge an amount to an end-user's account, refund amounts to that account, query charge amount status and list charge amount transactions. Applications can also reserve amounts, reserve additional amounts, charge against the reservation and release the reservation.

For a complete description of the OneAPI RESTful Location interface, including example operations, see Chapter 6, "OneAPI Payment."

Wholesale Applications Community Payment Endpoint Support

Services Gatekeeper can function as a Wholesale Applications Community (WAC) Operator Endpoint Server in the Payment API flow using OneAPI 2.1 Payment.

See Chapter 7, "Wholesale Applications Community Enhancements," for more information.

About Application Testing

You can develop applications and test them using the Application Test Environment (ATE).

This tool hosts a set of virtual communication services that correspond to many of the communication services provided by Services Gatekeeper. For example, you can test tasks such as opening sessions, sending and receiving messages, and examining delivery reports through a set of virtual communication services without having to connect to the network operator´s Services Gatekeeper installation.

For more information on using the Application Test Environment, see *Oracle Communications Services Gatekeeper SDK User's Guide*.

2

Interacting with the OneAPI RESTful Facade

This chapter provides an overview of how the Oracle Communications Services Gatekeeper RESTful Web Services work with the Oracle Communications Services Gatekeeper One API (OneAPI) SMS, MMS, Terminal Location, and Payment operations.

It describes the ways in which applications use OneAPI HTTP methods in association with the RESTful Web Services facade to interact with Services Gatekeeper. Applications add specific information to the request headers, and, in some cases, they send their message payload as an attachment.

Services Gatekeeper supports OneAPI SMS, MMS, Payment, and Terminal Location operations as specified by the GSMA.

About RESTful Facade Architecture

Services Gatekeeper supports OneAPI operations by complying with industry specifications published by the GSMA and the Parlay Group. Services Gatekeeper uses an implementation of Oracle's JSR 311 Jersey Java API for RESTful Web Services (JAX-RS) to facilitate communication between applications and network nodes.

For information about Oracle JAX-RS, see the Oracle Fusion Middleware documentation:

http://download.oracle.com/docs/cd/E17904_01/web.1111/e13734/rest.htm

For information about JSR 311, see the GlassFish Web site:

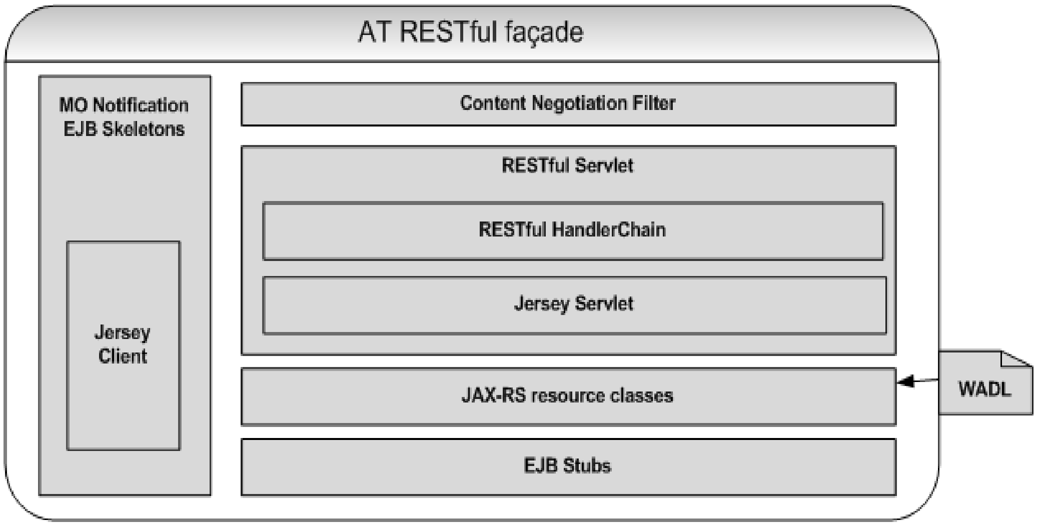
http://jsr311.java.net/

OneAPI SMS, MMS, Terminal Location and Payment services are supported.

RESTful Facade Modules

The Services Gatekeeper Access Tier (AT) RESTful facade consists of multiple modules as shown in the following figure.

Figure 2–1 RESTful Facade



Southbound (application-initiated) requests are received by the RESTful Servlet and are then directed to the RESTful handler chain in preparation for delivery by the JAX-RS. The JAX-RS communicates the RESTful requests to the appropriate Service Enabler for execution at the network tier by the correct Service Enabler Enterprise JavaBean (EJB). Communication between northbound (network-triggered) traffic and applications is handled by the same modules in reverse.

For information on Service Enablers, see *Oracle Communications Services Gatekeeper Communication Service Guide*.

A Web Application Description Language (WADL) file is used to generate the JAX-RS resource classes used by a communication service to translate between the RESTful request for delivery to the appropriate EJB. For information on WADL, see:

http://wadl.java.net/

For information on creating communication services from WADL files, see the discussion on Using Services Gatekeeper with REST Services in the release notes.

About Configuring OneAPI Server Functionality

The Services Gatekeeper OneAPI server functionality is embedded in the Parlay X plug-in, which is installed and deployed by default. No additional OneAPI configuration is necessary.

General Format of an Operation

The following basic elements are present in the requests that an application makes to the OneAPI RESTful facade, to the responses it receives from the facade, or to both:

- Request-URI and HTTP Methods in requests and Status -Line in responses
- Headers
- Message Body
- Attachments

Request-URI and HTTP Methods

Applications use one of four methods, "GET", "POST", "PUT", or "DELETE", to request that a required action be performed on an abstract or physical resource, which is identified by a Uniform Resource Identifier (URI). The Request-URI is therefore the most important part of any request that an application makes to the RESTful facade.

Here is an example of the OneAPI POST method used to send an SMS message. The request URI is contained in the first line of the example.

```
POST http://example.com/oneapi/1/smsmessaging/outbound/tel%3A%2B5550100/requests
HTTP/1.1
Host: example.com:80
Content-Type: application/x-www-form-urlencoded
Accept: application/json

address=tel%3A%2B15415550100&
address=tel%3A %2B15415550101&
senderAddress=tel:%2B5550100&
message=Hello%20World&
clientCorrelator=123456&
notifyURL=http://application.example.com/notifications/DeliveryInfoNotification&
callbackData=some-data-useful-to-the-requester&
senderName=ACME%20Inc.
```

See Chapter 3, "OneAPI Short Messaging," for information on the parameters used in this example request.

General Format of a Request-URI

A fully qualified OneAPI Request-URI is made up of a sequence of sections, concatenated as:

```
http://host:port/oneapi/api_version/service/service_specific_sections
```

where,

- *host:port*: The hostname and port of your Services Gatekeeper OneAPI installation; for example, **127.0.0.1** and **8001**.
- *api_version*: The version of the OneAPI interface deployed.

  In Services Gatekeeper, the current api version is always **1**.
- service: The communication service required by the method. For example, `smsmessaging`.
- *service_specific_sections*: One or more sections required by the OneAPI specification for the method called. See the examples provided in the following service-specific chapters for more information on required sections.

POST

The POST method accesses a resource factory to create a new resource that does not yet have a URI. Multiple requests to a resource factory can create multiple new resources.

The following statement sets up a subscription to SMS delivery notifications:

```
POST
http://example.com/oneapi/1/smsmessaging/outbound/tel%3A%2B5550100/subscriptions
HTTP/1.1
```

For the POST method:

- The URI in the request represents the factory resource that is accessed to create a new resource. In the above example, **/smsmessaging/outbound/tel%3A%2B5550100/subscriptions** is the factory resource accessed to create a resource.
- The request body contains the information required to create the resource.
- If the resource is created, the response body will contain the identifier for the new resource. If the operation fails, the response body will contain the error response.

PUT

The PUT method creates a resource that has a predetermined URI. This method can be used to update a resource or to start a stateful process. For example, an application uses the following statement to release a payment reservation:

```
PUT
http://example.com/oneapi/1/payment/tel%3A%2B5550100/transactions/amountReservatio
n/abc123
```

For the PUT method:

- The URI in the request represents the resource to update or to start a stateful process on. In the example, **/tel%3A%2B5550100/transactions/amountReservation/abc123** represents the resource accessed to release the payment reservation.
- The request body for this operation will contain the required information. The JSON object will contain, for example, information on the description and amount of the reservation to release.
- If the operation fails, the response body will contain the error response.

GET

The GET method retrieves the state of a specific resource that has been previously set up. The specific resource is identified in the query string. For example, an application attempts to retrieve the location of a terminal whose address is "tel&3A%2B15415550100" by using the following statement:

```
GET
http://example.com/oneapi/1/location/queries/location?&address=tel&3A%2B1541555010
0&requestedAccuracy=1000
```

For the GET method:

- The URI in the request represents the query string that uniquely identifies the resource whose status the application wishes to retrieve. In the example, the value for (**location?&address**) is the unique address of the terminal. (It is the address of the terminal, `{"address":"tel:15415550100",}` in JSON representation).
- The request body for this operation will be empty.
- The response will provide information on the location of the resource with an accuracy of 1000 meters. If the operation fails, it will contain the error response.

  In order to complete the operation, the application must access the specified location and use the correlator to retrieve the notification.

DELETE

The DELETE method removes a specified resource. The application provides the correlator or the identifier for the resource that must be removed in the Request-URI. An application stops SMS delivery notifications with the following request:

```
DELETE http://example.com/oneapi/1/smsmessaging/outbound/subscriptions/abc123
```

For the DELETE method:

- The URI in the request contains the correlator, which is a value that uniquely identifies the resource the application wishes to remove. In the example, **abc123** is the value which the application provided as the correlator when it requested notifications on a terminal's status.
- The request body for this operation will be empty.
- The response body will be empty, or will contain the error response if the operation fails.

Headers

The requests and responses for RESTful operations include the following header fields:

- Authorization: The Authorization header field is a required field and is found in all requests. It indicates the type of authentication and security. For example:

  ```
  Authorization: Basic QWxhZGRpbjpvcGVuIHNlc2FtZQ==
  ```

  For more information, see "Authentication and Security".
- Accept: OneAPI only supports JSON as the response format. For example:

  ```
  Accept: application/json
  ```
- Location: Location headers are found in responses to certain requests. They are used for the identification of a new resource or to redirect the recipient to a location other than the Request-URI for completion of the request.
- Content-Length: This is the length of the request (or response) body.
- Content-Type: The MIME-type value for the Content-Type header field may be **multipart/form-data**, **application/json**, **application/x-www-form-urlencoded** or **application/xml**.

  The **multipart/form-data** value for the Content-Type header field is described in the next section.
- X-Session-ID: A session ID is required if transaction sessions are used.

Headers for Multipart Messages with Attachments

The OneAPI RESTful-based communication services for Multimedia Messaging use HTTP attachments to transport their content. The OneAPI MMS interface supports multipart/form-data POST requests. When you use RESTful interfaces with the base Services Gatekeeper product, multiple attachments are supported in both application-initiated and network-triggered messages.

When a request message contains one or more messages embedded within it, a specified boundary is placed between the parts of the message and at the beginning and end of the message. For multipart message requests:

- The MIME-type value for the Content-Type header field may be **multipart/form-data** or **application/json**. If the MIME-type value for the Content-Type header field is **multipart/form-data**, the **boundary=** entry is used to provide a value for the boundary between the message parts.
- Each message part contains the following:
  - Content-Disposition header field whose value is **form-data** and contains a **name** attribute with the appropriate value. For example, the Message Part name is **messagePart**. For Multimedia Messages the attribute is **attachments**. Multimedia messages contain an additional **filename** attribute and value.
  - Content-Type header field whose value describes the data format included in the message part.
  - Content-Transfer-Encoding field with the appropriate value, if needed.
- If the content of the message is pure ASCII, the response body contains the message. Otherwise, the response body contains an identifier that is used to fetch the message.

Example 2–1 Example of a Multipart Message Request

```
POST http://example.com/oneapi/1/messaging/outbound/ tel%3A%2B5550100/requests
HTTP/1.1
Content-Length: 12345
Content-Type: multipart/form-data;
boundary="===============123456==";

MIME-Version: 1.0
Host: www.example.com
Date: Thu, 04 Jun 2009 02:51:59 GMT

--===============123456==
Content-Disposition: form-data; name=”root-fields”
Content-Type: application/x-www-form-urlencoded;
address=tel%3A%2B15415550100&
address=tel%3A%2B15415550101&
senderAddress=tel:%2B5550100&
&senderName=ExampleSender
Subject=My%20message&
notifyURL=http://example-application.com/notifications/DeliveryInfoNotification/54
311
&callbackData=
&clientCorrelator=123456
--===============123456==
Content-Disposition: form-data; name=”attachments”; filename=”picture.jpg”
Content-Type: image/gif

GIF89a...binary image data...
--===============123456==-
```

Status-Line

The Status-Line is the first line in any response that an application receives when it interacts with the OneAPI RESTful services interface in Services Gatekeeper. It takes the form:

```
HTTP/1.1 status_code reason_phrase
```

where:

- *status_code* is a three-digit number that indicates the success or failure to fulfill the request.
- *reason_phrase* is a brief description of the successful action performed; or the reason for the failure.

For example:

```
HTTP/1.1 201 Created
```

Table 2–1 lists some of the status codes and reason-phrases commonly encountered when interacting with the Services Gatekeeper OneAPI RESTful interfaces:

Table 2–1 A Sampling of Status Codes and Reason Phrases

Status Code	Reason Phrase	Description
200	OK	Success
201	Created	Success: The requested resource was created.
204	No Content	Success
400	Bad Request	The request cannot be processed. Check the error message for details.
401	Authentication failure	The authentication to the protected resource failed. Check your Services Gatekeeper OneAPI authentication requirements.
403	Forbidden	The authentication credentials provided in the request are not valid.
404	Not found	The resource specified in the URI cannot be found.
405	Method not supported	The method requested is not supported by the service.
501	Internal Server Error	Failure: An unexpected condition prevented the server from fulfilling the request.
503	Server busy and service unavailable. Please retry the request.	The server hosting Services Gatekeeper OneAPI RESTful interface is busy.

