Avaya Aura® Session Manager Element Manager Web Service

API Programming Reference

Release 10.2.0
Issue 7
December 2023

© 2017-2023 Avaya Inc.

All Rights Reserved.

Notice

While reasonable efforts have been made to ensure that the information in this document is complete and accurate at the time of printing, Avaya assumes no liability for any errors. Avaya reserves the right to make changes and corrections to the information in this document without the obligation to notify any person or organization of such changes.

Documentation disclaimer

"Documentation" means information published in varying mediums which may include product information, operating instructions and performance specifications that are generally made available to users of products. Documentation does not include marketing materials. Avaya shall not be responsible for any modifications, additions, or deletions to the original published version of Documentation unless such modifications, additions, or deletions were performed by or on the express behalf of Avaya. End User agrees to indemnify and hold harmless Avaya, Avaya's agents, servants and employees against all claims, lawsuits, demands and judgments arising out of, or in connection with, subsequent modifications, additions or deletions to this documentation, to the extent made by End User.

Link disclaimer

Avaya is not responsible for the contents or reliability of any linked websites referenced within this site or Documentation provided by Avaya. Avaya is not responsible for the accuracy of any information, statement or content provided on these sites and does not necessarily endorse the products, services, or information described or offered within them. Avaya does not guarantee that these links will work all the time and has no control over the availability of the linked pages.

Warranty

Avaya provides a limited warranty on Avaya hardware and software. Refer to your sales agreement to establish the terms of the limited warranty. In addition, Avaya's standard warranty language, as well as information regarding support for this product while under warranty is available to Avaya customers and other parties through the Avaya Support website: http://support.avaya.com/helpcenter/getGenericDetails?detailId=C20091120112456651010 under the link "Warranty & Product Lifecycle" or such successor site as designated by Avaya. Please note that if You acquired the product(s) from an authorized Avaya Channel Partner outside of the United States and Canada, the warranty is provided to You by said Avaya Channel Partner and not by Avaya.

"Hosted Service" means an Avaya hosted service subscription that You acquire from either Avaya or an authorized Avaya Channel Partner (as applicable) and which is described further in Hosted SAS or other service description documentation regarding the applicable hosted service. If You purchase a Hosted Service subscription, the foregoing limited warranty may not apply but You may be entitled to support services in connection with the Hosted Service as described further in your service description documents for the applicable Hosted Service. Contact Avaya or Avaya Channel Partner (as applicable) for more information.

Hosted Service

THE FOLLOWING APPLIES ONLY IF YOU PURCHASE AN AVAYA HOSTED SERVICE SUBSCRIPTION FROM AVAYA OR AN AVAYA CHANNEL PARTNER (AS APPLICABLE), THE TERMS OF USE FOR HOSTED SERVICES ARE AVAILABLE ON THE AVAYA WEBSITE, HTTPS://SUPPORT.AVAYA.COM/LICENSEINFO UNDER THE LINK "Avaya Terms of Use for Hosted Services" OR SUCH SUCCESSOR SITE AS DESIGNATED BY AVAYA, AND ARE APPLICABLE TO ANYONE WHO ACCESSES OR USES THE HOSTED SERVICE. BY ACCESSING OR USING THE HOSTED SERVICE, OR AUTHORIZING OTHERS TO DO SO, YOU, ON BEHALF OF YOURSELF AND THE ENTITY FOR WHOM YOU ARE DOING SO (HEREINAFTER REFERRED TO INTERCHANGEABLY AS "YOU" AND "END USER"), AGREE TO THE TERMS OF USE. IF YOU ARE ACCEPTING THE TERMS OF USE ON BEHALF A COMPANY OR OTHER LEGAL ENTITY, YOU REPRESENT THAT YOU HAVE THE AUTHORITY TO BIND SUCH ENTITY TO THESE TERMS OF USE. IF YOU DO NOT HAVE SUCH AUTHORITY, OR IF YOU DO NOT WISH TO ACCEPT THESE TERMS OF USE, YOU MUST NOT ACCESS OR USE THE HOSTED SERVICE OR AUTHORIZE ANYONE TO ACCESS OR USE THE HOSTED SERVICE.

Licenses

THE SOFTWARE LICENSE TERMS AVAILABLE ON THE AVAYA WEBSITE, HTTPS://SUPPORT.AVAYA.COM/LICENSEINFO

, UNDER THE LINK “AVAYA SOFTWARE LICENSE TERMS (Avaya Products)” OR SUCH SUCCESSOR SITE AS DESIGNATED BY AVAYA, ARE APPLICABLE TO ANYONE WHO DOWNLOADS, USES AND/OR INSTALLS AVAYA SOFTWARE, PURCHASED FROM AVAYA INC., ANY AVAYA AFFILIATE, OR AN AVAYA CHANNEL PARTNER (AS APPLICABLE) UNDER A COMMERCIAL AGREEMENT WITH AVAYA OR AN AVAYA CHANNEL PARTNER. UNLESS OTHERWISE AGREED TO BY AVAYA IN WRITING, AVAYA DOES NOT EXTEND THIS LICENSE IF THE SOFTWARE WAS OBTAINED FROM ANYONE OTHER THAN AVAYA, AN AVAYA AFFILIATE OR AN AVAYA CHANNEL PARTNER; AVAYA RESERVES THE RIGHT TO TAKE LEGAL ACTION AGAINST YOU AND ANYONE ELSE USING OR SELLING THE SOFTWARE WITHOUT A LICENSE. BY INSTALLING, DOWNLOADING OR USING THE SOFTWARE, OR AUTHORIZING OTHERS TO DO SO, YOU, ON BEHALF OF YOURSELF AND THE ENTITY FOR WHOM YOU ARE INSTALLING, DOWNLOADING OR USING THE SOFTWARE (HEREINAFTER REFERRED TO INTERCHANGEABLY AS “YOU” AND “END USER”), AGREE TO THESE TERMS AND CONDITIONS AND CREATE A BINDING CONTRACT BETWEEN YOU AND AVAYA INC. OR THE APPLICABLE AVAYA AFFILIATE (“AVAYA”).

Avaya grants You a license within the scope of the license types described below, with the exception of Heritage Nortel Software, for which the scope of the license is detailed below. Where the order documentation does not expressly identify a license type, the applicable license will be a Designated System License as set forth below in the Designated System(s) License (DS) section as applicable. The applicable number of licenses and units of capacity for which the license is granted will be one (1), unless a different number of licenses or units of capacity is specified in the documentation or other materials available to You. “Software” means computer programs in object code, provided by Avaya or an Avaya Channel Partner, whether as stand-alone products, pre-installed on hardware products, and any upgrades, updates, patches, bug fixes, or modified versions thereto. “Designated Processor” means a single stand-alone computing device. “Server” means a set of Designated Processors that hosts (physically or virtually) a software application to be accessed by multiple users. “Instance” means a single copy of the Software executing at a particular time: (i) on one physical machine; or (ii) on one deployed software virtual machine (“VM”) or similar deployment.

License types

Designated System(s) License (DS). End User may install and use each copy or an Instance of the Software only: 1) on a number of Designated Processors up to the number indicated in the order; or 2) up to the number of Instances of the Software as indicated in the order, Documentation, or as authorized by Avaya in writing. Avaya may require the Designated Processor(s) to be identified in the order by type, serial number, feature key, Instance, location or other specific designation, or to be provided by End User to Avaya through electronic means established by Avaya specifically for this purpose.

Named User License (NU). End User may: (i) install and use each copy or Instance of the Software on a single Designated Processor or Server per authorized Named User (defined below); or (ii) install and use each copy or Instance of the Software on a Server so long as only authorized Named Users access and use the Software as indicated in the order, Documentation, or as authorized by Avaya in writing. “Named User”, means a user or device that has been expressly authorized by Avaya to access and use the Software. At Avaya’s sole discretion, a “Named User” may be, without limitation, designated by name, corporate function (e.g., webmaster or helpdesk), an e-mail or voice mail account in the name of a person or corporate function, or a directory entry in the administrative database utilized by the Software that permits one user to interface with the Software.

Shrinkwrap License (SR). End User may install and use the Software in accordance with the terms and conditions of the applicable license agreements, such as "shrinkwrap" or "clickthrough" license accompanying or applicable to the Software ("Shrinkwrap License") as indicated in the order, Documentation, or as authorized by Avaya in writing.

Heritage Nortel Software

“Heritage Nortel Software” means the software that was acquired by Avaya as part of its purchase of the Nortel Enterprise Solutions Business in December 2009. The Heritage Nortel Software is the software contained within the list of Heritage Nortel Products located at https://support.avaya.com/LicenseInfo/ under the

link “Heritage Nortel Products,” or such successor site as designated by Avaya. For Heritage Nortel Software, Avaya grants Customer a license to use Heritage Nortel Software provided hereunder solely to the extent of the authorized activation or authorized usage level, solely for the purpose specified in the Documentation, and solely as embedded in, for execution on, or for communication with Avaya equipment. Charges for Heritage Nortel Software may be based on extent of activation or use authorized as specified in an order or invoice.

Copyright

Except where expressly stated otherwise, no use should be made of materials on this site, the Documentation, Software, Hosted Service, or hardware provided by Avaya. All content on this site, the documentation, Hosted Service, and the product provided by Avaya including the selection, arrangement and design of the content is owned either by Avaya or its licensors and is protected by copyright and other intellectual property laws including the sui generis rights relating to the protection of databases. You may not modify, copy, reproduce, republish, upload, post, transmit or distribute in any way any content, in whole or in part, including any code and software unless expressly authorized by Avaya. Unauthorized reproduction, transmission, dissemination, storage, and or use without the express written consent of Avaya can be a criminal, as well as a civil offense under the applicable law.

Virtualization

The following applies if the product is deployed on a virtual machine. Each product has its own ordering code and license types. Unless otherwise stated, each Instance of a product must be separately licensed and ordered. For example, if the end user customer or Avaya Channel Partner would like to install two Instances of the same type of products, then two products of that type must be ordered.

Third Party Components

“Third Party Components” mean certain software programs or portions thereof included in the Software or Hosted Service may contain software (including open source software) distributed under third party agreements (“Third Party Components”), which contain terms regarding the rights to use certain portions of the Software (“Third Party Terms”). As required, information regarding distributed Linux OS source code (for those products that have distributed Linux OS source code) and identifying the copyright holders of the Third Party Components and the Third Party Terms that apply is available in the products, Documentation or on Avaya’s website at: https://support.avaya.com/Copyright or such successor site as designated by Avaya. The open source software license terms provided as Third Party Terms are consistent with the license rights granted in these Software License Terms, and may contain additional rights benefiting You, such as modification and distribution of the open source software. The Third Party Terms shall take precedence over these Software License Terms, solely with respect to the applicable Third Party components to the extent that these Software License Terms impose greater restrictions on You than the applicable Third Party Terms.