The Status-Codes used by OneAPI and supported by Services Gatekeeper conform to Internet Engineering Task Force (IETF) standards. For a complete listing of the HTTP Status Codes and their definitions, see RFC 2616 on the IETF Web site:

`http://www.ietf.org/rfc/rfc2616.txt`

Message Body

The message body for a request or response is present only when required. The message body is a JSON object.

Request Body

When present, the request body provides additional data required to complete the specific request. The following request body for an example Send SMS operation provides the addresses of the recipients and sending party, and the message text:

Example 2–2 Send SMS Request

```
address=tel%3A%2B15415550100&
address=tel%3A %2B15415550101&
senderAddress=tel:%2B5550100&
message=Hello%20World&
```

Response Body

When present, the response body provides data that the application will need for later action. The following response body for the Send SMS operation provides the application with the URI for the sent message.

Example 2–3 Response Body for Send SMS Request

```
{"resourceReference": {"resourceURL": "
http://example.com/1/smsmessaging/outbound/ tel%3A%2B5415550100/requests/abc123"}}
```

Example of a Request and Response

Example 2–4 shows an application's request to query the location of a mobile terminal in the Service Gatekeeper OneAPI RESTful interface.

Example 2–4 Request Associated with Get Terminal Location

```
GET
http://example.com/oneapi/1/location/queries/location?&address=tel&3A%2B1541555010
0&requestedAccuracy=1000 HTTP/1.1
Host: example.com:80
Accept: application/json
```

Example 2–5 shows the response which the application receives containing the terminal location information for the requested subscriber.

Example 2–5 Response Associated with a Get Terminal Location Operation

```
HTTP/1.1 200 OKContent-Type: application/json
Content-Type: application/json
Content-Length: 1234
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"terminalLocationList": {"terminalLocation": {
    "address": "tel:15415550100",
    "currentLocation": {
        "accuracy": "100",
        "altitude": "1001.0",
        "latitude": "-80.86302",
        "longitude": "41.277306",
        "timestamp": "2009-06-03T00:27:23.000Z"
    },
    "locationRetrievalStatus": "Retrieved"
}}}
```

Authentication and Security

The OneAPI RESTful interfaces in Services Gatekeeper use HTTP basic authentication, using username and password. SSL is required. For example:

```
Authorization: Basic QWxhZGRpbjpvcGVuIHNlc2FtZQ==
```

For more information on HTTP basic authentication, see RFC 2617 on the IETF Web site:

http://www.ietf.org/rfc/rfc2617.txt

Services Gatekeeper supports sessionID security with use of the **X-SessionID** header.

Services Gatekeeper also supports subscriber authentication and authorization for access to protected resources by supporting OAuth 2.0. For information on using OAuth 2.0, see the chapter on Services Gatekeeper OAuth 2.0 Authorization Server in the release notes.

Notifications

When an application needs to receive a notification, about a message delivery receipt for example, the application posts a subscription request containing a **notificationURL** location where Services Gatekeeper delivers the notification.

Example 2–6 shows a POST operation subscribing to SMS delivery notifications for a **senderAddress**. The notificationURL is included in the message body.

Example 2–6 SMS Delivery Notification Subscription POST Request

```
POST http://example.com/oneapi/1/smsmessaging/outbound/
tel%3A%2B5550100/subscriptions HTTP/1.1
Host: example.com:80
Content-Type: application/x-www-form-urlencoded
Accept: application/json

notifyURL=http://www.yourURL.here&
criteria="SampleCriteria"&
callbackData=doSomething()
```

An application can subscribe only to its own notifications (that is, to the notifications associated with its start notification requests). Any attempt to subscribe to notifications for other applications will be rejected.

Errors and Exceptions

The Status-Line in the response message indicates the protocol version, the three-digit Status Code, and the reason for the failure of the request.

In addition, Service Exception and Policy Exception objects are represented in the response body as JSON objects of the following form:

```
{"requestError":
           {
           "policyException":"class name of the error object"
              {
              "messageID": "System Generated ID:
              "text":"error message"
              "variables"""relevant string"
              }
           }
 }
```

For Service Exceptions, the value for **type** is:

```
"type":"org.csapi.schema.parlayx.common.v2_1.ServiceException"
```

For Policy Exceptions, the value for **type** is:

```
"type":"org.csapi.schema.parlayx.common.v2_1.PolicyException"
```

3

OneAPI Short Messaging

This chapter describes Oracle Communications Services Gatekeeper OneAPI (OneAPI) Short Messaging (SMS) operations supported by the Oracle Communications Services Gatekeeper RESTful Facade.

About the OneAPI Short Messaging Interface

Applications use the RESTful OneAPI SMS interface to send SMSs, to fetch SMSs, and delivery status reports, and to start and stop notifications.

When the request body for an SMS operation contains a request for a delivery receipt, the application provides a **notifyURL** correlator for the message being sent and includes an endpoint address for returning the delivery notification.

The Services Gatekeeper OneAPI SMS interface complies with Open Mobile Alliance (OMA) specifications. For more information on OMA, see:

`http://technical.openmobilealliance.org/Technical/Comments.aspx`

The information provided in this chapter is based on the OneAPI specification and is provided here for convenience.

REST Service Descriptions Available at Run-time

When the Administration Server for your Services Gatekeeper domain is in the running state, the OneAPI REST service descriptions of these operations can be found at:

```
http://host:port/oneapi/1/smsmessaging/application.wadl
```

where *host* and *port* are the host name and port of the machine on which Services Gatekeeper is installed.

Sending SMS Messages

To send an SMS message, provide the OneAPI-formatted URI of the addresses that must receive the message in the request body. If the sender requires a delivery receipt, specify the required parameters for the receipt.

If the Send SMS operation is successful, the **Location** header field in the response will contain the request identifier (which is also provided in the response body for this operation).

If the application requires a receipt for delivery of the message, the application must provide the **notifyURL** to which notifications are to be sent in the message body.

Authorization

Basic or OAuth 2.0

HTTP Method

```
POST
```

URI

```
http://host:port/oneapi/1/smsmessaging/outbound/senderAddress/requests
```

where *host* and *port* are the host name and port of the machine on which Services Gatekeeper is installed. The *senderAddress* is the subscriber for which the message is being sent.

Request Header

The MIME-type for the Content-Type header field can be **application/x-www-form-urlencoded**, **application/json** or **application/xml**.

Request Body

The request body for the OneAPI send SMS operation accepts the following parameters:

- **address**: String. At least one address is the URL-escaped end user ID; in this case the MSISDN including the **'tel:'** protocol identifier and the country code preceded by '**+**'. i.e., **tel:+15415550100**.
- **message:** String. Must be URL-escaped as per RFC 1738. Messages over 160 characters may be sent as two or more messages by the operator.
- **senderAddress**: String. The address to whom a responding SMS may be sent.
- **clientCorrelator**: String. Optional. Uniquely identifies this create SMS request. If there is a communication failure during the request, using the same client correlator when retrying the request allows the operator to avoid sending the same SMS twice.
- **senderName:** String. Optional. The URL-escaped name of the sender to appear on the terminal. This is the address to whom a responding SMS may be sent.
- **notifyURL**: anyURL. The URL-escaped URL to which a notification of delivery sent. The notifyURL will be ignored if a notification subscription already exists for the senderAddress. The format of this notification is shown in Example 3–1.

callbackData: String. Will be passed back in this notification, so you can use it to identify the message the receipt relates to or any other useful data, such as a function name.

Response Header

The Location header field contains the URI:

```
http://host:port/oneapi/1/smsmessaging/outbound/address/requests/requestID
```

where, *requestID* is the string identifier returned in the response body.

If the request fails, the Status-Line header field will contain the status code and the reason for the failure. See "Errors and Exceptions" for more information.

Response Body

The body of the response contains the request identifier as the string value for the **resourceReference** attribute. It is the request identifier returned in the **Location** header field of the response message and is also included in the resourceReference body. The application uses this request identifier to retrieve the delivery status for the sent message.

The response body for this operation is represented by the following JSON structure, where the value part of the name/value pair indicates its data type:

```
{"resourceReference": {"resourceURL": "String"}}
```

Examples

Example 3–1 shows a sample OneAPI Send SMS request.

Example 3–1 OneAPI Send SMS Request

```
POST http://example.com/oneapi/1/smsmessaging/outbound/
tel%3A%2B5550100/requests HTTP/1.1
Host: example.com:80
Content-Type: application/x-www-form-urlencoded
Accept: application/json

address=tel%3A%2B15415550100&
address=tel%3A %2B15415550101&
senderAddress=tel:%2B5550100&
message=Hello%20World&
clientCorrelator=123456&
notifyURL=http://application.example.com/notifications/DeliveryInfoNotification&
callbackData=some-data-useful-to-the-requester&
senderName=ACME%20Inc.
```

Example 3–2 shows a sample Send SMS response.

Example 3–2 OneAPI Send SMS Response

```
HTTP/1.1 201 Created
Content-Type: application/json
Location: http://example.com/1/smsmessaging/outbound/
tel%3A%2B5550100/requests/abc123
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"resourceReference": {"resourceURL": "
http://example.com/oneapi/1/smsmessaging/outbound/
tel%3A%2B5550100/requests/abc123"}}
```

Query Delivery Status of SMS Message

The Query Delivery Status operation retrieves the delivery status of a message (that was previously sent) by using the system-generated **requestID** returned when the message was created.

If the Query Delivery Status is successful, the response body will contain the delivery status for each of the addresses contained in the original send SMS request.

Authorization

Basic or OAuth 2.0

HTTP Method

```
GET
```

URI

```
http://host:port/oneapi/1/smsmessaging/outbound/senderAddress/requests/requestID/deliveryInfos
```

where:

- *host* and *port* are the host name and port of the machine on which Services Gatekeeper is installed.
- *senderAddress* is the address to which a responding SMS may be sent.
- *requestID* is the identifier returned in the **result** object of the corresponding Send operation.

Request Body

There is no request body.

Response Header

Standard header fields. If the request fails, the Status-Line header field will contain the status code and the reason for the failure. See "Errors and Exceptions" for more information.

Response Body

The response body contains an array of structures as the value for **deliveryInfo**. Each element in the array contains values for the following parameters.

- **address:** String. The telephone number to which the initial message was sent.
- **deliveryStatus:** Enumeration value. Table 3–1 lists the possible statuses:

Table 3–1 Enumeration Values for Delivery Status

Value	Description
DeliveredToNetwork	Successful delivery to the network. For concatenated messages, returned only when all the SMS-parts have been successfully delivered to the network.

Table 3–1 (Cont.) Enumeration Values for Delivery Status

Value	Description
DeliveryUncertain	Delivery status unknown, for example, if it was handed off to another network.
DeliveryImpossible	Unsuccessful delivery; the message could not be delivered before it expired.
DeliveredToTerminal	Successful delivery to the terminal. For concatenated messages, returned only when all the SMS-parts have been successfully delivered to the terminal.
MessageWaiting	The message is still queued for delivery. This is a temporary state, pending transition to one of the preceding states.

A **resourceURL** is included as a reference to the response.

The response body for this operation is represented by the following JSON data structure, where the value part of each name/value pair indicates its data type:

```
{"deliveryInfoList": {
    "deliveryInfo": [
        {   "address": "String",
            "deliveryStatus": "String"},
        {   "address": "String",
            "deliveryStatus": "String"}],
    "resourceURL": "
http://example.com/oneapi/1/smsmessaging/outbound/senderAddress/requests/requestID
/deliveryInfos"
}}
```

Examples

Example 3–3 shows a sample OneAPI Query Delivery Status request.

Example 3–3 OneAPI Query Delivery Status Request

```
GET
http://example.com/oneapi/1/smsmessaging/outbound/tel%3A%2B5550100/requests/abc123
/deliveryInfos HTTP/1.1
Accept: application/json
```

Example 3–4 shows a sample OneAPI Query Delivery Status response.

Example 3–4 OneAPI Query Delivery Status Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"deliveryInfoList": {
    "deliveryInfo": [
        {   "address": "tel:+15415550100",
            "deliveryStatus": "MessageWaiting"},
        {   "address": "tel:+15415550101",
            "deliveryStatus": "MessageWaiting"}],
    "resourceURL": "
http://example.com/oneapi/1/smsmessaging/outbound/tel%3A%2B5550100/requests/abc123
/deliveryInfos "
}}
```

Subscribe to SMS Delivery Notification

The Subscribe to SMS Delivery Notification operation creates a subscription to delivery notifications for an application.

To set up an SMS notification, provide the criteria which will trigger notifications and a **notifyURL** for the delivery of the notifications. The criteria can be a string which, when matched, could be the notification of an SMS received or of a delivery receipt.

The request body contains the correlator for the notification, the **notifyURL** to which the call direction notifications must be sent and, optionally, the **callbackData** (a string to identify the notification).

If the subscription request is successful:

- The response header will contain the URI of the publish/subscribe server.
- A data object associated with the result of the short message operation will be sent to the **notifyURL** address specified in the request body. This data object will contain the appropriate notification (that the message was received or a delivery receipt for the call).

Authorization

Basic or OAuth 2.0

HTTP Method

```
POST
```

URI

```
http://host:port/oneapi/1/smsmessaging/outbound/senderAddress/subscriptions
```

where:

- *host* and *port* are the host name and port of the machine on which Services Gatekeeper is installed.
- *senderAddress* is the address to whom a responding SMS may be sent.

Request Header

The MIME-type for the Content-Type header field can be **application/json**, **application/xml**, or **application/x-www-form-urlencoded**.

Request Body

The request body for the subscription operation accepts the following parameters:

- **notifyURL:** URL. This will be used by the server to POST the notifications to you, so include the URL of your own listener application
- **clientCorrelator:** String. Optional. Uniquely identifies this create subscription request. If there is a communication failure during the request, using the same client correlator when retrying the request allows the operator to avoid creating a duplicate subscription.
- **callbackData**: String. Optional. A function name or other data that you would like included when the POST is received.

Response Header

The Location header field contains the URI of the notification server:

```
http://host:port/oneapi/1/smsmessaging/outbound/subscriptions/subscriptionID
```

where:

- *host* and *port* are the host name and port of the machine on which Services Gatekeeper is installed.
- *subscriptionID* is the reference to the created subscription.

If the request fails, the Status-Line header field will contain the status code and the reason for the failure. See "Errors and Exceptions" for more information.

Response Body

The response body contains a confirmation **deliveryReceiptSubscription** JSON data structure consisting of the parameters supplied in the subscription request.