The following applies only if the H.264 (AVC) codec is distributed with the product. THIS PRODUCT IS LICENSED UNDER THE AVC PATENT PORTFOLIO LICENSE FOR THE PERSONAL USE OF A CONSUMER OR OTHER USES IN WHICH IT DOES NOT RECEIVE REMUNERATION TO (i) ENCODE VIDEO IN COMPLIANCE WITH THE AVC STANDARD (“AVC VIDEO”) AND/OR (ii) DECODE AVC VIDEO THAT WAS ENCODED BY A CONSUMER ENGAGED IN A PERSONAL ACTIVITY AND/OR WAS OBTAINED FROM A VIDEO PROVIDER LICENSED TO PROVIDE AVC VIDEO. NO LICENSE IS GRANTED OR SHALL BE IMPLIED FOR ANY OTHER USE. ADDITIONAL INFORMATION MAY BE OBTAINED FROM MPEG LA, L.L.C. SEE HTTP://WWW.MPEGLA.COM

Service Provider

THE FOLLOWING APPLIES TO AVAYA CHANNEL PARTNER’S HOSTING OF AVAYA PRODUCTS OR SERVICES. THE PRODUCT OR HOSTED SERVICE MAY USE THIRD PARTY COMPONENTS SUBJECT TO THIRD PARTY TERMS AND REQUIRE A SERVICE PROVIDER TO BE INDEPENDENTLY LICENSED DIRECTLY FROM THE THIRD PARTY SUPPLIER. AN AVAYA CHANNEL PARTNER’S HOSTING OF AVAYA PRODUCTS MUST BE AUTHORIZED IN WRITING BY AVAYA AND IF THOSE HOSTED PRODUCTS USE OR EMBED CERTAIN THIRD PARTY SOFTWARE, INCLUDING BUT NOT LIMITED TO MICROSOFT SOFTWARE OR CODECS, THE AVAYA CHANNEL PARTNER IS REQUIRED TO INDEPENDENTLY OBTAIN ANY APPLICABLE LICENSE AGREEMENTS, AT THE AVAYA CHANNEL PARTNER’S EXPENSE,

DIRECTLY FROM THE APPLICABLE THIRD PARTY SUPPLIER.

WITH RESPECT TO CODECS, IF THE AVAYA CHANNEL PARTNER IS HOSTING ANY PRODUCTS THAT USE OR EMBED THE H.264 CODEC OR H.265 CODEC, THE AVAYA CHANNEL PARTNER ACKNOWLEDGES AND AGREES THE AVAYA CHANNEL PARTNER IS RESPONSIBLE FOR ANY AND ALL RELATED FEES AND/OR ROYALTIES. THE H.264 (AVC) CODEC IS LICENSED UNDER THE AVC PATENT PORTFOLIO LICENSE FOR THE PERSONAL USE OF A CONSUMER OR OTHER USES IN WHICH IT DOES NOT RECEIVE REMUNERATION TO: (I) ENCODE VIDEO IN COMPLIANCE WITH THE AVC STANDARD ("AVC VIDEO") AND/OR (II) DECODE AVC VIDEO THAT WAS ENCODED BY A CONSUMER ENGAGED IN A PERSONAL ACTIVITY AND/OR WAS OBTAINED FROM A VIDEO PROVIDER LICENSED TO PROVIDE AVC VIDEO. NO LICENSE IS GRANTED OR SHALL BE IMPLIED FOR ANY OTHER USE. ADDITIONAL INFORMATION FOR H.264 (AVC) AND H.265 (HEVC) CODECS MAY BE OBTAINED FROM MPEG LA, L.L.C. SEE HTTP://WWW.MPEGLA.COM.

Compliance with Laws

You acknowledge and agree that it is Your responsibility for complying with any applicable laws and regulations, including, but not limited to laws and regulations related to call recording, data privacy, intellectual property, trade secret, fraud, and music performance rights, in the country or territory where the Avaya product is used.

Preventing Toll Fraud

"Toll Fraud" is the unauthorized use of your telecommunications system by an unauthorized party (for example, a person who is not a corporate employee, agent, subcontractor, or is not working on your company's behalf). Be aware that there can be a risk of Toll Fraud associated with your system and that, if Toll Fraud occurs, it can result in substantial additional charges for your telecommunications services.

Avaya Toll Fraud intervention

If You suspect that You are being victimized by Toll Fraud and You need technical assistance or support, call Technical Service Center Toll Fraud Intervention Hotline at +1-800-643-2353 for the United States and Canada. For additional support telephone numbers, see the Avaya Support website: https://support.avaya.com or such successor site as designated by Avaya.

Security Vulnerabilities

Information about Avaya's security support policies can be found in the Security Policies and Support section of https://support.avaya.com/security.

Suspected Avaya product security vulnerabilities are handled per the Avaya Product Security Support Flow (https://support.avaya.com/css/P8/documents/100161515).

Downloading Documentation

For the most current versions of Documentation, see the Avaya Support website: https://support.avaya.com, or such successor site as designated by Avaya.

Contact Avaya Support

See the Avaya Support website: https://support.avaya.com for product or Hosted Service notices and articles, or to report a problem with your Avaya product or Hosted Service. For a list of support telephone numbers and contact addresses, go to the Avaya Support website: http://support.avaya.com (or such successor site as designated by Avaya), scroll to the bottom of the page, and select Contact Avaya Support.

Trademarks

The trademarks, logos and service marks ("Marks") displayed in this site, the Documentation, Hosted Service(s), and product(s) provided by Avaya are the registered or unregistered Marks of Avaya, its affiliates, its licensors, its suppliers, or other third parties. Users are not permitted to use such Marks without prior written consent from Avaya or such third party which may own the Mark. Nothing contained in this site, the Documentation, Hosted Service(s) and product(s) should be construed as granting, by implication, estoppel, or otherwise, any license or right in and to the Marks without the express written permission of Avaya or the applicable third party.

Avaya is a registered trademark of Avaya Inc.

All non-Avaya trademarks are the property of their respective owners. Linux® is the registered trademark of Linus Torvalds in the U.S. and other countries.

Contents

Chapter 1: Getting Started 10
Introduction 10
Intended audience 10
Change History 10
RESTful Based Web Service 11
RESTful Best Practice Alignment 11
RESTful Best Practice Divergence 11
Web Service Development Assumptions 11
RESTful Client Development Assumptions 12
First Steps 13
User access to the Session Manager Element Manager Web Service 13
System Manager Session Management 13
Gaining REST API Familiarity using Browser Tools 14
Next Steps 14
SSL/TLS Trust Setup 14
Contents of SDK zip 15

Chapter 2: Authentication and authorization 16
HTTP supported methods 16
OPTIONS support 16
HEAD support 17
Web Service URIs 17
URL encoding 17
Query format 17
HTTP Response Status Codes 18
Error response body content format 19
HTTP Header Properties 20
Required Request Headers 20
GZIP content compression 21
Character set negotiation 21
Response body content tips 22
XML resource result wrapping and unwrapped sub-resource arrays 22
Date Representation 22
Null value handling 22
URI Link representation 22

Chapter 3: Session Manager Status 24
Get a single SM status by ID or name 24

Query SM Status 24
Change SM Deny New Service state 25
Change SM Maintenance Mode State 25
SM Status response content 26
SM Status query response content normal format 28
SM Status query response brief format 32

Chapter 4: Registration Status and AST Device Notifications 33
SIP User Registrations 33
Registration data 33
Registration keys 33
Registration cache and data availability 34
Query expressions and results 35
Query control values 36
Query filter parameters 36
Query sort parameter 37
Get a single SIP registration status by registration key or self- reference 37
Query multiple SIP registrations 37
Registration Response 38
Registration Query Response 42
AST Device Notification 43
Request AST notification based on Query 44
Request AST notification based on registration key 44
Request AST notifications based on Session Manager 45
Notification Request Errors 46
Notification Response 47

Chapter 5: Groups – Data Centers and Regions 48

..... 48
Get a single group by ID or name 48
Query groups 49
Add a group 49
Modify specified group fields 50
Replace an existing group by ID or name 51
Delete a group by ID or name 51
Add an SM instance to an existing group 52
Remove an SM instance from an existing group 52

Chapter 6: Location to Region 54
Get a single location to region by ID or location name 54
Query location to regions 55
Add regions to a location 56
Add regions to a location 57

Chapter 7: Session Manager Instances 59
Get all SM/BSM instances 59
Get a single SM/BSM instance by ID 59
Create SM/BSM Instance 60
Edit SM/BSM instance 60
Delete SM instance 60
SM instance response content sample 62
SM instance request body sample 63

Chapter 8: SDK Java JAX-RS Examples 64
Working with the Java samples 64
JAX-RS Client Setup 64
Session Manager Status 65
Session Manger Operation Request 66
Registration Status 67
Registration Notification Request 68

Chapter 9: Related resources 71
Session Manager Documentation 71
Finding documents on the Avaya Support website 72
Avaya Documentation Portal navigation 72
Training 74
Viewing Avaya Mentor videos 74
Support 75
Using the Avaya InSite Knowledge Base 75

Chapter 9: Glossary 76

Chapter 1: Getting Started

Introduction

The Session Manager EM Web Services API Programming interface is one of the Web Services comprising of the System Manager Web Service. The Web Service provides programmatic access to Session Manager Dashboard and User Registration status data. It provides access to read the Dashboard and User Registration data and execute operations consistent with the Dashboard and User Registration GUIs. The data available through the web services maps to the data in the GUI interface table.

Intended audience

The audience for this document includes:

- Avaya R&D (Developers and SV)
- Avaya Professional Services
- Business Partners
- Avaya Sales

Change History

Issue	Date	Summary of changes
6	January 2023	No API Changes in 10.1.2
5	December 2021	No Changes in 10.1
4	June 2021	For Release 8.1.12 - Added chapter for Session Manager Groups webservices API - Added chapter for Location to region mapping webservices API
3	March 2021	For Release 8.1.11 - Addition of <Loadfactor> and <profile> elements for SM status response - Addition of policy, third and fourth registration status response
2	October 2020	For Release 8.1.3: - Addition of <deviceSerial> in registration response for J1XX devices - Addition of <vistingSurvReg> and <visitingSurvName> in registration

		response for Branch Visiting User - Addition of <pushNotificationCounts> in SM status response for Avaya Push Notification
1	July 2017	Release 7.1 document.

RESTful Based Web Service

The Web Service is based entirely on Representational State Transfer (RESTful) practices. This is a programmatic API. This API is focused on a URI resource driven approach.