```
{"deliveryReceiptSubscription": {
    "callbackReference": {
        "callbackData": "String",
        "notifyURL": " www.yourURL.here ",
        "criteria":"String"
    },
    "resourceURL": " URL "}}
```

A **resourceURL** is included as a reference to the response.

Notification Data Object for SMS Delivery Receipt Sent to notifyURL

After a OneAPI SMS subscription is made, Services Gatekeeper delivers a message receipt notification to the **notifyURL** specified in the subscription request.

This nested JSON object contains the following as the value of the attribute name **deliveryInfoNotification**:

- **callbackData:** String. The correlator used to identify the notification.
- **deliveryInfo:** JSON Object. Contains the following two parameters:
  - **address:** String. The message recipient's subscriber ID.
  - **deliveryStatus:** Enumeration value. Table 3–2 lists the possible statuses:

Table 3–2 Enumeration Values for Delivery Status

Value	Description
DeliveredToNetwork	Successful delivery to the network. For concatenated messages, returned only when all the SMS-parts have been successfully delivered to the network.
DeliveryUncertain	Delivery status unknown, for example, if it was handed off to another network.
DeliveryImpossible	Unsuccessful delivery; the message could not be delivered before it expired.

Table 3–2 (Cont.) Enumeration Values for Delivery Status

Value	Description
DeliveredToTerminal	Successful delivery to the terminal. For concatenated messages, returned only when all the SMS-parts have been successfully delivered to the terminal.

The notification data object delivered to the **notifyURL** address is represented by the following JSON data structure, where the value part of each name/value pair indicates its data type:

```
{"deliveryInfoNotification": {
    "callbackData": "String",
    "deliveryInfo": {
        "address": "String",
        "deliveryStatus": "Enumeration Value"},
    }}
```

Examples

Example 3–5 shows a sample OneAPI Subscribe to SMS Delivery Notifications request.

Example 3–5 OneAPI Subscribe to SMS Delivery Notifications Request

```
POST http://example.com/oneapi/1/smsmessaging/outbound/
tel%3A%2B5550100/subscriptions HTTP/1.1
Host: example.com:80
Content-Type: application/x-www-form-urlencoded
Accept: application/json

notifyURL=http://www.oracle.com&
criteria="GIGPICS"&
callbackData=doSomething()
```

Example 3–6 shows a sample OneAPI Subscribe to SMS Delivery Notifications response.

Example 3–6 OneAPI Subscribe to SMS Delivery Notifications Response

```
HTTP/1.1 201 Created
Content-Type: application/json
Location: http://example.com/oneapip/1/smsmessaging/outbound/subscriptions/sub789
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"deliveryReceiptSubscription": {
    "callbackReference": {
        "callbackData": "doSomething()",
        "notifyURL": " www.oracle.com ",
        "criteria":"Urgent"
    },
    "resourceURL": "
http://example.com/oneapi/1/smsmessaging/outbound/subscriptions/sub789 "}}
```

Example 3–7 shows a sample OneAPI SMS Delivery Notification message.

Example 3–7 OneAPI SMS Delivery Notification Message

```
{"deliveryInfoNotification": {
```

```
"callbackData": "12345",
"deliveryInfo": {
    "address": "tel:+15415550100",
    "deliveryStatus": "DeliveredToNetwork"},
}}
```

Stop Subscription to Delivery Notifications

The Stop Subscription to Delivery Notification operation terminates a previously set up SMS notification for the application.

To stop a previously set up SMS notification, provide the correlator for the notification passed earlier in the Subscribe to SMS Delivery Notification request.

There is no request or response body for the Stop Subscription to Delivery Notification operation. If the request fails, the body of the error response will contain the identifier for the notification and the type of exception.

Authorization

Basic or OAuth 2.0

HTTP Method

```
DELETE
```

URI

```
http://host:port/oneapi/1/smsmessaging/outbound/subscriptions/subscriptionID
```

where:

- *host* and *port* are the host name and port of the machine on which Services Gatekeeper is installed.
- *subscriptionID* is the reference to the created subscription.

Request Body

There is no request body.

Response Header

Standard header fields. If the request fails, the Status-Line header field will contain the status code and the reason for the failure. See "Errors and Exceptions" for more information.

Response Body

There is no response body.

Examples

Example 3–8 shows a sample OneAPI Stop Subscription to SMS Delivery Notifications request.

Example 3–8 OneAPI Stop Subscription to Deliver Notification Request

```
DELETE http://example.com/oneapi/1/smsmessaging/outbound/subscriptions/sub789
HTTP/1.1
Accept: application/json
Host: example.com:80
```

Example 3–9 shows a sample OneAPI Stop Subscription to SMS Delivery Notifications response.

Example 3–9 OneAPI Stop Subscription to Deliver Notification Response

```
HTTP/1.1 204 No Content
Date: Thu, 04 Jun 2009 02:51:59 GMT
```

Retrieve Messages Sent to Web Application

The OneAPI Retrieve Messages Sent to Web Application operation polls Services Gatekeeper for the SMSs that have been received from the network for an application.

The request header for the Retrieve Messages SMS operation contains the registration identifier necessary to retrieve the SMSs intended for the application. This registration value should have been set up with the off-line provisioning step that enables the application to receive notification that SMSs have been received.

There is no request body.

If the Retrieve Messages Sent to Web Application operation is successful, the response body will contain the message, the URI of the sender, the SMS service activation number, and the date and time when the message was sent.

Authorization

Basic or OAuth 2.0

HTTP Method

```
GET
```

URI

```
http://host:port/oneapi/1/smsmessaging/inbound/registrations/registrationID/messag
es?maxBatchSize=X
```

where:

- *host* and *port* are the host name and port of the machine on which Services Gatekeeper is installed.
- *registrationId* is the value previously set up to enable the application to receive notification that SMSs have been received according to specified criteria.
- the value of **maxBatchSize** is the maximum number of message to return.

Request Header

The MIME-type for the accept header field is **application/json**.

Request Body

There is no request body.

Response Header

Standard header fields. If the request fails, the Status-Line header field will contain the status code and the reason for the failure. See "Errors and Exceptions" for more information.

Response Body

The response body is an **inboundSMSMessageList** containing an array of structures as the value for **inboundSMSMessage**. The response body also contains the following parameters:

- **inboundSMSMessage:** Array. Contains the following parameters:
  - **dateTime:** dateTime. The date and time when the message was received.
  - **destinationAddress**: String. The number or shortcode for the application.
  - **messageID:** String. A server generated message identifier.
  - **message:** String. The SMS message.
  - **resourceURL:** URL. Link to the message.
  - **senderAddress:** String. The MSISDN of the sender.
- **numberOfMessagesInThisBatch:** Integer. The total number of messages in the batch.
- **resourceURL**: URL Self-referring resource URL.

The response body for this operation is represented by the following JSON data structure, where the value part of each name/value pair indicates its data type:

```
"inboundSMSMessage": [
        {   "dateTime": "dateTime",
            "destinationAddress": "String",
            "messageId": "String",
            "message": "String",
            "resourceURL": "URL",
            "senderAddress": "String"},
```

Examples

Example 3–10 shows a sample OneAPI Retrieve Messages request.

Example 3–10 OneAPI Retrieve Messages Request

```
GET
http://example.com/oneapi/1/smsmessaging/inbound/registrations/3456/messages?maxBa
tchSize=2  HTTP/1.1
Host: example.com:80
Accept: application/json
```

Example 3–11 shows a sample OneAPI Retrieve Messages response.

Example 3–11 OneAPI Retrieve Messages Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"inboundSMSMessageList": {
    "inboundSMSMessage": [
        {   "dateTime": "2009-11-19T12:00:00",
            "destinationAddress": "3456",
            "messageId": "msg1",
            "message": "Let’s Go Sharks!!",
            "resourceURL":
"http://example.com/oneapi/1/smsmessaging/inbound/registrations/3456/messages/msg1
",
            "senderAddress": "+15415550100"},
        {   "dateTime": "2009-11-19T12:00:00",
            "destinationAddress": "3456",
            "messageId": "msg2",
            "message": "Thorton Shoots! He Scores! Goal!",
            "resourceURL": "
http://example.com/oneapi/1/smsmessaging/inbound/registrations/3456/messages/msg2"
,
            "senderAddress": "+15415550100"}
    ],
    "numberOfMessagesInThisBatch": "2",
    "resourceURL":
"http://example.com/oneapi/1/smsmessaging/inbound/registrations/3456/messages",
    "totalNumberOfPendingMessages": "20"}}
```

Subscribe to Notifications of Messages Sent to Application

The Subscribe to Notifications Sent to Application operation creates a subscription to delivery notifications for when an application receives a message.

To set up an application notification, provide the **destinationAddress** that will trigger notifications and a **notifyURL** for the delivery of the notifications. The **destinationAddress** is the MSISDN, or code set up in Services Gatekeeper, to which subscribers may send an SMS to your application.

If the subscription request is successful:

- The response header will contain the URI of the publish/subscribe server.
- The response will contain a **resourceURL** indicating the URI of the newly created subscription.

Authorization

Basic or OAuth 2.0

HTTP Method

```
POST
```

URI

```
http://host:port/oneapi/1/smsmessaging/inbound/subscriptions
```

where:

- *host* and *port* are the host name and port of the machine on which Services Gatekeeper is installed.

Request Header

The MIME-type for the Content-Type header field is **application/x-www-form-urlencoded**.

Request Body

The request body for the subscription operation accepts the following parameters:

- **destinationAddress:** String. The MSISDN, or code agreed upon by the operator, to which subscribers may send an SMS to your application.
- **notifyURL:** URL. Is used by the server to POST the notifications to you, so include the URL of your own listener application.
- **criteria**: String. Optional. Case-insensitive text to match against the first word of the message, ignoring any leading whitespace. This allows you to reuse a short code among various applications, each of which can register its own subscription with different criteria.
- **notificationFormat:** Content Type. Optional. The content type in which notifications will be sent; for OneAPI only JSON is supported.
- **clientCorrelator:** String. Optional. Uniquely identifies this create subscription request. If there is a communication failure during the request, using the same

client correlator when retrying the request allows the operator to avoid creating a duplicate subscription.

- **callbackData**: String. Optional. A function name or other data that you would like included when the POST is received.

Response Header

The Location header field contains the URI of the publish/subscribe server:

```
http://host:port/oneapi/1/smsmessaging/inbound/subscriptions/subscriptionID
```

where:

- *host* and *port* are the host name and port of the machine on which Services Gatekeeper is installed.
- *subscriptionID* is the reference to the created subscription.

If the request fails, the Status-Line header field will contain the status code and the reason for the failure. See "Errors and Exceptions" for more information.

Response Body

The response body contains a confirmation **resourceReference** JSON data structure consisting of the parameters supplied in the subscription request.

```
{"resourceReference": {"resourceURL": "URL"}}
```

The **resourceURL** indicates the URI of the newly created subscription.

Notification Data Object for Application Notification Message Sent to notifyURL

After a OneAPI Application Notification subscription is made, Services Gatekeeper will deliver a message receipt notification to the specified **notifyURL** in the subscription request.

This nested JSON object contains the following parameters as the value of the attribute name **inboundSMSMessageNotification**:

- **callbackData:** String. The correlator used to identify the notification.
- **inboundSMSMessage:** Array. Contains the following parameters:
  - **dateTime:** dateTime. The date and time when the message was received.
  - **destinationAddress:** String. The number or shortcode for the application.
  - **messageID:** String. A server-generated message identifier.
  - **message:** String. The SMS message.
  - **resourceURL:** URL. A link to the message.
  - **senderAddress:** String. The MSISDN of the sender.

The notification data object delivered to the **notifyURL** address is represented by the following JSON data structure, where the value part of each name/value pair indicates its data type:

```
{"inboundSMSMessageNotification": {
    "callbackData": "String",
    "inboundSMSMessage": {
        "dateTime": "dateTime",
        "destinationAddress": "String",
        "messageId": "String",
        "message": "String",
        "senderAddress": "String"
    }
}}
```

Examples

Example 3–12 shows a sample OneAPI Subscribe to Notifications of Messages Sent to Applications request.

Example 3–12 OneAPI Subscribe to Notifications of Messages Sent to Applications Request

```
POST http://example.com/oneapi/1/smsmessaging/inbound/subscriptions HTTP/1.1
Host: example.com:80
Content-Type: application/x-www-form-urlencoded
Accept: application/json

destinationAddress=3456&
notifyURL=http://www.yoururl.here/notifications/DeliveryInfoNotification&
criteria=Vote&
notificationFormat=JSON&
callbackData=doSomething()&
clientCorrelator=12345
```

Example 3–13 shows a sample OneAPI Subscribe to Notifications of Messages Sent to Applications response.

Example 3–13 OneAPI Subscribe to Notifications of Messages Sent to Application Response

```
HTTP/1.1 201 Created
Content-Type: application/json
Location: http://example.com/oneapi/1/smsmessaging/inbound/subscriptions/sub678
Content-Length: 254
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"resourceReference": {"resourceURL":
"http://example.com/oneapi/1/smsmessaging/inbound/subscriptions/sub678"}}
```

Example 3–14 shows a sample OneAPI Application Notification Message.

Example 3–14 OneAPI Application Notification Message

```
{"inboundSMSMessageNotification": {
    "callbackData": "12345",
    "inboundSMSMessage": {
        "dateTime": "2009-11-19T12:00:00",
        "destinationAddress": "3456",
        "messageId": "mes1234",
        "message": "Vote for Mega Boy Band",
        "senderAddress": "+15415550100"
    }
}}
```

Stop Subscription to Application Message Notifications

The Stop Subscription to Application Message Notification operation terminates a previously set up application message notification subscription.

To stop a previously set up subscription, provide the correlator for the notification passed earlier in the Subscribe to Notifications of Messages Sent to Application request.

There is no request or response body for the Stop Subscription to Notifications of Messages Sent to Application operation. If the request fails, the body of the error response will contain the identifier for the notification and the type of exception.