RESTful Best Practice Alignment

- Make the API semantically straightforward and reduce complexity wherever possible.
- Make stateless API
- Support for HTTP GET/POST/PUT/DELETE method semantics to create, modify, and deletequery operations
- Align HTTP error code set with HTTP return code semantics
- Use Content-Type or embedding version in URL. Content-Type semantics is chosen to not affect client URL encoding when a new version of the API is released. In general, attempt for backward compatibility between versions.
- Always represent URIs as a resource.
- Use nouns to indicate URIs and plural nouns to indicate a collection of resources. Also, use a key to represent a unique element of a collection. For example, <entity>s/[id|.
- Use lower case to represent resource names in URI and content.
- Use camelCase to represent Atomic properties .
- Allow URI referencing between base level resources
- Incorporate self-reference and return of resource URI references in content where appropriate for singleton and multiple results
- Allow customer control and viewing of metadata
- Support XML and JSON content
- Support URL64 and simple URL encoding, where URL encoding is required.

RESTful Best Practice Divergence

- Most RESTful interfaces support client-side caching, however, this API has minimal support. There will not be any performance gain if you choose to use client-side caching.
- Support of language, encoding, and localization is minimal. Only UTF-8 is supported.

Web Service Development Assumptions

Use of this interface assumes the following:

- Web Application development experience
- Familiarity with Web Services or service based APIs
- Knowledge of how to establish SSL/TLS connections and trust certificate stores
- HTTP protocol and header knowledge
- Experience with the Dashboard and User Registration GUIs semantics
- Knowledge of Session Manager and related User Management

The implementation is client technology agnostic and focused on ease of development and understanding. Examples and samples focus on simplistic script or browser-based interactions.

RESTful Client Development Assumptions

There will be no formal REST service description since there really is not a standard for RESTful API description. URIs are represented informally as what is known as ”URI Template" syntax. XML namespace use in client is not supported in XML request content for the following reasons :

- To keep maximum flexibility for backwards compatibility
- To prevent inadvertent compatibility issues

This will be compensated a bit by providing a comprehensive error reporting scheme.

Numerous JEE standards, RFCs, ISO standards will be employed as per best REST semantics for HTTP header use. But as REST itself is not standards driven, more attention in this design is paid to usage and interaction semantics.

This is a stateless API and only supports a synchronous request/reply model as per common HTTP web application practices. Long running requests, polling or any sort of asynchronous handling are not supported. Also, jax-rs 2.0 async client handling, which can accomplish similar sorts of behavior, are not prevented. Only a single "first class" data entity can be added, updated, or deleted within a single POST or PUT or DELETE request. There are no plans to support HTTP request pipelining (support of multiple HTTP requests within a single POST) in the first release.

Clients should be designed for rapid, short response time interactions and take into consideration System Manager resource loading during client use. Client application resilience is a client side, not server side, responsibility.

Clients must be aware that this is a versioned interface, to allow for future interoperability. The intent in "v1" is to get all the fundamental operations up and going and enough to do the basics.

Clients are expected to do the following:

- Create an SSL/TLS session
- Handle HTTP basic authentication, and optionally session cookies to handle "Challenge Response" form of basic authentication

- Handle unexpected HTTP errors and TLS link errors gracefully
- Modify and parse HTTP headers
- Handle URI encoding and to a lesser extent decoding
- Handle URI matrix parameters when needed, these differ from query parameters and are not commonly used
- Map internal data models with JSON or XML parsers and formatters.

First Steps

- Install test System Manager.
- Configure at least one Session Manager.
- Make sure that there some Session Managers and SIP Users administered with Session Manager Communication Profiles on the target System Manager.

User access to the Session Manager Element Manager Web Service

HTTP basic authentication is required to access the web service. A System Manager administered user **Login Name** and **Password** needs to be added to each request. If an invalid user and password combination is sent in the HTTP authorization request, then a "401 Unauthorized" error response is received.

By default, the following users have access to the Routing Web service:

- System Administrator
- Avaya Services Administrator
- Session Manager and Routing Administrator

If a user is other than **Admin**, then the user must be assigned a custom role with **Web Services -> Routing** permissions enabled to access the web service. A user without RBAC permission will receive a **403 Forbidden** HTTP error response.

System Manager Session Management

Both the System Manager and GUI based web services use the same provider for HTTP session management across multiple requests. The difference is that we forcefully close each session after each reply. Therefore, re-submitting session cookies is not advised. Re-submitting a cookie for a stale session is considered an error on the System Manager but a new session is allocated without an authentication error.

There are circumstances where internal server error responses, for example, 500 - internal server error, incorrectly close the session. However, restriction on the number of sessions accessed by an administrator prevents subsequent authorization or GUI login attempts. Administrators are limited to 5 sessions by default, which can be increased to 25 sessions per administrator. These sessions comprise of both GUI and web service sessions.

Gaining REST API Familiarity using Browser Tools

Procedure

1. Set up a test System Manager to work with.
2. Use your favorite browser to enter `https://<fqdn>/ASM/ws/asmstatuses`
3. Accept any warning about an insecure site if you have not yet established trust to the System Manager host.
4. If prompted, enter the administrator login and password.
   You should see the content format of the REST API assuming there are some Session Managers administered.
5. After you have an authenticated session established, you can try any of the GET method examples as given in Registration Status and AST Device Notifications
6. Work with the Session Manager Element Manager Web Service by using a browser tool to gain more familiarity with the resource mappings and content representations before starting client development. Some worth considering client development tools are as follows:
   a. Firefox "RestClient" and
   b. Chrome "Postman" or "Advanced Rest Client".
   These clients remember history, take care of authentication and cookie handling, provide hyperlink exploration of URIs, print responses, handle JSON or XML, and have many other conveniences.

Next Steps

SSL/TLS Trust Setup

System Manager uses self-signed certificates by default. Since the SM EM Web Service is client agnostic, providing specific detail on SSL/TLS connection setup is not possible due to the variety of mechanisms employed which depend on the REST client package, language, operating system, and so on.

Initial setup for any client development requires trusting all CAs or importing the self-signed CA from System Manager into a trust store.

Do one of the following to setup trust:

- Use settings such as trust any certificate, during development
  - Many client packages support this setting
  - Java JSSE implementation requires some simple overwriting of socket factories through SSLContext
- Use a browser to connect to the System Manager and navigate to the site security information within the browser.
  For non-Windows clients
  - Export and save X.509 Certificate in PEM format

- Use tools such as Java keytool, OpenSSL, opensssl, to store and import the certificate into the trust store.

For Windows clients when in IE

- Click **Certificate Error** beside the address bar
- Click **View Certificates**
- Click **Install Certificate** and select the appropriate store for client access

- Access **System Manager -> Security -> Certificates -> Authority -> CA Structure & CRLs.** Then download the pem file to import the certificate into your trust certificate store.

Java JKS trust certificate store

To handle System Manager self-signed certificates, it is reasonable to create and export the CA certificate and import into a trust store. For example,

```
$ keytool - import  -alias myhostname -file myhostname.crt -keystore sample.jks
Enter keystore password:  sample
Re-enter  new  password: sample
```

Answer `yes` when asked whether this is a trusted certificate.

Contents of SDK zip

The Web Service is designed to be used without any SDK. This zip is used for development of java clients. It is focused on providing JAXB mappings between Session Manager Web Service model classes and JSON or XML content. It uses of one of the Java REST client frameworks such as Oracle Jersey, JBoss RESTEasy, Apache CXF or HTTPClient.

The contents of the zip are:

- asmelementmgr_jaxb.jar containing JAXB mappings for Java clients
- javadoc for the JAXB model classes
- Sample code which works with latest Oracle Jersey client. The code is dependent upon Jersey and JUnit.

Chapter 2: Authentication and authorization

The web service supports HTTP Basic authentication. The client encodes the "username:password" string and puts it into an Authorization HTTP header.

```
Authorization: Basic QWxhZGluZzpvcGVuIHNlc2FtZQh1.
GET https://{fqdn}/ASM/ws/...
```

If an invalid user and password combination is sent in the HTTP authorization request, then the following error response is received.

```
WWW-Authenticate: Basic realm="admin"
Status Code: 401 Not Authorized
```

It is up to the client developer to choose whether "Direct Authentication" or "Challenge Authentication" is to be supported. For "Direct Authentication", the authorization header is sent with every request.

If the client supports "Challenge Authentication", which is also referred to as "preemptive" authentication, then when the client receives a "401" error code the request must be resubmitted with the authorization header. There is no session management within System Manager for security reasons. A new session is created per request and then canceled at the end of the request. Therefore, authorization header must be sent with each request.

HTTP supported methods

Supported HTTP method semantics align with CRUD model of data access and JAX-RS 1.1/2.0 annotations. The methods are:

- GET: reads query or fetches single item for unique resource URI reference, query across resource collection for URI collection resource. No request body content is supported.
- POST: executes an operation. This might not require request body content
- PUT: not currently used
- DELETE: not currently used

OPTIONS support

An HTTP OPTIONS request can be made against any valid URI to determine what the valid operations are against that resource. The valid methods are returned in the HTTP "Allow" header.

For example,

```
OPTIONS https://spyglass2.dr.avaya.com/ASM/ws/asmstatuses/851975
```

Will return the following header response

```
Server: Apache-Coyote/1.1
Allow: POST, DELETE, OPTIONS, PUT
Content-Length: 0
```

HEAD support

HTTP HEAD requests are not recommended. Although the underlying jax-rs provider supports HEAD requests, it will not reliably test response header properties.

Web Service URIs

The base URI for access is `https://{fqdn}/ASM/ws` for all web service calls to the SM Element Manager

Sub URI	Resource
http:{fqdn}/ASM/ws/asmstatus	Dashboard GUI status information and operations
http:{fqdn}/ASM/ws/registrations	Registration GUI status information and AST device operations such as reboot, config, and so on.

URL encoding

`http://en.wikipedia.org/wiki/Percent-encoding` is a good reference of the significant characters in a URI. If any of these characters appear within a query or matrix parameter value, then they must be URI encoded. For the most part, the left side "name" values are URL safe and do not need encoding.

For example, if "()" or spaces appear in a Session Manager name, then "()" and space characters must be encoded.

```
GET https://{fqdn}/ASM/ws/asmstatuses/name/A%20Session%20Manager%281%29
```

Query format

The standard URI query format is per RFC 3986, section 3.4. In short a query starts with a "?" followed by the first query term, "&" is used for subsequent terms. For example

```
GET https://{fqdn}/ws/asmstatuses/?format=brief&sort=name
```

HTTP Response Status Codes

Success codes	Reason
200 OK	The request has succeeded. The information returned with the response depends on the request method used. GET: contains an entity corresponding to the requested resource. POST or PUT: contains an entity containing the result of the action.
201 Created	The request has succeeded, and a new entity is being created.