Authorization

Basic or OAuth 2.0

HTTP Method

```
DELETE
```

URI

```
http://host:port/oneapi/1/smsmessaging/inbound/subscriptions/subscriptionID
```

where:

- *host* and *port* are the host name and port of the machine on which Services Gatekeeper is installed.
- *subscriptionID* is the reference to the created subscription.

Request Body

There is no request body.

Response Header

Standard header fields. If the request fails, the Status-Line header field will contain the status code and the reason for the failure. See "Errors and Exceptions" for more information.

Response Body

There is no response body.

Examples

Example 3–15 shows a sample OneAPI Stop Subscription to Notifications of Messages Sent to Application request.

Example 3–15 OneAPI Stop Subscription to Notifications of Messages Sent to Application Request

```
DELETE http://example.com/oneapi/1/smsmessaging/inbound/subscriptions/sub123
HTTP/1.1
Accept: application/json
Host: example.com:80
```

Example 3–16 shows a sample OneAPI Stop Subscription to Notifications of Messages Sent to Application response.

Example 3–16 OneAPI Stop Subscription to Notifications of Messages Sent to Application Response

```
HTTP/1.1 204 No content
Accept: application/json
Date: Thu, 04 Jun 2009 02:51:59 GMT
```

4

OneAPI Multimedia Messaging

This chapter describes Oracle Communications Services Gatekeeper OneAPI (OneAPI) Multimedia Messaging (MMS) operations supported by the Oracle Communications Services Gatekeeper RESTful Facade.

About the OneAPI Multimedia Messaging Interface

Applications use the RESTful OneAPI MMS interface to send MMSs, to fetch MMSs and delivery status reports, and to start and stop notifications.

When the request body for an MMS operation contains a request for a delivery receipt, the application provides a **notifyURL** correlator for the message being sent and includes an endpoint address for returning the delivery notification.

The Services Gatekeeper OneAPI MMS interface complies with Open Mobile Alliance (OMA) specifications. For more information on OMA, see:

http://technical.openmobilealliance.org/Technical/Comments.aspx

The information provided in this document is based on the OneAPI specification and is provided here for convenience.

REST Service Descriptions Available at Run-time

When the Administration Server for your Services Gatekeeper domain is in the running state, the REST service descriptions of these operations can be found at

```
http://host:port/oneapi/1/messaging/application.wadl
```

where *host* and *port* are the host name and port of the machine on which the Services Gatekeeper Access Tier is installed.

Sending MMS Messages

To send a MMS message, provide the OneAPI-formatted URI of the addresses which must receive the message in the request body. If the sender requires a delivery receipt, specify the required parameters for the receipt.

If the Send MMS operation is successful, the response will contain the request identifier in the response body for this operation).

If the application requires a receipt for delivery of the message, the application can provide the **notifyURL** in the message body, to which notifications are to be sent.

Authorization

Basic or OAuth 2.0

HTTP Method

```
POST
```

URI

```
http://host:port/oneapi/1/messaging/outbound/senderAddress/requests
```

where:

- *host* and *port* are the host name and port of the machine on which the Services Gatekeeper Access Tier is installed.
- The *senderAddress* is the subscriber for which the message is being sent.

Request Header

The MIME-type for the Content-Type header field is **multipart/form-data**.

Request Body

The request body for the OneAPI Send MMS operation accepts the following parameters:

- **address:** String. At least one address is the (optionally) URL-escaped enduser ID; in this case the MSISDN including the '**tel:**' protocol identifier and the country code optionally preceded by '**+**'. For example, **tel:+15415550100**.
- **message:** String. Must be URL-escaped as per RFC 1738.
- **senderAddress**: String. The address to whom a responding MMS may be sent.

  senderName: String. The URL-escaped name of the sender to appear on the terminal. This is the address to whom a responding MMS may be sent.
- **clientCorrelator:** String. Optional. Uniquely identifies this create MMS request. If there is a communication failure during the request, using the same client correlator when retrying the request allows the operator to avoid sending the same MMS twice.
- **notifyURL:** anyURL. The URL-escaped URL to which you want a notification of delivery sent. The format of this notification is shown in Example 4–1.

callbackData: String. will be passed back in this notification, so you can use it to identify the message the receipt relates to or any other useful data, such as a function name.

Response Header

The Location header field contains the URI:

```
http://host:port/oneapi/1/messaging/outbound/senderAddress/requests/requestID
```

where *requestID* is the string identifier returned in the response body.

If the request fails, the Status-Line header field will contain the status code and the reason for the failure. See "Errors and Exceptions" for more information.

Response Body

The body of the response contains the request identifier as the string value for the **resourceReference** attribute. It is the request identifier returned in the **Location** header field of the response message. The application uses this request identifier to retrieve the delivery status for the sent message.

The response body for this operation is represented by the following JSON structure, where the value part of the name/value pair indicates its data type:

```
{"resourceReference": {"resourceURL": "URL"}}
```

Examples

Example 4–1 shows a sample of a OneAPI Send MMS request.

Example 4–1 OneAPI Send MMS Request

```
POST http://example.com/oneapi/1/messaging/outbound/tel%3A%2B5550102/requests
HTTP/1.1
Content-Length: 12345
Content-Type: multipart/form-data;
                     boundary="===============123456==";

MIME-Version: 1.0
Host: www.example.com
Date: Thu, 04 Jun 2009 02:51:59 GMT

--===============123456==
Content-Disposition: form-data; name=”root-fields”
Content-Type: application/x-www-form-urlencoded;
address=tel%3A%2B15415550100&
address=tel%3A%2B15415550101&
senderAddress=tel:%2B5550102&
&senderName=ExampleCompany
Subject=My%20message&
notifyURL=http://example-application.com/notifications/DeliveryInfoNotification/54
311
&callbackData=
&clientCorrelator=123456
--===============123456==
Content-Disposition: form-data; name=”attachments”; filename=”picture.jpg”
Content-Type: image/gif

GIF89a...binary image data...
--===============123456==
```

Example 4–2 shows a sample of a OneAPI Send MMS response.

Example 4–2   OneAPI Send MMS Response

```
HTTP/1.1 201 Created
Content-Type: application/json
Location: http://example.com/oneapi/1/messaging/outbound/
tel%3A%2B5550102/requests/abc123
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"resourceReference": {"resourceURL": " http://example.com/1/messaging/outbound/
tel%3A%2B5550102/requests/abc123"}}
```

Query Delivery Status of MMS Message

The Query Delivery Status operation retrieves the delivery status of a message that was previously sent using the system-generated **requestID** returned when the message was created.

If the Query Delivery Status is successful, the response body will contain the delivery status for each of the addresses contained in the original send MMS request.

Authorization

Basic or OAuth 2.0

HTTP Method

```
GET
```

URI

```
http://host:port/oneapi/1/messaging/outbound/senderAddress/requests/requestID/deli
veryInfos
```

where:

- *host* and *port* are the host name and port of the machine on which the Services Gatekeeper Access Tier is installed.
- *senderAddress* is the address to which a responding message may be sent.
- *requestID* is the identifier returned in the **result** object of the corresponding Send operation.

Request Body

There is no request body.

Response Header

Standard header fields. If the request fails, the Status-Line header field will contain the status code and the reason for the failure. See "Errors and Exceptions" for more information.

Response Body

The response body contains an array of structures as the value for **deliveryInfo**. Each element in the array contains values for the following parameters.

- **address:** String. The telephone number to which the initial message was sent.
- **deliveryStatus**: Enumeration value. Table 4–1 lists the possible statuses:

Table 4–1 Enumeration Values for Delivery Status

Value	Description
DeliveredToNetwork	Successful delivery to the network. For concatenated messages, returned only when all the MMS-parts have been successfully delivered to the network.

Table 4–1 (Cont.) Enumeration Values for Delivery Status

Value	Description
DeliveryUncertain	Delivery status unknown, for example, if it was handed off to another network.
DeliveryImpossible	Unsuccessful delivery; the message could not be delivered before it expired.
DeliveredToTerminal	Successful delivery to the terminal. For concatenated messages, returned only when all the MMS-parts have been successfully delivered to the terminal.
MessageWaiting	The message is still queued for delivery. This is a temporary state, pending transition to one of the preceding states.

A **resourceURL** is included as a reference to the response.

The response body for this operation is represented by the following JSON data structure, where the value part of each name/value pair indicates its data type:

```
{"deliveryInfoList": {
    "deliveryInfo": [
        {   "address": "String",
            "deliveryStatus": "String"},
        {   "address": "String",
            "deliveryStatus": "String"}],
    "resourceURL": "
http://example.com/oneapi/1/messaging/outbound/senderAddress/requests/requestID/de
liveryInfos"
}}
```

Examples

Example 4–3 shows a sample of a OneAPI Query Delivery Status request.

Example 4–3 OneAPI Query Delivery Status Request

```
GET
http://example.com/oneapi/1/messaging/outbound/tel%3A%2B5550100/requests/abc123/de
liveryInfos HTTP/1.1
Accept: application/json
Host: example.com:80
```

Example 4–4 shows a sample of a OneAPI Query Delivery Status response.

Example 4–4 OneAPI Query Delivery Status Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"deliveryInfoList": {
    "deliveryInfo": [
        {
            "address": "tel:15415550101",
            "deliveryStatus": "MessageWaiting"
        },
        {
            "address": "tel:15415550102",
            "deliveryStatus": "MessageWaiting"
        }
    ],
    "resourceURL": "
http://example.com/oneapi/1/messaging/outbound/tel%3A%2B5550100/requests/abc123/de
liveryInfos"
}}
```

Subscribe to MMS Delivery Notification

The Subscribe to MMS Delivery Notification operation creates a subscription to delivery notifications for an application.

To set up a MMS notification, provide a **notifyURL** for the delivery of the notifications. The request body contains the correlator for the notification, the **notifyURL** to which the call direction notifications must be sent and, optionally, the **callbackData** (a string to identify the notification).

If the subscription request is successful:

- The response header will contain the URI of the publish/subscribe server.
- A data object associated with the result of the multimedia message operation will be sent to the **notifyURL** address specified in the request body. This data object will contain the appropriate notification (that the message was received or a delivery receipt for the call).

Authorization

Basic or OAuth 2.0

HTTP Method

```
POST
```

URI

```
http://host:port/oneapi/1/messaging/outbound/senderAddress/subscriptions
```

where:

- *host* and *port* are the host name and port of the machine on which the Services Gatekeeper Access Tier is installed.
- *senderAddress* is the address to whom a responding MMS may be sent.

Request Header

The MIME-type for the Content-Type header field is **application/x-www-form-urlencoded** or **application/json**.

Request Body

The request body for the subscription operation accepts the following parameters:

- **notifyURL**: URL. This will be used by the server to POST the notifications to you, so include the URL of your own listener application.
- **clientCorrelator:** String. Optional. Uniquely identifies this create subscription request. If there is a communication failure during the request, using the same client correlator when retrying the request allows the operator to avoid creating a duplicate subscription.
- **callbackData**: String. Optional. A function name or other data that you would like included when the POST is received.

Response Header

The Location header field contains the URI of the publish/subscribe server:

```
http://host:port/oneapi/1/messaging/outbound/subscriptions/subscriptionID
```

where:

- *host* and *port* are the host name and port of the machine on which the Services Gatekeeper Access Tier is installed.
- *subscriptionID* is the reference to the created subscription.

If the request fails, the Status-Line header field will contain the status code and the reason for the failure. See "Errors and Exceptions" for more information.

Response Body

The response body contains a confirmation **deliveryReceiptSubscription** JSON data structure consisting of the parameters supplied in the subscription request.

```
{"deliveryReceiptSubscription": {
    "callbackReference": {
        "callbackData": "String",
        "notifyURL": " www.yourURL.here ",
    },
    "resourceURL": " URL "}}
```

A **resourceURL** is included as a reference to the response.

Notification Data Object for MMS Delivery Receipt Sent to notifyURL

After a OneAPI MMS subscription is made, Services Gatekeeper delivers a message receipt notification to the **notifyURL** specified in the subscription request.

This nested JSON object contains the following as the value of the attribute name **deliveryInfoList**:

- **deliveryInfo**: Array. Contains the following two parameters:
  - **address:** String. The message recipient's subscriber ID.
  - **deliveryStatus:** Enumeration value. Table 4–2 lists the possible statuses:

Table 4–2 Enumeration Values for Delivery Status

Value	Description
DeliveredToNetwork	Successful delivery to the network. For concatenated messages, returned only when all the MMS-parts have been successfully delivered to the network.
DeliveryUncertain	Delivery status unknown, for example, if it was handed off to another network.
DeliveryImpossible	Unsuccessful delivery; the message could not be delivered before it expired.
DeliveredToTerminal	Successful delivery to the terminal. For concatenated messages, returned only when all the MMS-parts have been successfully delivered to the terminal.

- **link**: Refers to the message URL (that was returned when the message was originally created).

The notification data object delivered to the **notifyURL** address is represented by the following JSON data structure, where the value part of each name/value pair indicates its data type:

```
{"deliveryInfoList": {
    "deliveryInfo": [
        {  "address": "String"
           "deliveryStatus": "String"},
    ],
    "link": {
        "href":   "URL",
        "rel": "String"
    },
    "resourceURL":"URL"
    }}
```

Examples

Example 4–5 shows a sample of a OneAPI Subscribe to MMS Delivery Notifications request.

Example 4–5 OneAPI Subscribe to MMS Delivery Notifications Request

```
POST http://example.com/oneapi/1/messaging/outbound/tel%3A%2B5550100/subscriptions
HTTP/1.1
Accept: application/json
Content-Type: application/json; charset=UTF-8
Host: example.com:80

{"deliveryReceiptSubscription": {
    "callbackReference": {
        "callbackData": " doSomething()",
        "notifyURL": "http://www.oracle.com"
    }
}}
```

Example 4–6 shows a sample of a OneAPI Subscribe to MMS Delivery Notifications response.

Example 4–6 OneAPI Subscribe to MMS Delivery Notifications Response

```
HTTP/1.1 201 Created
Content-Type: application/json
Location: http://example.com/oneapi/1/messaging/outbound/subscriptions/sub789
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"deliveryReceiptSubscription": {
    "callbackReference": {
        "callbackData": "doSomething()",
        "notifyURL": " www.oracle.com "
    },
    "resourceURL": "
http://example.com/oneapi/1/messaging/outbound/subscriptions/sub789"}}
```

Example 4–7 shows a sample of a OneAPI MMS Delivery Notification Message.

Example 4–7 OneAPI MMS Delivery Notification Message

```
{"deliveryInfoList": {
    "deliveryInfo": [
        {  "address": "tel:15415550101",
           "deliveryStatus": "DeliveredToTerminal"},
        {   "address": "tel:15415550102",
            "deliveryStatus": "DeliveredToTerminal"}
    ],
    "link": {
        "href":
"http://example.com/oneapi/1/messaging/outbound/tel%3A%2B5550100/requests/{request
Id)",
        "rel": "OutboundMessageRequest"
    },
    "resourceURL":
http://example.com/oneapi/1/messaging/outbound/tel%3A%2B5550100/requests/req123/De
liveryInfos
 }}
```

Stop Subscription to Delivery Notifications

The Stop Subscription to Delivery Notification operation terminates a previously set up MMS notification for the application.

To stop a previously set up MMS notification, provide the correlator for the notification passed earlier in the Subscribe to MMS Delivery Notification request.

There is no request or response body for the Stop Subscription to Delivery Notification operation. If the request fails, the body of the error response will contain the identifier for the notification and the type of exception.

Authorization

Basic or OAuth 2.0

HTTP Method

```
DELETE
```

URI

```
http://host:port/oneapi/1/messaging/outbound/subscriptions/subscriptionID
```

where:

- *host* and *port* are the host name and port of the machine on which the Services Gatekeeper Access Tier is installed.
- *subscriptionID* is the reference to the created subscription.

Request Body

There is no request body.

Response Header

Standard header fields. If the request fails, the Status-Line header field will contain the status code and the reason for the failure. See "Errors and Exceptions" for more information.

Response Body

There is no response body.

Examples

Example 4–8 shows a sample of a OneAPI Stop Subscription to Delivery Notification request.

Example 4–8 OneAPI Stop Subscription to Deliver Notification Request

```
DELETE http://example.com/oneapi/1/messaging/outbound/subscriptions/sub789
HTTP/1.1
Accept: application/json
Host: example.com:80
```

Example 4–9 shows a sample of a OneAPI Stop Subscription to Delivery Notification response.

Example 4–9 OneAPI Stop Subscription to Deliver Notification Response

```
HTTP/1.1 204 No Content
Date: Thu, 04 Jun 2009 02:51:59 GMT
```

Retrieve Messages Sent to Web Application

The OneAPI Retrieve Messages Sent to Web Application operation polls Services Gatekeeper for the MMSs that have been received from the network for an application.

The request header for the Retrieve Messages Sent to Web application operation contains the registration identifier necessary to retrieve the MMSs intended for the application. This registration value should have been set up with the off-line provisioning step that enables the application to receive notification that MMSs were received.

There is no request body.

If the Retrieve Messages Sent to Web Application operation is successful, the response body will contain the message, the URI of the sender, the MMS service activation number, and the date and time when the message was sent.

Authorization

Basic or OAuth 2.0

HTTP Method

```
GET
```

URI

```
http://host:port/oneapi/1/messaging/inbound/registrations/registrationID/messages?
maxBatchSize=?
```

where:

- *host* and *port* are the host name and port of the machine on which the Services Gatekeeper Access Tier is installed.
- *registrationId* is the value previously set up to enable the application to receive notification that MMSs have been received, according to specified criteria.
- **matchBatchSize** specifies the maximum number of messages to retrieve in the request batch.

Request Header

The MIME-type for the Content-Type header field is **application/x-www-form-urlencoded** or **application/json**.

Request Body

There is no request body.

Response Header

Standard header fields. If the request fails, the Status-Line header field will contain the status code and the reason for the failure. See "Errors and Exceptions" for more information.

Response Body

The response body is an **inboundMessageList** containing an array of structures as the value for **inboundMessage**. The response body also contains the following parameters:

- **inboundMessage:** Array. Contains the following parameters:
  - **dateTime:** dateTime. The date and time when the message was received.
  - **destinationAddress**: String. The registration ID of the application.
  - **messageID:** String. A server generated message identifier.
  - **inboundMMSMessage:** String. Contains the subject of the message, which may determine whether you want to retrieve the entire MMS.
  - **resourceURL:** URL. Link to the message.
  - **senderAddress:** String. The MSISDN of the sender.
- **numberOfMessagesInThisBatch:** Integer. The total number of messages in the batch.
- **resourceURL**: URL Self-referring resource URL.

The response body for this operation is represented by the following JSON data structure, where the value part of each name/value pair indicates its data type:

```
"inboundMessage": [
        {   "dateTime": "dateTime",
            "destinationAddress": "String",
            "messageId": "String",
            "inboundMMSMessage": "String",
            "resourceURL": "URL",
            "senderAddress": "String"},
```

Retrieving Full Messages

If an application needs to retrieve an entire MMS message, a separate GET operation is required. Use the following operation to retrieve a full message, specifying the **messageID** from the **inboundMessageList**.

URI

```
http://host:port/oneapi/1/messaging/inbound/registrations/registrationID/messages/
messageID?resFormat=JSON
```

where:

- *host* and *port* are the host name and port of the machine on which the Services Gatekeeper Access Tier is installed.
- *registrationId* is the value previously set up to enable the application to receive notification that MMSs have been received, according to specified criteria.
- *messageID* specifies the MMS from the **inboundMessageList** to be retrieved.

Request Header

The **resFormat=JSON** portion of the URI ensures that the response content-type is JSON.

Request Body

There is no request body.

Response Header

Standard header fields. If the request fails, the Status-Line header field will contain the status code and the reason for the failure. See "Errors and Exceptions" for more information.

Response Body

The response body is an **inboundMessageList** containing an array of structures as the value for **inboundMessage**. The response body also contains the following parameters:

- **inboundMessage:** Array. Contains the following parameters:
  - **dateTime:** dateTime. The date and time when the message was received.
  - **destinationAddress**: String. The registration ID of the application.
  - **messageID:** String. A server generated message identifier.
  - **inboundMMSMessage:** String. Contains the subject of the message, which may determine whether you want to retrieve the entire MMS.
  - **resourceURL:** URL. Link to the message.
  - **senderAddress:** String. The MSISDN of the sender.
- **numberOfMessagesInThisBatch:** Integer. The total number of messages in the batch.
- **resourceURL**: URL Self-referring resource URL.

The response body for this operation is represented by the following JSON data structure, where the value part of each name/value pair indicates its data type:

```
"inboundMessage": [
        {   "dateTime": "dateTime",
            "destinationAddress": "String",
            "messageId": "String",
            "inboundMMSMessage": "String",
            "resourceURL": "URL",
            "senderAddress": "String"},

====Content Divider====

Attachment(s)
```

Examples

Example 4–10 shows a sample of a OneAPI Retrieve Messages request.

Example 4–10 OneAPI Retrieve Messages Request

```
GET
http://example.com/oneapi/1/messaging/inbound/registrations/3456/messages?maxBatch
Size=2  HTTP/1.1
Host: example.com:80
Accept: application/json
```

Example 4–11 shows a sample of a OneAPI Retrieve Messages response.

Example 4–11 OneAPI Retrieve Messages Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"inboundMessageList": {
    "inboundMessage": [
        {   "dateTime": "2010-11-19T12:00:00",
            "destinationAddress": "6789",
            "inboundMMSMessage": {"subject": "Rock Festival 2010"},
            "messageId": "msg1",
            "resourceURL":
"http://example.com/oneapi/1/messaging/inbound/registrations/3456/messages/msg1",
            "senderAddress": "tel:+5550100" },
        {   "dateTime": "2010-11-19T12:15:00",
            "destinationAddress": "6789",
            "inboundMMSMessage": {"subject": "London Marathon"},
            "messageId": "msg2”,
            "resourceURL": "
http://example.com/oneapi/1/messaging/inbound/registrations/sub789/messages/msg2",
            "senderAddress": "tel:+5550101"
        }
    ],
    "numberOfMessagesInThisBatch": "2",
    "resourceURL": "
http://example.com/1/messaging/inbound/registrations/3456/messages?maxBatchSize=2
",
    "totalNumberOfPendingMessages": "20"
}}
```

Example 4–12 shows a sample of a OneAPI Retrieve Full Message request.

Example 4–12 OneAPI Retrieve Full Message Request

```
GET
http://example.com/oneapi/1/messaging/inbound/registrations/3456/messages/msg1?res
Format=JSON HTTP/1.1
Host: example.com:80
```

Example 4–13 shows a sample of a OneAPI Retrieve Full Messages response.

Example 4–13 OneAPI Retrieve Full Message Response

```
HTTP/1.1 200 OK
Content-Type: multipart/form-data; boundary=”=====12345====”
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT

====12345====
Content-Disposition=multipart/form-data; name=”root-fields”
Content-Type=application/json
Content-Length: nnnn
{"inboundMessage": {
    "dateTime": "2010-11-19T12:00:00",
    "destinationAddress": "6789",
    "messageId": "msg1",
    "inboundMMSMessage": {"subject": Rock Festival 2010"},
    "resourceURL": "
http://example.com/oneapi/1/messaging/inbound/registrations/3456/messages/msg1",
    "senderAddress": " tel:+5550100"
}}


====12345====

Content-Disposition: form-data; name=”attachments”
Content-Type: multipart/mixed; boundary=”====aaabbb”
====aaabbb
Content-Disposition:attachments;filename=”textBody.txt”;
Content-Type: text/plain; charset=UTF-8
Content-Transfer-Encoding: 8 bit

Look at the attached picture
====aaabbb
Content-Disposition:attachments;filename=”image1.gif”;
Content-Type: image/gif
MIME-Version: 1.0
Content-ID: <99334422@example.com>

GIF89a...binary image data...
====12345====
```

Subscribe to Notifications of Messages Sent to Application

The Subscribe to Notifications Sent to Application operation creates a subscription to delivery notifications for when an application receives a message.

To set up an application notification, provide the **destinationAddress** which will trigger notifications and a **notifyURL** for the delivery of the notifications. The destinationAddress is the MSISDN, or code set up in Services Gatekeeper, to which subscribers may send an MMS to your application.

If the subscription request is successful:

- The response header will contain the URI of the publish/subscribe server.
- The response body will contain a **resourceURL** indicating the URI of the newly created subscription.

Authorization

Basic or OAuth 2.0

HTTP Method

```
POST
```

URI

```
http://host:port/oneapi/1/messaging/inbound/subscriptions
```

where:

- *host* and *port* are the host name and port of the machine on which the Services Gatekeeper Access Tier is installed.

Request Header

The MIME-type for the Content-Type header field is **application/json** or **application/x-www-form-urlencoded**.

Request Body

The request body for the subscription operation accepts the following parameters:

- **destinationAddress**: String. The MSISDN, or code agreed upon by the operator, to which people may send an MMS to your application.
- **notifyURL**: URL. This is used by the server to POST the notifications to you, so include the URL of your own listener application.
- **criteria**: String. Optional. Case-insensitive text to match against the first word of the message, ignoring any leading whitespace. This allows you to reuse a short code among various applications, each of which can register its own subscription with different criteria.
- **notificationFormat**: Content Type. Optional. The content type that notifications will be sent; for OneAPI, only JSON is supported.
- **clientCorrelator:** String. Optional. Uniquely identifies this create subscription request. If there is a communication failure during the request, using the same

client correlator when retrying the request allows the operator to avoid creating a duplicate subscription.

- **callbackData**: String. Optional. A function name or other data that you would like included when the POST is received.

Response Header

The Location header field contains the URI of the publish/subscribe server:

```
http://host:port/oneapi/1/messaging/inbound/subscriptions/subscriptionID
```

where:

- *host* and *port* are the host name and port of the machine on which the Services Gatekeeper Access Tier is installed.
- *subscriptionID* is the reference to the created subscription.

If the request fails, the Status-Line header field will contain the status code and the reason for the failure. See "Errors and Exceptions" for more information.

Response Body

The response body contains a confirmation **resourceReference** JSON data structure consisting of the parameters supplied in the subscription request.

```
{"resourceReference": {"resourceURL": "URL"}}
```

The **resourceURL** indicates the URI of the newly created subscription.

Notification Data Object for Application Notification Message Sent to notifyURL

After a OneAPI Application Notification subscription is made, Services Gatekeeper will deliver a message receipt notification to the specified **notifyURL** in the subscription request for every MMS received. If **callbackData** was provided in the subscription request, the notification will contain this information.

This nested JSON object contains the following parameters as the value of the attribute name **inboundMMSMessageNotification**:

- **callbackData:** String. The correlator used to identify the notification.
- **inboundMessage:** Array. Contains the following parameters:
  - **destinationAddress**: String. The number or shortcode for the application.
  - **messageID:** String. A server-generated message identifier.
  - **inboundMMSMessage:** String. Message subject or text.
  - **link**: URL. The URL of the subscription that received this message.
  - **resourceURL**: URL. Link to the message.
  - **senderAddress**: String. The MSISDN of the sender.

The notification data object delivered to the **notifyURL** address is represented by the following JSON data structure, where the value part of each name/value pair indicates its data type:

```
{"inboundMessageNotification": {
    "callbackData": "String",
    "inboundMMSMessage": {
        "destinationAddress": "String",
        "messageId": "String",
        "inboundMMSMessage": "String",
        "link": {
        "href": "String",
        "rel": "String",
        "resourceURL": "String",
        "senderAddress": "String"
    }
}}
```

Examples

Example 4–14 shows a sample of a OneAPI Subscribe to Notifications of Messages Sent to Applications request.

Example 4–14 OneAPI Subscribe to Notifications of Messages Sent to Applications Request

```
POST http://example.com/oneapi/1/messaging/inbound/subscriptions HTTP/1.1
Host: example.com:80
Content-Type: application/x-www-form-urlencoded
Accept: application/json

destinationAddress=3456&
notifyURL=http://www.yoururl.here/notifications/DeliveryInfoNotification&
criteria=Vote&
notificationFormat=JSON&
callbackData=doSomething()&
clientCorrelator=12345
```

Example 4–15 shows a sample of a OneAPI Subscribe to Notifications of Messages Sent to Applications response.