Error Codes	Reason
400 Bad Request	The request could not be understood by the server due to malformed syntax. For example, specifying "application/xml" media type in the Content-Type header of an HTTP request and including a valid JSON content as request body.
401 Unauthorized	The request requires user authentication.
403 Forbidden	Operation is forbidden which can occur for several reasons such as no metadata, or the value violates the constraints of the metadata. The most common will be that the user does not have "Web Services: Routing" permission.
404 Not Found	The referenced resource does not exist.
405 Method Not Allowed	The HTTP request method is not supported for the resource. The response will contain an Allow header which contains a list of valid methods.
406 Not Acceptable	The requested content type is not supported.
409 Conflict	Conflict with the associated metadata. Either the data does not match or there is no metadata.
415 Unsupported Media Type	The request contains an unknown content type or unspecified content type. Check Accept or Content-Type headers
500 Internal server error	Request could not be fulfilled due to internal resource issue or unexpected error

Error response body content format

Whenever a 400 or greater HTTP status code is returned, there can be an error in the body content even though a valid URI is entered. Certain types of errors such as URI, URI formatting or Internal Server errors will not always have error content. Therefore, the client logic must be developed to return an error when there is an empty body or the following content.

The media type will be based on the request header Accept property type as used for other content.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<error>
    <status>404</status>
    <statusMessage>404 Not Found</statusMessage>
    <message>The specified id, 12345, does not exist</message>
    <additionalStatus>0</additionalStatus>
        <additionalMessage>None really for not found case, used for 500 messages
though</additionalMessage>
    <requestMethod>GET</requestMethod>
    <requestURI>/domains/12345</requestURI>
    <date>Fri Nov 22 18:09:50 MST 2013</date>
</error>



{
status: 404
statusMessage: "404 Not Found"
message: "The specified id, 12345, does not exist"
additionalStatus: 0
additionalMessage: "None really for not found case, used for 500 messages though"
requestMethod: "GET"
requestURI: "/domains/12345"
date: "Fri Nov 22 18:10:24 MST 2013"
}
```

The *message*, *additionalStatus*, and *additionalMessage* are optionally populated based on HTTP response error code. "message" will most often be populated with a user friendly message. Unexpected internal errors may populate *additionalMessage* with stack traces, so there can be a fair amount of text.

HTTP Header Properties

Required Request Headers

Content-Type, Accept and Authorization headers must be sent for all requests.

Content-Type	application/xml or application/json
Accept	application/xml or application/json
Authorization	Basic username:password The password:portion is base64 encoded: Basic YWRtaW46QWRtaW4xMjMk

Version control

An Accept property is required for all requests and must indicate a JSON or XML media. The SM EM Web Service is versioned. The first version will be known as "v1". Versioning is controlled by using the media type. This first version is formally expressed as:

```
Accept: application/vnd.avaya.asm.status-v1+xml
```

Or,

```
Accept: application/vnd.avaya.asm.status-v1+json
```

If no version information is provided, then the following latest version is assumed.

```
Accept: application/xml
```

Or,

```
Accept: application/json
```

Content control

Accept and Content-Type properties are required to be sent for all requests even those that do not take or return content such as DELETE.

This is twofold:

- Error content can always arrive in a response for any request made.
- Due to versioning implementation keying off media type, all requests need a media type even if the default is in use.

JSON and XML content are supported. HTTP Content-Type header will control request content serializing. HTTP Accept request header will control response or error content format.

```
Accept: application/xml
Content-Type: application/xml
```

```
Accept: application/json
Content-Type: application/json
```

GZIP content compression

HTTP compression is supported. The server will decode gzip compressed content with the following header in the request.

```
Content-Encoding: gzip
```

The server will encode gzip compressed response with the following header request

```
Accept-Encoding: gzip
```

If the client needs an accept coding of the following

```
Content-Encoding: gzip, deflate
```

Then for proper handling, the "deflate" directive will be passed through when received in the "Accept-Encoding" header property value.

Character set negotiation

Accept-Language and Accept-Encoding are basically ignored for character set negotiation. UTF-8 is used for content and response. No data localization is supported for SM EM web service content. Therefore, support of client localization control is not needed.

Response body content tips

XML resource result wrapping and unwrapped sub-resource arrays

XML Query results are wrapped with a plural "collection" element.

```
<locations count= "5"  limit= "100"  offset= "0"  query= ""  totalcount= "5" >
  <location>
  ...
</locations>
```

Date Representation

Dates and timer durations are represented in the same format as on the GUI.

Null value handling

In response body, content fields with null are not transferred. Specifically, if the XML element or JSON field does not appear in the content, then that value is null on the server. However, inbound matrix parameter and query are a special case. If a parameter needs to be set to null, then omit the "=" sign. This will indicate setting the value to null or using null for matching purposes.

URI Link representation

Links are represented as <link> element in the content and RsLink, RsEntityLink, and RsLinkType classes in JAXB. A link has the following attributes:

- Href: fully qualified URI link
- hrefName : unique name of referred component if ref="reference" and the URI refers to resource.
- rel - type of link. These are not IANA types as per hypermedia linking.
  - "self", all entities in response will contain a self-link
  - "reference" may refer to a resource or a sub-resource. If a resource and that resource has a name element, then a hrefName will appear in the link.
  - "referred".

hrefName can also be considered as an alternate "href" attribute. It can also be used similar to a href in a way that it can be URI encoded and used in a reference URI.

```
<registrations count="1" limit="1000" offset="0"
       query="" totalcount="1">
       <registration>
              <ast>false</ast>
              <firstName>Mouse</firstName>
              <id>50</id>
              <lastName>Mickey</lastName>
              <location>Westminister</location>
              <login>mmouse@avaya.com</login>
              <primController></primController>
              <primReg>false</primReg>
              <secController></secController>
              <secReg>false</secReg>
              <survController></survController>
              <survReg>false</survReg>
              <link href="https://augusta1.dr.avaya.com/ASM/ws/registrations/50_"
                     rel="self" />
       </registration>
</registrations>
 GET https://augusta1.dr.avaya.com/ASM/ws/registrations/50___
```

Responses will include "self" referencing <link> elements for all resource and sub resource entities. rel="self" is only supported on response content.

RsEntityLink is a special JAXB wrapper around RsLink used where one resource refers to another base level resource. Below cause for a notification is refering to the SM status resource associated with the notification counts.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<notifystatuses sent="1" requested="1" notify="reboot">
    <notifystatus sent="1" requested="1">
        <asmstatus>
            <link rel="reference" hrefname="smone"
href="https://localhost/ASM/ws/asmstatuses/1" />
        </asmstatus>
    </notifystatus>
</notifystatuses>
```

Chapter 3: Session Manager Status

All the Session Manager status data appears on the Session Manager dashboard GUI. All the data which appears on the GUI and is retrieved from the web service is from a System Manager cache. The cache is updated every two minutes for "Core" SMs and every five minutes for every Branch SM.

Get a single SM status by ID or name

Request	`GET https://{fqdn}/ASM/ws/asmstatuses/{id}`
	`GET https://{fqdn}/ASM/ws/asmstatuses/name/{name}`
	id is the numeric surrogate id, which is the last value of the self-reference URI or "id" element. Use of {name} must be preferred.

Request Content	None.
Response Content	See SM Status response. Note that updatedatetime varies with < smstatuses> element.
Response Errors	404: Not Found

Query SM Status

Request	`GET https://{fqdn}/ASM/ws/asmstatuses`

Optional Query Parameters	offset: Starts record number to return and allows chunking of large data sets.
	lmit: Returns number of query records. Maximum is 100 if not in brief format. For example,
	`.../ASM/ws/asmstatuses?limit=50`
	brief: Returns URI references only, as in "brief listing". For example,
	`.../ASM/ws/asmstatuses?brief=true &name=<value>`
	Where, <value> is matched exactly against Session Manager name.

Optional Query Filtering and Sort Parameters	&smType=SM or &smType=BSM: Filters on Session Manager type. Only a single sort parameter is supported, the default is the SM Instance id. For example, &sort=name or &sort=smType
Request Content	None.

Response Content	See SM Status response. Total number of records will be indicated in the "count" attribute within the <smstatuses> element.
Response Errors	400: Bad Request. It can occur for invalid query or sort parameter. 404: Not Found

Change SM Deny New Service state

The two request types to handle new SIP service request are as follows:

- True: SM denies any new SIP service requests
- False: SM accepts and handles New Service SIP requests. This is only effective if the SM can be connected to and is operational and is not in Maintenance Mode.

It is the responsibility of the client to check the "connected" state and validate the desired state change occurring with subsequent calls. The request is made to the SM asynchronously and might take a few seconds to take effect.

Request	`POST https://{fqdn}/ASM/ws/asmstatuses/32567/denyNewService/true` `POST https://{fqdn}/ASM/ws/asmstatuses/32567/denyNewService/false`
Request Content	None.
Response Content	See SM Status response.
Response Errors	404: Not Found

Change SM Maintenance Mode State

Session Manager can be put into Maintenance Mode prior to being provisioned or if the SM is not connected to the System Manager. The next successful connection made to the Session Manager will then force the SM into maintenance mode. Maintenance mode will change SIP processing state to Deny New Service true only if the SM <connection> value is true. Therefore, the best policy is to make the request and then make subsequent requests to ensure that that the valid state has been entered.

Request	`POST https://{fqdn}/ASM/ws/asmstatuses/32567/maintenanceMode/true` `POST`

	https://{fqdn}/ASM/ws/asmstatuses/32567/maintenanceMode/false
Request Content	None.
Response Content	See SM Status response.
Response Errors	404: Not Found

SM Status response content

The response contains a link with a URI to the underlying Session Manager administration for this status. This is an XML or JSON representation of the information from the Session Manager -> Dashboard screen. If an element does not appear, it means that it is essentially a null value or unset. This would map to the GUI interface and values displayed in table columns as "—". The entry for "blackdiamond1" shows a complete data set. The entry for "bogusbsm1" shows an entry where status was not able to be retrieved from the SM. You can key off the <connected> element value to determine the current connection state from the System Manager to the Session Manager.

```
XML
  <asmstatus>
        <id>1</id>
        <name>blackdiamond1</name>
        <activeCallCount>0</activeCallCount>
        <connected>true</connected>
        <denyNewService>false</denyNewService>
        <easgState>Disabled</easgState>
        <maintenanceMode>false</maintenanceMode>
        <managedByThisSMGR>true</managedByThisSMGR>
        <pushNotificationCounts>
             <currentActivationCount>0</currentActivationCount>
             <dayNotificationCount>0</dayNotificationCount>
             <dayNotificationErrCount>0</dayNotificationErrCount>
             <hourNotificationCount>0</hourNotificationCount>
        </pushNotificationCounts>
        <registeredCurrent>0</registeredCurrent>
        <registeredMax>0</registeredMax>
        <release>8.1.3.0.213107</release>
        <licenseStatus>
            <graceExpireDateTime>2015-01-01T17:13:50.061-07:00</graceExpireDateTime>
            <licenseMode>ERROR</licenseMode>
            <reasonCode>EXCEEDANCE</reasonCode>
```

```
        </alarmCounts>
        <loadfactor>
             <current>
                 <loadfactor>100</loadfactor>
                 <users>2000</users
                 <devices>2000</devices>
                 <sessions>2000</sessions>
             </current>
             <peakHour>
                 <loadfactor>100</loadfactor>
                 <users>2000</users
                 <devices>2000</devices>
                 <sessions>2000</sessions>
                 </peakHour>
            <peakDay>
                 <loadfactor>100</loadfactor>
                 <users>2000</users
                 <devices>2000</devices>
                 <sessions>2000</sessions>
            </peakDay>
        <loadfactor>
        <profile>
            <profile>2</profile>
            <users>2000</users
            <devices>2000</devices>
            <sessions>2000</sessions>
        </profile>
        <entityMonitoring>

             <numDeny>0</numDeny>

             <numDown>0</numDown>

             <numInitializing>0</numInitializing>

             <numNotMonitored>0</numNotMonitored>

             <numPartiallyUp>0</numPartiallyUp>

             <numUp>0</numUp>

        </entityMonitoring>

        <securityModule>Up</securityModule>

        <smType>SM</smType>

        <testsResult>true</testsResult>

        <updatedatetime>2014-08-27T13:43:15.508-06:00</updatedatetime>

        <ustoreStatus>true</ustoreStatus>

        <asmInstance>

            <link href="https://augusta2.dr.avaya.com/ASM/ws/asminstances/1"
hrefName="blackdiamond1" rel="reference"/>

        </asmInstance>

        <link href="https://augusta2.dr.avaya.com/ASM/ws/asmstatuses/1" rel="self"/>

    </asmstatus>
```