Example 4–15 OneAPI Subscribe to Notifications of Messages Sent to Application Response

```
HTTP/1.1 201 Created
Content-Type: application/json
Location: http://example.com/oneapi/1/messaging/inbound/subscriptions/sub123
Content-Length: 254
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"resourceReference": {"resourceURL": "
http://example.com/oneapi/1/messaging/inbound/subscriptions/sub123”}}
```

Example 4–16 shows a sample of a OneAPI Application Notification Message.

Example 4–16 OneAPI Application Notification Message

```
{"inboundMessageNotification":
{"inboundMessage": {
    "destinationAddress": "tel:+1-555-0100",
    "messageId": "msg123",
    "inboundMMSMessage": {"subject": "That Chili Relleno was delicious?"},
    "link": {
        "href":
"http://example.com/oneapi/1/messaging/inbound/subscriptions/sub123",
        "rel": "Subscription"
    },
    "resourceURL": "http://example.com/oneapi/1/
messaging/inbound/registrations/reg123/messages/msg123",
    "senderAddress": "tel:+5550101"
}}}
```

Stop Subscription to Application Message Notifications

The Stop Subscription to Application Message Notification operation terminates a previously set up subscription to an application message notification.

To stop a previously set up subscription, provide the correlator for the notification passed earlier in the Subscribe to Notifications of Messages Sent to Application request.

There is no request or response body for the Stop Subscription to Notifications of Messages Sent to Application operation. If the request fails, the body of the error response will contain the identifier for the notification and the type of exception.

Authorization

Basic or OAuth 2.0

HTTP Method

```
DELETE
```

URI

```
http://host:port/oneapi/1/messaging/inbound/subscriptions/subscriptionID
```

where:

- *host* and *port* are the host name and port of the machine on which the Services Gatekeeper Access Tier is installed.
- *subscriptionID* is the reference to the created subscription.

Request Body

There is no request body.

Response Header

Standard header fields. If the request fails, the Status-Line header field will contain the status code and the reason for the failure. See "Errors and Exceptions" for more information.

Response Body

There is no response body.

Examples

Example 4–17 shows a sample of a OneAPI Stop Subscription to Notifications of Messages Sent to Application request.

Example 4–17 OneAPI Stop Subscription to Notifications of Messages Sent to Application Request

```
DELETE http://example.com/oneapi/1/messaging/inbound/subscriptions/sub123 HTTP/1.1
Accept: application/json
Host: example.com:80
```

Example 4–18 shows a sample of a OneAPI Stop Subscription to Notifications of Messages Sent to Application response.

Example 4–18 OneAPI Stop Subscription to Notifications of Messages Sent to Application Response

```
HTTP/1.1 204 No content
Accept: application/json
Date: Thu, 04 Jun 2009 02:51:59 GMT
```

5

OneAPI Terminal Location

This chapter describes the operations in the Oracle Communications Services Gatekeeper OneAPI (OneAPI) Terminal Location interface of the Oracle Communications Services Gatekeeper RESTful Web Services.

About the Terminal Location Interface

Applications use the RESTful OneAPI Terminal Location interface to get a location for an individual terminal or a group of terminals.

The Services Gatekeeper OneAPI Location interface complies with Open Mobile Alliance (OMA) specifications. For more information on OMA, see:

`http://technical.openmobilealliance.org/Technical/Comments.aspx`

The information provided in this document is based on the OneAPI specification and is provided here for convenience.

REST Service Descriptions Available at Run-time

When the Administration Server for your Services Gatekeeper domain is in the running state, the REST service descriptions of these operations can be found at

```
http://host:port/oneapi/1/terminallocation/application.wadl
```

where *host* and *port* are the host name and port of the machine on which the Services Gatekeeper Access Tier (AT) services are running.

Query Mobile Terminal Location

The Query Terminal Location operation retrieves the location of one or more terminals.

To retrieve the location of a specific terminal, provide its URI as the address value of the query object in the Request-URI of the GET method. Repeat the address parameter in the request for multiple terminals.

If the Query Terminal Location operation is successful, the response body contains a nested JSON data object containing the physical coordinates of each specific terminal and the date and time for when such data was last collected.

A **locationRetrievalStatus** is also provided for each terminal in the response body.

Authorization

Basic or OAuth 2.0

HTTP Method

```
GET
```

URI

```
http://host:port/oneapi/1/terminallocation/1.0/location/queries/location?query${query}
```

where:

- *host* and *port* are the host name and port of the machine on which the Services Gatekeeper Access Tier (AT) services are running.
- *${query}* contains the following parameters:
  - **address:** String. The MSISDN of the mobile device to locate. Repeat the address parameter for multiple devices. The protocol and '**+**' identifier must be used for MSISDN, and must be URL-escaped. **%3A** represents '**:**' and **%2B** represents '**+**'.
  - **requestedAccuracy:** Integer. The preferred accuracy of the result, in meters. Typically, when you request an accurate location, it will take longer to retrieve than a coarse location. For example, **requestedAccuracy=10** will take longer than **requestedAccuracy=100**.

Request Header

The MIME-type for the Content-Type header field is **application/json**.

Request Body

There is no request body.

Response Header

Standard header fields. If the request fails, the Status-Line header field will contain the status code and the reason for the failure. See "Errors and Exceptions" for more information.

Response Body

The location of the specific terminal is returned in the body of the response as the value of the **terminalLocation** JSON object. For more than one terminal, a **terminalLocationList** consisting of multiple **terminalLocation** objects are returned.

The parameters in this object are:

- **address:** String. Denotes the terminal located (local and international numbers are supported).
- **currentLocation:** JSON Object. Contains the following parameters:
  - **accuracy:** Integer
  - **altitude:** Integer
  - **latitude:** Number (floating point)
  - **longitude**: Number (floating point)
  - **timestamp:** String. The date and time when the terminal's geographical coordinates were collected, given in ISO 8601 extended format, `yyyy-mm-ddThh-mm-ss`.
- **locationRetrievalStatus:** String. Contains one of the possible values:
  - **Retrieved:** Successful retrieval of the terminal location for the address
  - **Not Retrieved:** Services Gatekeeper was unable to locate the terminal location for the address
  - **Error:** A service policy or exception has occurred. See "Errors and Exceptions" for more information.

The response body for this operation is represented by the following JSON data structure, where the value part of the name/value pair indicates its data type. A response containing multiple terminal locations are contained in a **terminalLocationList** structure, as in Example 5–4, "Query Multiple Mobile Terminals Location Response".

```
{"terminalLocation": {
    "address": "String",
    "currentLocation": {
        "accuracy": "Integer",
        "altitude": "Float",
        "latitude": "Float",
        "longitude": "Float",
        "timestamp": "Calendar"
    },
```

Examples

Example 5–1 shows a sample of a OneAPI Single Mobile Terminal Location request.

Example 5–1 Query Single Mobile Terminal Location Request

```
GET
http://example.com/oneapi/1/location/queries/location?&address=tel&3A%2B1541555010
0&requestedAccuracy=1000 HTTP/1.1
Host: example.com:80
Accept: application/json
```

Example 5–2 shows a sample of a OneAPI Single Mobile Terminal response.

Example 5–2   Query Single Mobile Terminal Location Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 1234
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"terminalLocationList": {"terminalLocation": {
    "address": "tel:15415550100",
    "currentLocation": {
        "accuracy": "100",
        "altitude": "1001.0",
        "latitude": "-80.86302",
        "longitude": "41.277306",
        "timestamp": "2009-06-03T00:27:23.000Z"
    },
    "locationRetrievalStatus": "Retrieved"
}}}
```

Example 5–3 shows a sample of a OneAPI Multiple Mobile Terminals Location request.

Example 5–3   Query Multiple Mobile Terminals Location Request

```
GET
http://example.com/oneapi/1/location/queries/location?address=tel%3A%2B15415550100
&address=tel%3A%2B15415550101&requestedAccuracy=1000 HTTP/1.1
Host: example.com:80
Accept: application/json
```

Example 5–4 shows a sample of a OneAPI Multiple Mobile Terminals Location request.

Example 5–4   Query Multiple Mobile Terminals Location Response

```
HTTP/1.1 200 OK
X-Powered-By: Servlet/2.5
Server: Example/v3
Content-Type: application/json
Content-Length: 1234
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"terminalLocationList": {"terminalLocation": [
    {   "address": "tel:15415550100",
        "currentLocation": {
            "accuracy": "100",
            "altitude": "1001.0",
            "latitude": "-80.86302",
            "longitude": "41.277306",
            "timestamp": "2009-06-03T00:27:23.000Z"},
        "locationRetrievalStatus": "Retrieved"},
    {   "address": "tel:15415550101",
        "errorInformation": {
            "messageId": "SVC0001",
            "text": "A service error occurred. %1 %2",
            "variables": ["Location information is not available for",
"tel:15415550101"]
        },
        "locationRetrievalStatus": "Error"
    }
]}}
```

6

OneAPI Payment

This chapter describes the operations in the Payment interface of the Oracle Communications Services Gatekeeper OneAPI (OneAPI) RESTful Web Services.

About the Payment Interface

Applications use the RESTful Payment interface to charge an amount to an end-user's account using Diameter and to refund amounts to that account. Applications can also reserve amounts, reserve additional amounts, charge against the reservation or release the reservation.

The Services Gatekeeper OneAPI Payment interface complies with Open Mobile Alliance (OMA) specifications. For more information on OMA, see:

`http://technical.openmobilealliance.org/Technical/Comments.aspx`

The information provided in this document is based on the OneAPI specification and provided here for convenience.

REST Service Descriptions Available at Run-time

When the Administration Server for your Services Gatekeeper domain is in the running state, the REST service descriptions of these operations can be found at

```
http://host:port/oneapi/1/payment/application.wadl
```

where *host* and *port* are the host name and port of the machine on which the Services Gatekeeper Access Tier (AT) services are running.

Charge Amount

The Charge Amount operation charges an amount directly to an end-user's application using the Diameter protocol.

To charge an amount for a call, provide the telephone address of the end-user (**endUserId**), a reference code (**code**) in case there is any dispute regarding the charges, and the billing information to charge for the call.

Authorization

Basic or OAuth 2.0

HTTP Method

```
POST
```

URI

```
http://host:port/oneapi/1/payment/endUserId/transactions/amount
```

where *host* and *port* are the host name and port of the machine on which the Services Gatekeeper Access Tier (AT) services are running. The *endUserId* is the address of the subscriber to charge (MSISDN).

Request Header

The MIME-type for the Content-Type header field can be either **application/x-www-form-urlencoded** or **application/json**.

Request Body

The request body for the Charge Amount operation accepts the following parameters:

- **endUserId:** String. The URL-escaped end user ID. For example, a MSISDN including the '**tel:**' protocol identifier and the country code preceded by '**+**'.For example, **tel:+15415550100**.
- **transactionOperationStatus:** Enumeration. This indicates the desired resource state, in this case '**charged**'. See "Resource States" for more information.
- **description:** String. The human-readable text to appear on the bill provided so the subscriber can easily see what was purchased.
- **currency:** String. The 3-figure currency code defined in ISO4217.
- **amount**: Decimal. The amount to be charged.
- **code**: String. The charging code, from an existing contractual description that references an operator price point.
- **clientCorrelator**: String. Optional. This uniquely identifies the create charge request. If there is a communication failure during the charge request, using the same client correlator when retrying the request allows the operator to avoid applying the same charge twice.
- **onBehalfOf**: String. Optional. Allows aggregators or partners to specify the actual payee.

- **purchaseCategoryCode**: String. Optional. An indication of the content type. Values meaningful to the billing system would be published by a OneAPI implementation.
- **channel**: String. Optional. Can be “**Wap**”, “**Web**”, “**MMS**”, or “**SMS**”, depending on the source of user interaction.
- **taxAmount:** Decimal. Optional. Indicates a tax amount already charged by the merchant or application.
- **serviceID:** String. Optional. The ID of the partner or merchant service.
- **productID:** String. Optional. Combines with the **serviceID** to uniquely identify the product being purchased.

Response Header

The Location header field contains the URI:

```
http://host:port/oneapi/1/payment/endUserId/transactions/amount/transactionID
```

where *transactionID* is the string identifier returned in the response body.

The Status-Line header field returned indicates if the charge has been created or accepted. An accepted response indicates that additional processing is required before the transaction is complete. If the request fails, the Status-Line header field will contain the status code and the reason for the failure. See "Errors and Exceptions" for more information.

Response Body

The response body contains a confirmation **amountTransaction** JSON data structure consisting of the parameters supplied in the payment request.

```
{"amountTransaction": {
    "clientCorrelator": "String",
    "endUserId": "String",
    "paymentAmount": {
        "chargingInformation": {
            "amount": "Decimal",
           "currency": "String",
            "description": " String"
        },
        "totalAmountCharged": "Decimal"
    },
    "code": "String",
    "resourceURL": "URL"
    "transactionOperationStatus": "String"
}}
```

A **resourceURL** is included as a reference to the response. The **transactionOperationStatus** provides the resource state. See, "Resource States" for more information.

Examples

Example 6–1 shows a sample of a OneAPI Charge Payment request.

Example 6–1 OneAPI Charge Payment Request

```
POST http://example.com/oneapi/1/payment/tel%3A%2B15415550100/transactions/amount
HTTP/1.1
```

```
Accept: application/json
Host: example.com:80
Content-Type: application/x-www-form-urlencoded
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT

endUserId= tel%3A%2B15415550100&
transactionOperationStatus=charged&
description= Alien%20%20Game&
currency=USD&
amount=10&
code=REF-12345&
clientCorrelator=54321&
onBehalfOf=Example%20Games%20Inc&
purchaseCategoryCode=Game&
channel=WAP&
taxAmount=0
```

Example 6–2 shows a sample of a OneAPI Charge Payment response.

Example 6–2 OneAPI Charge Payment Response

```
HTTP/1.1 201 Created
Content-Type: application/json
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT
Location:
http://example.com/1/payment/tel%3A%2B15415550100/transactions/amount/abc123

{"amountTransaction": {
    "clientCorrelator": "54321",
    "endUserId": "tel:+15415550100",
    "paymentAmount": {
        "chargingInformation": {
            "amount": "10",
           "currency": "USD",
            "description": " Alien Game"
        },
        "totalAmountCharged": "10"
    },
    "code": "REF-12345",
    "resourceURL": "
http://example.com/oneapi/1/payment/tel%3A%2B15415550100/transactions/amount/abc12
3
    "transactionOperationStatus": "Charged"
}}
```

Query Transaction Status

The Query Transaction Status operation retrieves the current status of a payment transaction. This operation is useful in cases where the original charge request returns a status of **processing**. Applications may require a confirmed charge to a subscriber's account before allowing access to paid resources.

To query for a transaction's status, provide the **transactionID** of the earlier transaction.

Authorization

Basic or OAuth 2.0

HTTP Method

```
GET
```

URI

```
http://host:port/oneapi/1/payment/endUserId/transactions/amount/transactionID
```

where *host* and *port* are the host name and port of the machine on which the Services Gatekeeper Access Tier (AT) services are running. The *endUserId* is the address of the subscriber originally charged (MSISDN). The *transactionID* is provided by Services Gatekeeper in the original payment request response.

Request Header

The header contains the **Host** and **Authorization** code (if required).

Request Body

There is no request body for this operation.

Response Header

The header contains the response status, **Content-Type**, **Content-Length** and **Date**.

If the request fails, the **Status-Line** header field will contain the status code and the reason for the failure. See "Errors and Exceptions" for more information.

Response Body

The response body contains the transaction's **amountTransaction** JSON data structure consisting of the parameters supplied in the payment request.