SM Status query response content normal format

XML

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<asmstatuses
    count="3"
    limit="0"
    offset="0"
    query=""
    totalcount="3">
    <asmstatus>
        <id>3</id>
        <name>testscrush</name>
        <activeCallCount>0</activeCallCount>
        <connected>true</connected>
        <denyNewService>false</denyNewService>
        <licenseStatus>
            <graceExpireDateTime>2015-05-01T11:27:35.461-06:00</graceExpireDateTime>
            <licenseMode>ERROR</licenseMode>
            <reasonCode>NOLICENSE</reasonCode>
        </licenseStatus>
          <maintenanceMode>false</maintenanceMode>
          <managedByThisSMGR>true</managedByThisSMGR>
          <pushNotificationCounts>
             <currentActivationCount>0</currentActivationCount>
             <dayNotificationCount>0</dayNotificationCount>
             <dayNotificationErrCount>0</dayNotificationErrCount>
             <hourNotificationCount>0</hourNotificationCount>
          </pushNotificationCounts>
          <registeredCurrent>0</registeredCurrent>
          <registeredMax>0</registeredMax>
          <release>8.1.3.0.213107</release>
          <replication>false</replication>
          <alarmCounts>
              <majorCount>0</majorCount>
              <minorCount>0</minorCount>
              <warningCount>0</warningCount>
          </alarmCounts>
```

```
        <entityMonitoring>
            <numDeny>0</numDeny>
            <numDown>0</numDown>
            <numInitializing>2</numInitializing>
            <numNotMonitored>0</numNotMonitored>
            <numPartiallyUp>0</numPartiallyUp>
            <numUp>2</numUp>
        </entityMonitoring>
        <securityModule>Down</securityModule>
        <smType>SM</smType>
        <testsResult>false</testsResult>
        <updatedatetime>2015-04-21T12:31:04.348-06:00</updatedatetime>
        <ustoreStatus>false</ustoreStatus>
        <asmInstance>
            <link
                href="https://augusta2.dr.avaya.com/ASM/ws/asminstances/3"
                hrefName="testscrush"
                rel="reference" />
        </asmInstance>
        <link
            href="https://augusta2.dr.avaya.com/ASM/ws/asmstatuses/3"
            rel="self" />
    </asmstatus>
    <asmstatus>
             <id>4</id>
        <name>magnolia1</name>
        <activeCallCount>0</activeCallCount>
        <connected>true</connected>
        <denyNewService>false</denyNewService>
        <licenseStatus>
            <graceExpireDateTime>2015-05-01T11:27:35.462-06:00</graceExpireDateTime>
            <licenseMode>ERROR</licenseMode>
            <reasonCode>NOLICENSE</reasonCode>
        </licenseStatus>
        <maintenanceMode>false</maintenanceMode>
        <managedByThisSMGR>true</managedByThisSMGR>
        <registeredCurrent>0</registeredCurrent>
        <registeredMax>0</registeredMax>
        <release>6.3.12.0.631205</release>
        <replication>true</replication>
```

```
        <alarmCounts>
            <majorCount>0</majorCount>
            <minorCount>0</minorCount>
            <warningCount>0</warningCount>
        </alarmCounts>
        <entityMonitoring>
            <numDeny>0</numDeny>
            <numDown>0</numDown>
            <numInitializing>0</numInitializing>
            <numNotMonitored>0</numNotMonitored>
            <numPartiallyUp>0</numPartiallyUp>
            <numUp>0</numUp>
        </entityMonitoring>
        <securityModule>Up</securityModule>
        <smType>SM</smType>
        <testsResult>true</testsResult>
        <updatedatetime>2015-04-21T12:31:04.364-06:00</updatedatetime>
        <ustoreStatus>true</ustoreStatus>
        <asmInstance>
            <link
                href="https://augusta2.dr.avaya.com/ASM/ws/asminstances/4"
                hrefName="magnolia1"
                rel="reference" />
        </asmInstance>
        <link
            href="https://augusta2.dr.avaya.com/ASM/ws/asmstatuses/4"
            rel="self" />
    </asmstatus>
    <asmstatus>
             <id>8</id>
        <name>bogusbsm1</name>
        <connected>false</connected>
        <licenseStatus>
            <graceExpireDateTime>2015-05-06T14:35:52.033-06:00</graceExpireDateTime>
            <licenseMode>ERROR</licenseMode>
            <reasonCode>NOLICENSE</reasonCode>
        </licenseStatus>
        <maintenanceMode>true</maintenanceMode>
        <managedByThisSMGR>true</managedByThisSMGR>
```

```
          <alarmCounts>
              <majorCount>0</majorCount>
              <minorCount>0</minorCount>
              <warningCount>0</warningCount>
          </alarmCounts>
          <smType>BSM</smType>
          <updatedatetime>2015-04-21T12:25:12.159-06:00</updatedatetime>
          <asmInstance>
              <link
                  href="https://augusta2.dr.avaya.com/ASM/ws/asminst
                  ances/8" hrefName="bogusbsm1"
                  rel="reference" />
          </asmInstance>
          <link
              href="https://augusta2.dr.avaya.com/ASM/ws/asmsta
              tuses/8" rel="self" />
      </asmstatus>
  </asmstatuses>
```

SM Status query response brief format

XML

```
 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<asmstatuses count="4" limit="1000" offset="0" query="?format=brief" totalcount="4">
    <asmstatus>
        <link href="https://augusta1.dr.avaya.com/ASM/ws/asmstatuses/1" rel="reference"/>
    </asmstatus>
    <asmstatus>
        <link href="https://augusta1.dr.avaya.com/ASM/ws/asmstatuses/2" rel="reference"/>
    </asmstatus>
    <asmstatus>
        <link href="https://augusta1.dr.avaya.com/ASM/ws/asmstatuses/3" rel="reference"/>
    </asmstatus>
    <asmstatus>
        <link href="https://augusta1.dr.avaya.com/ASM/ws/asmstatuses/4" rel="reference"/>
    </asmstatus>
</asmstatuses>
```

Chapter 4: Registration Status and AST Device Notifications

SIP User Registrations

Registration data

This registration data is retrieved from the same System Manager cache that supports the User Registration GUI. Therefore, you must view the data and the associated documentation to understand the semantics of the data.

A record is presented for all users even when there is no active registration for a user. This is called a "place holder" record. It is primarily filled with a SM Communication Profile administration of a user. When a device registration occurs for a user, the place holder record is then replaced with a more complete data set representing an active SIP registration. As a user can have multiple SIP device registrations, a new registration record is added for each device.

Each device can register with multiple SMs for a single SIP registration. To determine if the record represents an active registration, the primReg, secReg, survReg field values must be checked and one of them should be "true". Alternatively, the <controller> element, which indicates the Session Manager that is the active controller, must be non-empty.

When a data item is displayed by three dashes (---) on the GUI, this indicates that the value is either not applicable or cannot be retrieved. This is mapped to null values within the REST data model. Null valued elements do not appear in response body content.

Registration keys

A registration key is a unique id that identifies a single SIP user registration within the system. The key must be treated as an "opaque" unique identifier by the REST client. This is a server-side generated value composed of "<communication profile id>_<ip address>_<instance id>"

Portion	Value	Note
<communication profile id>	Long integer representing Session Manager Communication Profile of a user.	Fixed per user. It can appear in multiple registrations since Session Manager supports multiple device registrations.
<instance id>	String generated on the SM to uniquely identify multiply registered device.	Supported in release 6.3 and later

Portion	Value	Note
<ip address>		This is only provided for backward compatibility in pre 6.3 releases. It is not filled in release 6.3 and later releases.

Example

Registration key	Note
35761_	Placeholder record
35761_urn:uuid:00000000-0000-1000-8000-3cb15b611a39_	6.3 release and later with active registration
35761_148.147.173.169:5061	Pre 6.3 active registration

Registration cache and data availability

The registration cache is a volatile cache. If the GUI is not accessed for 30 minutes, the cache is removed from the System Manager memory. However, if the GUI is not left idle, the cache gets automatically reloaded in every ten minutes. It can also be reloaded when required by an administrator by pressing the refresh button on the GUI. Pressing the refresh button on the GUI resets the 10 minute timer. The REST client developer must be aware of this when a request is made. For certain cache states such as when cache is not loaded or the cache is in the process of reloading, REST request cannot be processed. In this case, the REST client receives a 503 - Service Unavailable reply along with an error message. The Retry-After HTTP header will be set with an estimated wait time.

The wait time is estimated based on last cache reload time or size. Reload time is a function of number of Session Managers, Users, and SIP Registrations. On a small system this is usually 15 seconds. On a high capacity system this could be in minutes. Minimum reload time is 10 seconds and maximum is 120 seconds.

A very large SM user configuration can result in 300 - 600 MB cache on System Manager.

Cache Unavailable Response

XML

```
Content-Encoding →gzip
Content-Length →259
Content-Type →application/xml
Date →Tue, 14 Mar 2017 22:19:10 GMT
Retry-After →15
Server →Apache-Coyote/1.1

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<error>
```

```
 <status>503</status>
<statusMessage>503 Service Unavailable</statusMessage>
<message>Registration cache not currently loaded</message>
<additionalStatus>CACHEUNLOADED</additionalStatus>
<additionalMessage>Retry in 15 seconds</additionalMessage>
<requestMethod>GET</requestMethod>
<requestURI>/registrations</requestURI>
<date>Tue Mar 14 16:19:10 MDT 2017</date>
```

JSON

```
Content-Encoding →gzip
Content-Length →259
Content-Type →application/xml
Date →Tue, 14 Mar 2017 22:19:10 GMT
Retry-After →15
Server →Apache-Coyote/1.1

{
status: 503
statusMessage: "503 Service Unavaialble"
message: "Registration cache not currently loaded"
additionalStatus: "CACHEUNLOADED"
additionalMessage: "Retry in 15 seconds"
requestMethod: "GET"
requestURI: "/registrations"
date: "Fri Nov 22 18:10:24 MST 2017"
}
```

Query expressions and results

Since there can be very large amounts of registrations, a mechanism of chunking the data into multiple requests is supported. Query control is accomplished by URI query parameters and related attributes returned. A query expression is used for both retrieving registrations in addition to applying a query filter to AST device notifications.

Query control values

Name	Value	Query parameter	Response element attribute
limit	Number of values to return per query. Query can return 0 - count elements. Using a limit with 0 is a way to get total count before performing an actual query.	X	
offset	Starting index to return values by incrementing "offset" with multiple calls. This provides a way to paginate a large data set.	X	
count	Number of records in response.		X
totalcount	Number of total records available.		X
query	Returns the request query made for this response.		X

Maximum query limit is 1000 for full response format Registration. And, maximum of 10000 for brief response format, when "&brief=true" is added as a query parameter. This does not mean that your configuration and HTTP clients or proxies can support this sort of volume. This is simply a server side limit to reduce server side impacts on resources.

A query result does not return 404 - Not Found, instead it returns a result payload with "count=0" attribute.

You must use clients that can support GZIP payload compression if you expect larger HTTP response bodies.

Query filter parameters

Multiple query parameters can be added to the URI in the form of "?param=value". Subsequent parameters can be added by using "&param2=value2". Multiple query parameters are logically "anded" together. Query parameter values might require Uuencoding.

The query values parallel GUI values that can be filtered on in the Registration GUI. The supported query parameter name is similar to the element name found in the response body content. Valid query parameters are: deviceType lastName controller secName ast remoteOffice deviceVendor ipAddress primName handle deviceVersion login sharedControl firstName primReg simultaneousDevices secReg deviceMac location deviceModel deviceSerial survName survReg.

Query value string expression matching

As with the GUI table filtering, the query values supply "starts with" matching. So "&lastName=Field" will match both "Field" and "FieldAndStream".

"%" is used as a wild card character matcher for strings. It will match any character and one or more characters. The wild card character can appear multiple times within a query parameter value. It can also

appear at beginning or end of string matching. It is handled very similarly to the SQL "like" expression matching.

Escaping "%" by using "%%" is not supported. In general, your REST client may require proper Uuencoding for using the wild card character. "%25" represents "%".

```
GET https: //{fqdn}/ASM/ws/registrations?ast=true&premReg=true?primSM=abc%25&brief=true
```

Query sort parameter

Only a single sort parameter is supported. It only supports ascending sort order and the sort uses a simple string comparator. The default sort value is "id" which is known internally as the User's SM Communication profile id. A database surrogate unique index field. Any of the query filter parameters elements returned as a response element can be used as a sort value. As there can be multiple registrations for a user this is not necessarily a unique sort index.

```
GET https:
//{fqdn}/ASM/ws/registrations?ast=true&premReg=true?primSM=abc%25&brief=true?sort=handle
```

Get a single SIP registration status by registration key or self-reference

Whenever a registration response is returned, it contains a self-reference. This self-reference is a valid URL for retrieving a single, unique registration based on registration key value.

Request	`GET https://{fqdn}/ASM/ws/registrations/<registrationkey>`
Optional Query parameters	`&format=<format>` `brief` just the URI link `normal:` basic registration value. It is the default parameter. `full:` basic data set plus time stamps and other details
Request Content	None.
Response Content	See Registration Response.
Response Errors	404: Not Found 503: Service Unavailable

Query multiple SIP registrations

Request	`GET https://{fqdn}/ASM/ws/registrations/<registrationkey>`

Optional Query Parameters	`&format=<format>` `brief:` just the URI link `normal:` basic registration values. It is the default format parameter value. `full:` basic data set plus time stamps and other details
Request Content	None.
Response Content	See Registration Response.
Response Errors	<table><tr><th>HTTP Reply Code</th><th>Additional Status</th><th>Message</th></tr><tr><td>503: Bad Request</td><td>INVALIDQUERY</td><td>Indicates error in filter or sort parameters or other general query format issue</td></tr><tr><td>503: Service Unavailable</td><td>CACHEUNLOADED</td><td>Registration cache not currently loaded</td></tr></table>

Registration Response

Brief Format

XML

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<registrations totalcount="1" query="?format=brief"
       offset="0" limit="1000" count="1">
       <registration>
              <link rel="reference"
                     href="https://localhost/ASM/ws/registrations/1_%3Curn:uuid:00000000-
0000-1000-8000-3cb15b611a39%3E" />
       </registration>
</registrations>
```

Normal Format - No Active Registration

A couple of ways to determine that a record is a "placeholder" record and not an active registration are:

- There is no <controller> element set
- None of the prim/sec/survController elements have a "(AC)" active controller indicator
- None of the prim/sec/survReg elements are true
- The self link href does not contain a "urn", also referred to as an instance id on the

GUI. The "ast" value is not reliable. There can be non-Avaya devices in use.

XML

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<registrations count="1" limit="0" offset="0" query=""
       totalcount="1">
       <registration>
              <ast>false</ast>
              <firstName>Mouse</firstName>
              <id>50</id>
              <lastName>Mickey</lastName>
              <location>Westminister</location>
              <login>mmouse@avaya.com</login>
              <primController></primController>
              <primName>cherryhills2</primName>
              <primReg>false</primReg>
              <remoteOffice>false</remoteOffice>
              <secController></secController>
              <secReg>false</secReg>
              <simultaneousDevices>0/1</simultaneousDevices>
              <survController></survController>
              <survReg>false</survReg>
              <visitingSurvName>SW 234 BSM</visitSurvName>
              <visitingSurvReg>true</visitSurvReg>
              <link href="https://augusta1.dr.avaya.com/ASM/ws/registrations/50_"
                     rel="self" />
```

Normal Format - Active Registration

XML

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<registrations totalcount="1" query="" offset="0" limit="1000"
       count="1">
       <registration>
              <ast>true</ast>
```

```
            <controller>smone</controller>
            <deviceMac>01:02:03:04:05:AF</deviceMac>
            <deviceModel>9650</deviceModel>
            <deviceType>SIP Phone</deviceType>
            <deviceVendor>Avaya</deviceVendor>
            <deviceVersion>2.6.1.18</deviceVersion>
            <deviceSerial>JBD00002345</deviceSerial>
            <firstName>fred</firstName>
            <id>1</id>
            <ipAddress>192.168.0.1</ipAddress>
            <lastName>flintstone</lastName>
            <location>alocation</location>
            <login>user1@avaya.com</login>
            <primController>(AC)</primController>
            <primReg>true</primReg>
            <regTimeActive>Wed Apr 12 14:20:48 PDT 2017</regTimeActive>
            <secController></secController>
            <secReg>true</secReg>
            <survController></survController>
            <survReg>true</survReg>
            <visitingSurvName>SW 234 BSM</visitSurvName>
            <visitingSurvReg>true</visitSurvReg>
            <link rel="self"
                  href="https://localhost/ASM/ws/registrations/1_%3Curn:uuid:00000000-
0000-1000-8000-3cb15b611a39%3E" />
      </registration>
</registrations>
```

Full Format

XML

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<registrations totalcount="1" query="?format=full?format=full"
      offset="0" limit="1000" count="1">
      <registration>
            <ast>true</ast>
            <controller>smone</controller>
            <deviceMac>01:02:03:04:05:AF</deviceMac>
            <deviceModel>9650</deviceModel>
            <deviceType>SIP Phone</deviceType>
            <deviceVendor>Avaya</deviceVendor>
```

```
            <deviceVersion>2.6.1.18</deviceVersion>
            <deviceSerial>JBD00002345</deviceSerial>
            <firstName>fred</firstName>
            <id>1</id>
            <ipAddress>192.168.0.1</ipAddress>
            <lastName>flintstone</lastName>
            <location>alocation</location>
            <login>user1@avaya.com</login>
            <primController>(AC)</primController>
            <primName>smone</primName>
            <primReg>true</primReg>
            <regInterruptDurationActive>17636 d 7 hr 26 min</regInterruptDurationActive>
            <regInterruptDurationPrim>17636 d 7 hr 26 min</regInterruptDurationPrim>
            <regInterruptDurationSec>17636 d 7 hr 26 min</regInterruptDurationSec>
            <regInterruptDurationSurv>17636 d 7 hr 26 min</regInterruptDurationSurv>
            <regTimeActive>Wed Apr 12 14:26:40 PDT 2017</regTimeActive>
            <regTimePrim>Wed Apr 12 14:26:40 PDT 2017</regTimePrim>
            <regTimeSec>Wed Apr 12 14:26:41 PDT 2017</regTimeSec>
            <regTimeSurv>Wed Apr 12 14:26:42 PDT 2017</regTimeSurv>
            <remoteOffice>true</remoteOffice>
            <secController></secController>
            <secName>smtwo</secName>
            <secReg>true</secReg>
            <sharedControl>true</sharedControl>
            <simultaneousDevices>1/3</simultaneousDevices>
            <survController></survController>
            <survName>bsm</survName>
            <survReg>true</survReg>
            <visitingSurvName>SW 234 BSM</visitSurvName>
            <visitingSurvReg>true</visitSurvReg>
            <link rel="self"
                  href="https://localhost/ASM/ws/registrations/1_%3Curn:uuid:00000000-
0000-1000-8000-3cb15b611a39%3E" />
      </registration>
</registrations>
```

Registration Query Response

XML

```
<registrations count="5" limit="1000" offset="0" query="?format=detail" totalcount="5">
    <registration>
        <actualLocation>Tibor1</actualLocation>
        <ast>true</ast>
        <controller>asmblade5</controller>
        <deviceMac>02:05:85:7f:eb:80</deviceMac>
        <deviceModel>96x1</deviceModel>
        <deviceVendor>Avaya</deviceVendor>
        <deviceVersion>7.1.9.0.1</deviceVersion>
        <firstName>Dennis</firstName>
         <fourthController></fourthController>
        <fourthReg>false</fourthReg>
       <handle>52003@avaya.com</handle>
        <id>101</id>
        <ipAddress>135.105.196.94:1026</ipAddress>
        <lastName>Sanger</lastName>
        <location>Thornton</location>
        <login>dsanger1@avaya.com</login>
        <policy>fixed</policy>
        <ppmSubscribeTimeActive>Fri Jan 08 09:18:25 MST 2021</ppmSubscribeTimeActive>
        <primController>(AC)</primController>
        <primName>asmblade5</primName>
        <primReg>true</primReg>
        <regInterruptDurationActive>0 d 0 hr 0 min</regInterruptDurationActive>
        <regInterruptDurationPrim>0 d 0 hr 0 min</regInterruptDurationPrim>
        <regTimeActive>Fri Jan 08 09:18:24 MST 2021</regTimeActive>
        <regTimePrim>Fri Jan 08 09:18:24 MST 2021</regTimePrim>
        <remoteOffice>false</remoteOffice>
        <secController></secController>
        <secReg>false</secReg>
        <sharedControl>false</sharedControl>
        <simultaneousDevices>1/10</simultaneousDevices>
        <survController></survController>
        <survReg>false</survReg>
       <thirdController></thirdController>
        <thirdReg>false</thirdReg>
        <visitingSurvReg>false</visitingSurvReg>
```