```
{"amountTransaction": {
    "clientCorrelator": "String",
    "endUserId": "String",
    "paymentAmount": {
        "chargingInformation": {
            "amount": "Decimal",
           "currency": "String",
            "description": " String"
        },
        "totalAmountCharged": "Decimal"
    },
    "code": "String",
    "resourceURL": "URL"
    "transactionOperationStatus": "String"
}}
```

A **resourceURL** is included as a reference to the response. The **transactionOperationStatus** provides the current resource state. See "Resource States" for more information.

Examples

Example 6–3 shows a sample of a OneAPI Query Transaction Status request.

Example 6–3 OneAPI Query Transaction Status Request

```
GET http://example.com/1/payment/tel%3A%2B15415550100/transactions/amount/abc123
Host: example.com
Authorization: n0t4fr4id333
```

Example 6–2 shows a sample of a OneAPI Query Transaction Status response.

Example 6–4 OneAPI Query Transaction Status Response

```
HTTP/1.1 201 Created
Content-Type: application/json
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT
Location: 
http://example.com/1/payment/tel%3A%2B15415550100/transactions/amount/abc123

{"amountTransaction": {
    "clientCorrelator": "54321",
    "endUserId": "tel:+15415550100",
    "paymentAmount": {
        "chargingInformation": {
            "amount": "10",
           "currency": "USD",
            "description": " Alien Game"
        },
        "totalAmountCharged": "10"
    },
    "code": "REF-12345",
    "resourceURL": "
http://example.com/oneapi/1/payment/tel%3A%2B15415550100/transactions/amount/abc12
3
    "transactionOperationStatus": "Charged"
}}
```

List Transactions for Application User

The List Transactions for Application User operation lists all the transactions a subscriber has performed within an application.

To list a subscriber's transactions for an application, provide a valid OAuth token provided by Services Gatekeeper to identify the application and subscriber.

The response body contains a list of applicable transactions and their current status.

Authorization

Basic or OAuth 2.0

HTTP Method

```
GET
```

URI

```
http://host:port/oneapi/1/payment/endUserId/transactions/amount
```

where *host* and *port* are the host name and port of the machine on which the Services Gatekeeper Access Tier (AT) services are running. The *endUserId* is the address of the subscriber to charge (MSISDN).

Request Header

Provide a valid OAuth bearer token to identify both the end user and the application in the **Authorization** header field.

Request Body

There is no request body for this operation.

Response Header

The Location header field contains the URI:

```
http://host:port/oneapi/1/payment/endUserId/transactions/amount/transactionID
```

where *transactionID* is the string identifier returned in the response body.

If the request fails, the Status-Line header field will contain the status code and the reason for the failure. See "Errors and Exceptions" for more information.

Response Body

The response body contains a list of **amountTransaction** JSON data structures consisting of the parameters supplied in the payment requests.

```
{"amountTransaction": {
    "clientCorrelator": "String",
    "endUserId": "String",
    "paymentAmount": {
        "chargingInformation": {
            "amount": "Decimal",
           "currency": "String",
            "description": " String"
        },
        "totalAmountCharged": "Decimal"
    },
    "code": "String",
    "resourceURL": "URL"
    "transactionOperationStatus": "String"
}}
```

A **resourceURL** is included as a reference to the response for each transaction. The **transactionOperationStatus** provides the resource state. See "Resource States" for more information.

Examples

Example 6–5 shows a sample of a OneAPI List Transactions for Application User request.

Example 6–5 OneAPI List Transactions for Application User Request

```
GET http://example.com/1/payment/tel%3A%2B15415550100/transactions/amount
HTTP/1.1
Accept: application/json
Authorization: n0t4fr4id333
Date: Thu, 06 Feb 1976 02:51:59 GMT
```

Example 6–6 shows a sample of a OneAPI List Transactions for Application User response.

Example 6–6 OneAPI List Transactions for Application User Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT
{"paymentTransactionList": {
"amountTransaction": [
   {
      "endUserId": "tel:+15415550100",
      "paymentAmount": {
        "chargingInformation": {
             "amount": "9",
            "currency": "USD",
            "code": "abc123",
            "description": "Alien%20Invaders"
        },
      },
      "code": "REF-ASM600-239238",
      "resourceURL":
"https://example.com/1/payment/tel%3A%2B15415550100/transactions/amount/tx-a3c0e4e
006da40a8a5b5-045972478cc3",
      "transactionOperationStatus": "Charged"
   },
   {
      "endUserId": "tel:+15415550100",
      "paymentAmount": {
        "chargingInformation": {
            "amount": "9",
            "currency": "USD",
            "code": "def456",
            "description": "Snakes%20Alive"
        },
      },
      "code": "REF-ASM600-239568",
      "resourceURL":
"https://example.com/1/payment/tel%3A%2B15415550100/transactions/amount/tx-134sf3e
6e6405gfd904e62d8ed84343u",
      "transactionOperationStatus": "Charged"
   },
   {
      "endUserId": "tel:+15415550100",
      "paymentAmount": {
        "chargingInformation": {
           "amount": "9",
            "currency": "USD",
            "code": "wac-c116480e-316a-44e7-be76-5fde978b2f59",
            "description": "Monkey%20Tennis."
        },
      },
      "code": "REF-ASM600-239211",
      "resourceURL":
"https://example.com/1/payment/tel%3A%2B15415550100/transactions/amount/tx-391sff4
e6401gf3f404d82d9fe954545v",
      "transactionOperationStatus": "Charged"
  },
],
"resourceURL":
"https://example.com/1/payment/tel%3A%2B15415550100/transactions/amount"
}}
```

Refund Amount

The Refund Amount operation refunds a currency amount directly to a subscriber's application using the Diameter protocol.

To refund an amount to a subscriber, submit a `POST` operation request in the same format as the Charge Amount operation.

Authorization

Basic or OAuth 2.0

HTTP Method

```
POST
```

URI

```
http://host:port/oneapi/1/payment/endUserId/transactions/amount
```

where *host* and *port* are the host name and port of the machine on which the Services Gatekeeper Access Tier (AT) services are running. The *endUserId* is the address of the subscriber to refund (MSISDN).

Request Header

The MIME-type for the Content-Type header field can be either **application/x-www-form-urlencoded** or **application/json**.

Request Body

The request body for the Refund Amount operation accepts the following parameters:

- **endUserId:** String. The URL-escaped end user ID. For example, a MSISDN including the '**tel**:' protocol identifier and the country code preceded by '**+**'. For example, **tel:+15415550100**.
- **transactionOperationStatus:** Enumeration.This indicates the desired resource state, in this case '**refunded**'. See "Resource States" for more information.
- **description:** String. The human-readable text to appear on the bill, so the subscriber can easily see what was refunded.
- **currency:** String. The 3-figure currency code defined in ISO4217.
- **amount**: Decimal. The amount to be refunded.
- **code:** String. The charging code, from an existing contractual description that references an operator price point.
- **clientCorrelator**: String. Uniquely identifies this refund request. If there is a communication failure during the refund request, using the same client correlator when retrying the request allows the operator to avoid applying the same refund twice.
- **onBehalfOf**: String. Optional. Allows aggregators or partners to specify the actual payee.

- **purchaseCategoryCode**: String. Optional. An indication of the content type. Values meaningful to the billing system would be published by a OneAPI implementation.
- **channel**: String. Optional. Can be “**Wap**”, “**Web**”, “**MMS**”, or “**SMS**”, depending on the source of user interaction.
- **taxAmount:** Decimal. Optional. Indicates a tax amount already charged by the merchant or application.
- **serviceID:** String. Optional. The ID of the partner or merchant service.
- **productID:** String. Optional. Combines with the **serviceID** to uniquely identify the product being refunded.

Response Header

The Location header field contains the URI:

```
http://host:port/oneapi/1/payment/endUserId/transactions/amount/transactionID
```

where, *transactionID* is the string identifier returned in the response body.

If the request fails, the Status-Line header field will contain the status code and the reason for the failure. See "Errors and Exceptions" for more information.

Response Body

The response body contains a confirmation **amountTransaction** JSON data structure consisting of the parameters supplied in the refund request.

```
{"amountTransaction": {
    "clientCorrelator": "String",
    "endUserId": "String",
    "paymentAmount": {
        "chargingInformation": {
            "amount": "Decimal",
           "currency": "String",
            "description": " String"
        },
        "totalAmountRefunded": "Decimal"
    },
    "code": "String",
    "resourceURL": "URL"
    "transactionOperationStatus": "String"
}}
```

A **resourceURL** is included as a reference to the response. The **transactionOperationStatus** provides the resource state. See "Resource States" for more information.

Examples

Example 6–7 shows a sample of a OneAPI Refund request.

Example 6–7   OneAPI Refund Request

```
POST http://example.com/oneapi/1/payment/tel%3A%2B15415550100/transactions/amount
HTTP/1.1
Accept: application/json
Host: example.com:80
Content-Type: application/x-www-form-urlencoded
```

```
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT

endUserId=tel%3A%2B15415550100&
transactionOperationStatus=refunded&
description= Alien%20Game&
currency=USD&
amount=10&
code=REF-12345&
clientCorrelator=54321&
onBehalfOf=Example%20Games%20Inc&
purchaseCategoryCode=Game&
channel=WAP&
taxAmount=0
```

Example 6–8 shows a sample of a OneAPI Refund response.

Example 6–8   OneAPI Refund Response

```
HTTP/1.1 201 Created
Content-Type: application/json
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT
Location:
http://example.com/oneapi/1/payment/tel%3A%2B15415550100/transactions/amount/efg78
9

{"amountTransaction": {
    "clientCorrelator": "54321",
    "endUserId": "tel:+15415550100",
    "paymentAmount": {
        "chargingInformation": {
            "amount": "10",
            "currency": "USD",
            "description": "Alien Invaders"
        },
        "totalAmountRefunded": "10"
    },
    "code": "REF-12345",
    "resourceURL":
"http://example.com/oneapi/1/payment/tel%3A%2B15415550100/transactions/amount/efg7
89",
    "transactionOperationStatus": "Refunded"
}}
```

Reserve Amount

The Reserve Amount operation reserves a currency amount for a subscriber account to use as future payment.

To reserve an amount for future payment, provide the address of the enduser and the billing information for the transaction.

If the Reserve Amount operation successful, the response body will contain the string identifier for the reservation.

The default expiration or timeout of reserved amounts in Services Gatekeeper is **86400** seconds. To change the timeout duration to another value:

1. Locate and unpack the following .jar file:

   *OCSG_Domain***config****store_schema****com.bea.wlcp.wlng.plugin.payment.px30.diameter.store_5.1.0.0.jar**
2. Open the **wlng-cachestore-config-extensions.xml** file for editing.
3. Locate the **<expiry expiry_age="86400"/>** parameter.
4. Edit the value by entering the number of seconds before a reservation expires.
5. Save the file.
6. Repackage the .jar file in the original location.
7. Restart the instance of Services Gatekeeper.

Changing the default expiration setting also changes the Parlay X payment reservation timeout in Services Gatekeeper.

Authorization

Basic or OAuth 2.0

HTTP Method

```
POST
```

URI

```
http://host:port/oneapi/1/payment/endUserId/transactions/amountReservation
```

where *host* and *port* are the host name and port of the machine on which the Services Gatekeeper Access Tier (AT) services are running. The *endUserId* is the address of the subscriber to charge the reservation to (MSISDN).

Request Header

The MIME-type for the Content-Type header field can be either **application/x-www-form-urlencoded** or **application/json**.

Request Body

The request body for the Reserve Amount operation accepts the following parameters:

- **endUserId**: String. The URL-escaped enduser ID. For example, a MSISDN including the '**tel:**' protocol identifier and the country code preceded by '**+**'. For example, **tel:+15415550100**.

- **transactionOperationStatus:** Enumeration. This indicates the desired resource state, in this case '**reserved**'. See "Resource States" for more information.
- **description:** String. The human-readable text to appear on the bill, so the user can easily see what they reserved.
- **currency:** String. The 3-figure currency code defined in ISO4217.
- **amount:** Decimal. The amount to be reserved.
- **code**: String. The charging code, from an existing contractual description that references an operator price point.
- **clientCorrelator:** String. Uniquely identifies this create reservation request. If there is a communication failure during the reservation request, using the same client correlator when retrying the request allows the operator to avoid applying the same reservation twice.
- **referenceSequence:** Integer. Allows Services Gatekeeper to distinguish easily between new and repeated requests in the case of a communication failure. For each transaction within a reservation sequence, iterate the referenceSequence by 1. For example:

  initial reservation use **referenceSequence**=1

  reserve additional amount use **referenceSequence**=2

  charge reservation use **referenceSequence**=3

  release reservation use **referenceSequence**=4

  If you do not need to reserve an additional amount, then the **referenceSequence** for charge reservation is 2 and the **referenceSequence** for release reservation is 3.
- **onBehalfOf**: String. Optional. Allows aggregators or partners to specify the actual payee.
- **purchaseCategoryCode:** String. Optional. An indication of the content type. Values meaningful to the billing system would be published by a OneAPI implementation.
- **channel:** String. Optional. Can be "**Wap**", "**Web**", "**MMS**", or "**SMS**", depending on the source of user interaction.
- **taxAmount**: Decimal. Optional. Indicates a tax amount already charged by the merchant or application.
- **serviceID:** String. Optional. The ID of the partner or merchant service.
- **productID**: String. Optional. Combines with the **serviceID** to uniquely identify the product being reserved.

Response Header

The Location header field contains the URI:

```
http://host:port/oneapi/1/payment/endUserId/transactions/amountReservation/transac
tionID
```

where, *transactionID* is the string identifier returned in the response body.

If the request fails, the Status-Line header field will contain the status code and the reason for the failure. See "Errors and Exceptions" for more information.

Response Body

The response body contains a confirmation **amountReservationTransaction** JSON data structure consisting of the parameters supplied in the reservation request.

```
{"amountReservationTransaction": {
    "clientCorrelator": "String",
    "endUserId": "String",
    "paymentAmount": {
        "chargingInformation": {
            "amount": "Decimal",
           "currency": "String",
            "description": " String"
        },
    },
    "code": "String",
    "resourceURL": "URL"
    "transactionOperationStatus": "String"
}}
```

A **resourceURL** is included as a reference to the response. The **transactionOperationStatus** provides the resource state. See "Resource States" for more information.

Reserving Additional Amount

The Reserve Amount operation is used to update an existing reservation to request additional funds or resources. See "Charge Amount" for more information. Change the following parameters in the request for additional funds:

- **transactionOperationStatus=reserved**: Indicates that we are not changing the resource state, just the value being reserved. See "Resource States" for more information.
- **amount**: The additional reserved amount for this request (not the total amount reserved so far).
- **referenceSequence**: Each time you reserve an additional amount against an existing reservation, make sure to iterate the **referenceSequence** each time. This ensures Services Gatekeeper can distinguish between new requests for additional amounts, and those that are being repeated due to a communication failure.

Examples

Example 6–9 shows a sample of a OneAPI Reserve Amount request.

Example 6–9 OneAPI Reserve Amount Request

```
POST
http://example.com/oneapi/1/payment/tel%3A%2B15415550100/transactions/amountReserv
ation HTTP/1.1
Accept: application/json
Host: example.com:80
Content-Type: application/x-www-form-urlencoded
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT

endUserId= tel%3A%2B15415550100&
transactionOperationStatus=reserved&
description= Streaming%20video%20of%20the%Big%20Fight&
currency=USD&
```

```
amount=10&
referenceCode=Video-abc123&
clientCorrelator=54321&
referenceSequence=1&
onBehalfOf=Example%20Video%20Inc&
purchaseCategoryCode=Video&
channel=WAP&
taxAmount=0
```

Example 6–10 shows a sample of a OneAPI Reserve Amount response.

Example 6–10 OneAPI Reserve Amount Response

```
HTTP/1.1 201 Created
Content-Type: application/json
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT
Location:
http://example.com/oneapi/1/payment/tel%3A%2B15415550100/transactions/amountReserv
ation/abc123

{"amountReservationTransaction": {
    "clientCorrelator": "54321",
    "endUserId": "tel:+15415550100",
    "paymentAmount": {"chargingInformation": {
        "amount": "10",
        "currency": "USD",
        "description": "Streaming video of the Big Fight"
    }},
    "code": "REF-12345",
    "referenceSequence": "1",
    "transactionOperationStatus": "Reserved"
}}
```

Example 6–11 shows a sample of a OneAPI Reserve Additional Amount request.

Example 6–11 OneAPI Reserve Additional Amount Request

```
POST
http://example.com/oneapi/1/payment/tel%3A%2B15415550100/transactions/amountReserv
ation/abc123 HTTP/1.1
Accept: application/json
Host: example.com:80
Content-Type: application/x-www-form-urlencoded
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT

transactionOperationStatus=reserved&
amount=5&
code=REF-12346&
referenceSequence=2&
```

Example 6–12 shows a sample of a OneAPI Reserve Additional Amount response.

Example 6–12 OneAPI Reserve Additional Amount Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT
```

```
{"amountReservationTransaction": {
    "endUserId": "tel:+15415550100",
    "paymentAmount": {
        "amountReserved": "15",
        "chargingInformation": {
            "amount": "10",
            "currency": "USD",
            "description": "Streaming Video of the Big Fight"
        },
        "totalAmountCharged": "0"
    },
    "code": "REF-12346",
    "referenceSequence": "2",
    "resourceURL": "
http://example.com/oneapi/1/payment/tel%3A%2B15415550100/transactions/amountReserv
ation/abc123",
    "transactionOperationStatus": "Reserved"
}}
```

Charge Reservation

The Charge Reservation operation charges a previously reserved amount against a subscriber’s account.

To charge a previously reserved amount to a subscriber’s account, provide the information for billing, the reservation identifier obtained from the initial request to reserve funds, and the reference code for any possible disputes.

Authorization

Basic or OAuth 2.0

HTTP Method

POST

URI

```
http://host:port/oneapi/1/payment/endUserId/transactions/amountReservation/transac
tionID
```

where *host* and *port* are the host name and port of the machine on which the Services Gatekeeper Access Tier (AT) services are running. The *endUserId* is the address of the subscriber to charge (MSISDN). The *transactionID* is the unique value generated by Services Gatekeeper from the original reservation request.

Request

Change the following parameters in the reservation request to charge against a reservation:

- **transactionOperationStatus=charged**: Indicates a change in the resource state to charge against the reservation. See "Resource States" for more information.
- **amount**: The total amount to charge against the reservation.
- **referenceSequence**: Iterate the **referenceSequence** with the charge request. This ensures the OneAPI server can distinguish between new requests for charging against a reservation, and those that are being repeated due to a communication failure.
- **description**: Update the description to reflect the completed transaction.

The **clientCorrelator** parameter is not used as the resource has already been created.

Response Header

If the request fails, the Status-Line header field will contain the status code and the reason for the failure. See "Errors and Exceptions" for more information.

Response Body

The response body contains a confirmation **amountReservationTransaction** JSON data structure consisting of the parameters supplied in the charge request.

```
{"amountReservationTransaction":
    "endUserId": "String",
    "paymentAmount": {
      "amountReserved":"Decimal",
        "chargingInformation": {
            "amount": "Decimal",
           "currency": "String",
            "description": " String"
        },
        "totalAmountCharged": "Decimal"
    },
    "code": "String",
    "resourceURL": "URL"
    "transactionOperationStatus": "String"
}
```

Once the charge has been applied, the **amountReserved** parameter should have a value of zero. The **totalAmountCharged** parameter contains the final amount to charge against the reservation.

A **resourceURL** is included as a reference to the response. The **transactionOperationStatus** provides the resource state. See "Resource States" for more information.

Examples

Example 6–13 shows a sample of a OneAPI Charge Amount Against Reservation request.

Example 6–13 OneAPI Charge Amount Against Reservation Request

```
POST
http://example.com/1/payment/tel%3A%2B15415550100/transactions/amountReservation/a
bc123  HTTP/1.1
Accept: application/json
Host: example.com:80
Content-Type: application/x-www-form-urlencoded
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT

transactionStatus=charged&
description= Three%20rounds%20of%20the%Big%20Fight&
amount=15&
code=REF-123457&
referenceSequence=3&
```

Example 6–14 shows a sample of a OneAPI Charge Amount Against Reservation response.

Example 6–14 OneAPI Charge Amount Against Reservation Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"amountReservationTransaction": {
    "endUserId": "tel:+15415550100",
    "paymentAmount": {
        "amountReserved": "0",
        "chargingInformation": {
            "amount": "15",
            "currency": "USD",
            "description": " Streaming Video of the Big Fight "
        },
        "totalAmountCharged": "15"
    },
    "code": "REF-123457",
    "referenceSequence": "3",
    "resourceURL": "
http://example.com/oneapi/1/payment/tel%3A%2B15415550100/transactions/amountReserv
ation/abc123  ",
    "transactionOperationStatus": "Charged"
}}
```

Release Reservation

The Release Reservation operation returns funds left in a reservation to the subscriber account against which the reservation was made.

To return funds left in a reservation to an account, provide the *transactionID* identifier obtained from the initial reservation request.

Authorization

Basic or OAuth 2.0

HTTP Method

```
POST
```

URI

```
http://host:port/oneapi/1/payment/endUserId/transactions/amountReservation/transactionID
```

where *host* and *port* are the host name and port of the machine on which the Services Gatekeeper Access Tier (AT) services are running.The *endUserId* is the address of the subscriber to charge (MSISDN). The *transactionID* is the unique value generated by Services Gatekeeper from the original reservation request.

Request

Change the following parameters in the reservation request to release a reservation:

- **transactionOperationStatus=released**: Indicates a change in the resource state to release the reservation. See "Resource States" for more information.
- **referenceSequence**: Iterate the **referenceSequence** when releasing a reservation. This ensures the OneAPI server can distinguish between new requests for reservation release, and those that are being repeated due to a communication failure.

Response Header

If the request fails, the Status-Line header field will contain the status code and the reason for the failure. See "Errors and Exceptions" for more information.

Response Body

The response body contains a confirmation **amountReservationTransaction** JSON data structure consisting of the parameters supplied in the release reservation request.

```
{"amountReservationTransaction": {
   "endUserId": "String",
   "paymentAmount": {
     "amountReserved":"Decimal",
       "chargingInformation": {
           "amount": "Decimal",
          "currency": "String",
           "description": " String"
       },
       "totalAmountCharged": "Decimal"
```

```
    },
    "code": "String",
    "resourceURL": "URL"
    "transactionOperationStatus": "String"
}
```

Once the resource reservation has been released the **amountReserved** parameter should have a value of zero. The **totalAmountCharged** parameter contains the final amount to charge against the reservation.

A **resourceURL** is included as a reference to the response. The **transactionOperationStatus** provides the resource state. See "Resource States" for more information.

Examples

Example 6–15 shows a sample of a OneAPI Release Reservation request.

Example 6–15 OneAPI Charge Release Reservation Request

```
POST
http://example.com/oneapi/1/payment/tel%3A%2B15415550100/transactions/amountReserv
ation/abc123  HTTP/1.1
Accept: application/json
Host: example.com:80
Content-Type: application/x-www-form-urlencoded
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT

transactionOperationStatus=released&
referenceSequence=4&
```

Example 6–16 shows a sample of a OneAPI Release Reservation response.

Example 6–16 OneAPI Release Reservation Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"amountReservationTransaction": {
    "endUserId": "tel:+15415550100",
    "paymentAmount": {
        "amountReserved": "0",
        "chargingInformation": {
            "amount": "10",
            "currency": "USD",
            "description": "Streaming Video of the Big Fight"
        },
        "totalAmountCharged": "5"
    },
    "code": "REF-12346",
    "referenceSequence": "4",
    "resourceURL": "
http://example.com/oneapi/1/payment/tel%3A%2B15415550100/transactions/amountReserv
ation/abc123 ",
    "transactionOperationStatus": "Released"
}}
```

Resource States

The client application passes the **transactionOperationStatus** in the request body so that the resource can be placed into a desired state. Services Gatekeeper either confirms this desired state in the **transactionStatus** response field, or instead shows a failure state as shown in Table 6–1 and Table 6–2.

Table 6–1 Resource States for Charges and Refunds

Value	Description
Charged	A successful charge was made.
Refunded	A successful refund was made.
Denied	The policy exception in the response will explain the reason. For example, insufficient balance, security issue, etc.
Refused	The charge or refund was refused, or not explicitly accepted.

Table 6–2 Resource States for Reservations

Value	Description
Reserved	A successful reservation was created.
Denied	The policy exception in the response will explain the reason. For example, insufficient balance, security issue, etc.
Refused	The reservation was refused, or not explicitly accepted.
Charged	A reservation has been charged against.
Released	The reservation has ended.

Payment Exceptions

For an overview of exceptions in the Services Gatekeeper RESTful interface, see "Errors and Exceptions" for more information. A list of service and policy exceptions specific to the payment interface is provided in Table 6–3.

Table 6–3 Service Exceptions for Payment

ID	Exception Text	Variables
SVC0270	Charging operation failed, the charge was not applied	None
SVC0271	Refunds not supported	Guidance from the implementation on what to do instead should be provided
SVC0273	Refund failed	The reason the refund failed. Valid reasons include: ■ The user did not accept the refund ■ The refund request is for an amount greater than the original charge

7

Wholesale Applications Community Enhancements

This chapter describes how to use Oracle Communications Services Gatekeeper as a Wholesale Applications Community (WAC) Operator Endpoint server in the Oracle Communications Service Gatekeeper's OneAPI (OneAPI) V2.1 Payment API flow.

Services providers can expose Diameter-based network charging capability using OneAPI Payment Server functionality. Policy configuration allows the enforcement of request rate and quotas assuring that charging systems are protected from denial of service (DoS) attacks.

About WAC

The WAC is an industry alliance of communications service providers, device manufacturers and software vendors formed to standardize the development and distribution of mobile applications.

For information about WAC, see:

`http://www.wacapps.net/`

Oracle is a member of the WAC.

Supported WAC Functions

Services Gatekeeper supports the following functions used in the WAC baseline architecture for the Payment API Flow:

- OneAPI V2.1 Payment Services

For information about the WAC Payment API Flow, see:

`https://members.wholesaleappcommunity.com/redmine/projects/napigw0-2/wiki/Network_APIs_Platform_Beta_02_Technical_Specification`

OneAPI V2.1 Payment Services

Services Gatekeeper supports the OneAPI RESTful Payment API V2.1 allowing service providers to accept WAC payment requests as part of the Payment API Flow. Services Gatekeeper handles WAC OneAPI payment transactions by forwarding requests to a Diameter plug-in for processing in the network tier.

The WAC Payment API flow currently supports the following transactions:

- Charge a user for a one-time payment

- Check status of previous transaction
- Display a list of all transactions made with an application

The Services Gatekeeper implementation is based on the specifications provided by the Global System for Mobile Communications (GSM).

For information on using the OneAPI V2.1 Payment interface with Services Gatekeeper, see Chapter 6, "OneAPI Payment."