```
        <link href="https://smcp-
smgr1.dr.avaya.com/ASM/ws/registrations/101__%22%3Curn:uuid:00000000-0000-1000-8000-
0205857feb80%3E%22" rel="self"/>
    </registration>
</registrations>
```

AST Device Notification

An AST device notification can only be made to a registered device which has an "avaya_ccs_profile" event registration except for "forceunregister". “forceunregister” is supported for any device that can subscribe with a “reg” event. This indicated by the "ast" element value associated with the registration. If "true" then a notification is sent to the Session Manager which is the active controller for the registration.

The notification URI requests behave like an asynchronous call. The notification request process:

- Is "best attempt" but not guaranteed to have occurred. It will make an immediate request to SM instances if they are connected and not in Maintenance Mode.
- Will be throttled by the SM for large amounts of requests
- Will only make a request on active registration. "placeholder" records will be ignored
- Actual endpoint notification is asynchronous. The web service calls return upon complete transmission to the SM
- Might return an error during notifications made to multiple SMs. The error will be indicated as a stale data error. The command might have partially succeeded. This is a side effect of cache loading and variable service times.

Programmatic confirmation of a notification depends upon the notification type. For example, a "reboot" or "forceunregister" verification could be done based on re-registration time stamps. "Failback" could be verified by monitoring the controller value. You must use the User Registration GUI notification feature and monitor data changes to get a feel for what might be appropriate for programmatic confirmation if desired.

Currently supported notifications:

Notification or action	Value used in request query parameter	Note
Reboot	reboot	Default
Reload Complete	reboot	
Reload Config	reloadconfig	
Reload Contacts	reloadcontacts	
Failback	failback	

Notification or action	Value used in request query parameter	Note
Force unregister	forceunregister	This is not limited to AST devices

Request AST notification based on Query

Request	`POST https://{fqdn}/ASM/ws/registrations?notify=reboot&ast=true&pr imReg=true&primName=smOne`
	`POST https://{fqdn}/ASM/ws/registrations?notify=reboot&ast=true&de viceType=9461&deviceVersion=6.3.19`
	`At least one query parameter is required.`

Optional Query Parameters	`&notify=<AST notification>` Default is "reboot" if query parameter does not appear.
Request Content	None.
Response Content	See Notification Response
Response Errors	See Notification Request Errors

Request AST notification based on registration key

Request	`POST https://{fqdn}/ASM/ws/registrations/<registratio n key>?notify=reboot`
	Sends notification based on single registration key
	`POST https://{fqdn}/ASM/ws/registrations/keys?notify=failback`
	Sends a notification back for a larger list of registration keys

Optional Query Parameters	`&notify=<AST notification>` Default is "reboot" if query parameter does not appear.

Request Content	None.
Response Content	See Notification Response
Response Errors	See Notification Request Errors

Request AST notifications based on Session Manager

This request is made to all AST devices on the Session Manager with appropriate event in the registration. If a Session Manager is not connected or is in maintenance mode, a not found error is returned.

Request	`POST https://{fqdn}/ASM/ws/registrations/asm/name/<name>`
	`POST https://{fqdn}/ASM/ws/registrations/asm/name/East%20Coast%20SM`
	`POST https://{fqdn}/ASM/ws/registrations/asm/<id>`
	`POST https://{fqdn}/ASM/ws/registrations/asm/35767`
	<id> - is surrogate id available from Dashboard Status query of Session Manager instance information
	<name> - is as administered in Session Manager SIP Entity.

Optional Query Parameters	`&notify=<AST notification>` Default is "reboot" if notify query parameter does not appear.
Request Content	None.
Response Content	See Notification Response
Response Errors	See Notification Request Errors

Notification Request Errors

HTTP Reply Code	Additional Status	Message	Note
404 - Not Found	ASMNOTFOUND	Session Manager associated for registration notify not found	Generally, will occur due to bad SM id or name entry. Can less frequently occur during cache reload or a malformed Registration Key link request.
500 - Internal Server Error	DBERR	Error loading cache from database during cache reload	Unlikely this will be encountered.
500 - Internal Server Errort	JMXCONNERR	Problem connecting to Session Manager to transmit AST notify Request.	
409 - Conflict	STALEDATA	Notification request contains data which is out of date with the cache, a cache reload occurred	Request format of registration key is valid, but was not found during cache lookup. In most cases the endpoint has unregistered or changed it is active controller (failover, failback, etc.). Or there is a problem with accessing registrations from a SM.
503 - Bad Request	INVALIDQUERY	Various	Additional Message will indicate invalid query aspects, whether incorrect filter or sort parameters, or other problems with query construction.
503 - Bad Request	BADNOTIFYTYPE	Various	Additional Message will indicate valid notification types
503 - Bad Request	NOTIFYNOKEYS	Notification request contains no registration keys	Will occur for POST notify request with specific registration keys.
503 - Bad Request	NOTIFYBADKEY	Registration link must exist in request and contain href URI or hrefname containing registration key	Incorrectly formed Registration Key in request.

Notification Response

Rough indication of requests made to each Session Manager is provided in the requested or sent attributes. This is as per SM status result as well as roll up counts. The links in the <notificationstatus> element provide linkage into the Session Manager status REST calls. If the notification is based on per Session Manager, then these values are estimates based on summary table information. For query and key based requests, the counts matche the number of requests sent to each SM. The query attribute is set only when the query form is used. "unsent" could occur if the SM is in maintenance mode or is unavailable.

XML

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<notifystatuses sent="1" requested="1" notify="reboot">
       <notifystatus sent="1" requested="1">
              <asmstatus>
                     <link rel="reference" hrefname="smone"
href="https://localhost/ASM/ws/asmstatuses/1" />
              </asmstatus>
       </notifystatus>
</notifystatuses>
```

Chapter 5: Groups – Data Centers and Regions

Groups webservice API allows to query, create, update, and delete Session Manager groups. A Session Manager group can be a data center or a region. Follow the below given rules while using this webservice APIs.

1. The "Enable Policy Based Assignment of Session Managers" global setting must be enabled to work with groupType=region
2. Adding a region requires a minimum of one SM
3. An SM can only be assigned to a single data center
4. An SM can only be assigned to a single region
5. Unassign all SMs from a data center before deleting the data center
6. Regions can be deleted with SMs assigned
    a. SMs cannot be removed from a region
    b. If it is the last SM in a region
7. If it would reduce the SM count such that a location to region mapping would have less unique SMs then:
    a. Regions cannot be deleted if referenced by location to region or SM commprofiles with fixed-region policies
    b. Regions cannot be deleted if it is the last region and there are SM commprofiles with location-region policies

Get a single group by ID or name

Request	GET https://{fqdn}/ASM/ws/admin/asmgroups/32768 GET https://{fqdn}/ASM/ws/admin/asmgroups/name/Eastern%20Region
Request Content	None.
Response Content	<asmgroup> <description/> <name>Eastern Region</name> <groupType>data-center</groupType> <link href="https://{fqdn}/ASM/ws/admin/asmgroups/32768" rel="self"/> <asminstance> <link href="https://{fqdn}/ASM/ws/admin/asminstances/66667" hrefName="foxhollow1"/> </asminstance> </asmgroup>
Response Errors	404 – Not Found 500 – Server Error

Query groups

Request	GET https://{fqdn}/ASM/ws/admin/asmgroups GET https://{fqdn}/ASM/ws/admin/asmgroups?groupType=data-center&sort=name&limit=25
Optional Query parameters	name, groupType are supported as query arguments and as sort term. Count/limit/offset attributes support query control same as other resource queries.
Request Content	None
Response Content	<asmgroups count="2" limit="100" offset="0" totalcount="2"> <asmgroup/> <name>Eastern Region</name> <groupType>region</groupType> <link href="https://{fqdn}/ASM/ws/admin/asmgroups/32768" rel="self"/> <asminstance> <link href="https://{fqdn}/ASM/ws/admin/asmgroups/32768" hrefName="foxhollow1"/> </asminstance> </asmgroup> <asmgroup/> <name>DC1</name> <groupType>data-center</groupType> <link href="https://{fqdn}/ASM/ws/admin/asmgroups/32768" rel="self"/> <asminstance> <link href="https://{fqdn}/ASM/ws/admin/asminstance/234567" hrefName="foxhollow1"/> </asminstance> <asminstance> <link href="https://{fqdn}/ASM/ws/admin/asminstance/333454" hrefName="foxhollow2"/> </asminstance> </asmgroup> </asmgroups>
Response Errors	500 – Server Error

Add a group

Request	POST https://{fqdn}/ASM/ws/admin/asmgroups
Request Content	<asmgroup> <description/> <name>Eastern Region</name> <groupType>data-center</groupType>

	<link href="https://{fqdn}/ASM/ws/admin/asmgroups/32768" rel="self"/> <asminstance> <link href="https://{fqdn}/ASM/ws/admin/asminstances/32768" hrefName="foxhollow1"/> </asminstance> </asmgroup>
Response Content	<asmgroup> <description/> <name>Eastern Region</name> <groupType>data-center</groupType> <link href="https://{fqdn}/ASM/ws/admin/asmgroups/32768" rel="self"/> <asminstance> <link href="https://{fqdn}/ASM/ws/admin/asminstances/66667" hrefName="foxhollow1"/> </asminstance> </asmgroup>
Error Response	400 – Validation Error 500 – Server Error

Modify specified group fields

Request	POST https://{fqdn}/ASM/ws/admin/asmgroups/32768;description=testGroup
Request Content	None
Response Content	<asmgroups count="2" limit="100" offset="0" totalcount="2"> <asmgroup/> <name>Eastern Region</name> <groupType>region</groupType> <link href="https://{fqdn}/ASM/ws/admin/asmgroups/32768" rel="self"/> <asminstance> <link href="https://{fqdn}/ASM/ws/admin/asmgroups/32768" hrefName="foxhollow1"/> </asminstance> </asmgroup> <asmgroup/> <name>DC1</name> <groupType>data-center</groupType> <link href="https://{fqdn}/ASM/ws/admin/asmgroups/32768" rel="self"/> <asminstance> <link href="https://{fqdn}/ASM/ws/admin/asminstance/234567" hrefName="foxhollow1"/>

	</asminstance> <asminstance> <link href="https://{fqdn}/ASM/ws/admin/asminstance/333454" hrefName="foxhollow2"/> </asminstance> </asmgroup> </asmgroups>
Response Errors	400 – Validation Error 500 – Server Error

Replace an existing group by ID or name

Request	PUT https://{fqdn}/ASM/ws/admin/asmgroups/32768 PUT https://{fqdn}/ASM/ws/admin/asmgroups/name/Northern%DC1
Request Content	<asmgroup> <description/> <name>Eastern Region</name> <groupType>data-center</groupType> <link href="https://{fqdn}/ASM/ws/admin/asmgroups/32768" rel="self"/> <asminstance> <link href="https://{fqdn}/ASM/ws/admin/asminstances/32768" hrefName="foxhollow1"/> </asminstance> </asmgroup>
Response Content	<asmgroup> <description/> <name>Eastern Region</name> <groupType>data-center</groupType> <link href="https://{fqdn}/ASM/ws/admin/asmgroups/32768" rel="self"/> <asminstance> <link href="https://{fqdn}/ASM/ws/admin/asminstances/66667" hrefName="foxhollow1"/> </asminstance> </asmgroup>
Response Errors	400 – Validation Error 500 – Server Error

Delete a group by ID or name

Request	DELETE https://{fqdn}/ASM/ws/admin/asmgroups/32768 DELETE https://{fqdn}/ASM/ws/admin/asmgroups/name/DC1
Request Content	None
Response Content	None

Response Errors	404 – Not found 500 – Server Error

Add an SM instance to an existing group

Note: Remember that a SM can only belong to a single region or data-center group type

Request	POST https://{fqdn}/ASM/ws/admin/asmgroups/32768/asminstances
Request Content	<asminstance> <link href="https://{fqdn}/ASM/ws/admin/asminstances/32768" hrefName="foxhollow1"/> </adminstance>
Response	<asmgroup> <description/> <name>Eastern Region</name> <groupType>data-center</groupType> <link href="https://{fqdn}/ASM/ws/admin/asmgroups/32768" rel="self"/> <asminstance> <link href="https://{fqdn}/ASM/ws/admin/asminstances/66667" hrefName="foxhollow1"/> </asminstance> </asmgroup>
Response Errors	404 – Not found 400 – Validation Error 500 – Server Error

Remove an SM instance from an existing group

Request	DELETE https://{fqdn}/ASM/ws/admin/asmgroups/32768/asminstances/294912
Request Content	None
Response	<asmgroup> <description/> <name>Eastern Region</name> <groupType>data-center</groupType> <link href="https://{fqdn}/ASM/ws/admin/asmgroups/32768" rel="self"/> <asminstance> <link href="https://{fqdn}/ASM/ws/admin/asminstances/66667" hrefName="foxhollow1"/>

	</asminstance> </asmgroup>
Response Errors	404 – Not found 400 – Validation Error 500 – Server Error

Chapter 6: Location to Region

Location to Region webservice API allows to query, create, update, and delete location to Region resources. Follow the below given rules while using this webservice APIs.

- Ensure that the "Enable Policy Based Assignment of Session Managers" global setting is enabled and region(s) are administered.
- There can be a region [1-4] assigned or none to indicate the location doesn't map to any regions
- Regions can be duplicated through region [1-4] if needed.
- Regions must be assigned sequentially that is, region1 must be assigned before region2

Get a single location to region by ID or location name

Note that region {id} or name value is the same as given in the Routing location resources URI https://{fqdn}/NRP/admin/locations/{id}. Consider it a synonym.

Request	GET https://{fqdn}/ASM/ws/locationregions/32768 GET https://{fqdn}/ASM/ws/locationregions/name/Northern%20location
Request Content	NA
Response Content	<?xml version="1.0" encoding="UTF-8" standalone="yes"?> <locationregion> <description>Project Alaska. swe 03/13/13</description> <link href="https://augusta3.dr.avaya.com/ASM/ws/locationregions/131089" rel="self"/> <location> <link href="https://augusta3.dr.avaya.com/NRP/admin/locations/131089" hrefName="Alaska-SBC" rel="reference" title="Project Alaska. swe 03/13/13"/> </location> <name>Alaska-SBC</name> <region1> <link href="https://augusta3.dr.avaya.com/ASM/ws/asmgroups/33" rel="reference"/> </region1> <region2> <link href="https://augusta3.dr.avaya.com/ASM/ws/asmgroups/34" rel="reference"/> </region2> <region3> <link href="https://augusta3.dr.avaya.com/ASM/ws/asmgroups/35" rel="reference"/> </region3> <region4> <link href="https://augusta3.dr.avaya.com/ASM/ws/asmgroups/36" rel="reference"/>

	</region4> </locationregion>
Response Errors	404 – Not Found 500 – Server Error

Query location to regions

Request	GET https://{fqdn}/ASM/ws/locationregions GET https://{fqdn}/ASM/ws/locationregions?name=A%20location
Optional query parameters	offset - starting record number to return, allows chunking of large data sets, defaults to 0 limit- number of query records to return, current maximum is 1000 for normal format, 10000 if brief format. For example .../ASM/ws/locationregions?limit=50 brief - just return URI references only, as in "brief listing". For example .../ASM/ws/locationregions?brief=true
Optional Query Filtering and Sort Parameters	&name=<value> Where value will be matched exactly against Session Manager name. The comparison uses a "starts with" matching. Same as filtering values within the GUI.
Request Content	None
Response Content	<?xml version="1.0" encoding="UTF-8" standalone="yes"?> <locationregions count="24" limit="1000" offset="0" query="" totalcount="24"> <locationregion> <description></description> <link href="https://augusta3.dr.avaya.com/ASM/ws/asminstances/131087" rel="self"/> <location> <link href="https://augusta3.dr.avaya.com/NRP/admin/locations/131087" hrefName="AAM-Green" rel="reference" title=""/> </location> <name>AAM-Green</name> <region1> <link href="https://augusta3.dr.avaya.com/ASM/ws/asmgroups/33" rel="reference"/> </region1> <region2> <link href="https://augusta3.dr.avaya.com/ASM/ws/asmgroups/34" rel="reference"/> </region2> </locationregion> <locationregion> <description>Precedence calling</description> <link

	href="https://augusta3.dr.avaya.com/ASM/ws/asminstances/131088" rel="self"/> <location> <link href="https://augusta3.dr.avaya.com/NRP/admin/locations/131088" hrefName="GreenCM01-MLPP" rel="reference" title="Precedence calling"/> </location> <name>GreenCM01-MLPP</name> </locationregion> <locationregion> <description>Project Alaska. swe 03/13/13</description> <link href="https://augusta3.dr.avaya.com/ASM/ws/asminstances/131089" rel="self"/> <location> <link href="https://augusta3.dr.avaya.com/NRP/admin/locations/131089" hrefName="Alaska-SBC" rel="reference" title="Project Alaska. swe 03/13/13"/> </location> <name>Alaska-SBC</name> <region1> <link href="https://augusta3.dr.avaya.com/ASM/ws/asmgroups/33" rel="reference"/> </region1> <region2> <link href="https://augusta3.dr.avaya.com/ASM/ws/asmgroups/34" rel="reference"/> </region2> <region3> <link href="https://augusta3.dr.avaya.com/ASM/ws/asmgroups/35" rel="reference"/> </region3> <region4> <link href="https://augusta3.dr.avaya.com/ASM/ws/asmgroups/36" rel="reference"/> </region4> </locationregion> .... </locationregions>
Response Errors	404 – Not Found 500 – Server Error

Add regions to a location

This version uses a region name only, the following form supports ID and name based locationregions URIs

Request	POST https://{fqdn}/ASM/ws/locationregions/name/Eastern%20location;region1=Western%20Region
Request Content	NA
Response Content	<?xml version="1.0" encoding="UTF-8" standalone="yes"?> <locationregion> <description>Project Alaska. swe 03/13/13</description> <link href="https://augusta3.dr.avaya.com/ASM/ws/locationregions/131089" rel="self"/> <location> <link href="https://augusta3.dr.avaya.com/NRP/admin/locations/131089" hrefName="Alaska-SBC" rel="reference" title="Project Alaska. swe 03/13/13"/> </location> <name>Alaska-SBC</name> <region1> <link href="https://augusta3.dr.avaya.com/ASM/ws/asmgroups/33" rel="reference"/> </region1> <region2> <link href="https://augusta3.dr.avaya.com/ASM/ws/asmgroups/34" rel="reference"/> </region2> <region3> <link href="https://augusta3.dr.avaya.com/ASM/ws/asmgroups/35" rel="reference"/> </region3> <region4> <link href="https://augusta3.dr.avaya.com/ASM/ws/asmgroups/36" rel="reference"/> </region4> </locationregion>
Response Errors	400 – Validation Error 500 – Server Error

Add regions to a location

Request	PUT https://{fqdn}/ASM/ws/locationregions/2828282
Request Content	<locationregion> <region1> <link href="https://augusta3.dr.avaya.com/ASM/ws/asmgroups/33" hrefName="Eastern region" rel="reference"/> </region1> <region2> <link

	href="https://augusta3.dr.avaya.com/ASM/ws/asmgroups/34" hrefName="Western region" rel="reference"/> </region2> <region3> <link href="https://augusta3.dr.avaya.com/ASM/ws/asmgroups/35" hrefName="Western region" rel="reference"/> </region3> <region4> <link hrefName="North Region" rel="reference"/> </region4> </locationregion>
Response Content	<?xml version="1.0" encoding="UTF-8" standalone="yes"?> <locationregion> <description>Project Alaska. swe 03/13/13</description> <link href="https://augusta3.dr.avaya.com/ASM/ws/locationregions/131089" rel="self"/> <location> <link href="https://augusta3.dr.avaya.com/NRP/admin/locations/131089" hrefName="Alaska-SBC" rel="reference" title="Project Alaska. swe 03/13/13"/> </location> <name>Alaska-SBC</name> <region1> <link href="https://augusta3.dr.avaya.com/ASM/ws/asmgroups/33" rel="reference"/> </region1> <region2> <link href="https://augusta3.dr.avaya.com/ASM/ws/asmgroups/34" rel="reference"/> </region2> <region3> <link href="https://augusta3.dr.avaya.com/ASM/ws/asmgroups/35" rel="reference"/> </region3> <region4> <link href="https://augusta3.dr.avaya.com/ASM/ws/asmgroups/36" rel="reference"/> </region4> </locationregion>
Response Errors	400 – Validation Error 500 – Server Error

Chapter 7: Session Manager Instances

The REST API for Session Manager instances provides a set of HTTP endpoints for creating, retrieving, updating, and deleting instances. The API follows the principles of Representational State Transfer (REST) and uses the standard HTTP methods: GET, POST and DELETE. A summary of the API is as follows:

GET /instances: This endpoint retrieves a list of all Session Manager instances.

GET /instances/{id}: This endpoint retrieves the details of a specific Session Manager instance.

POST /instance: This endpoint creates a new Session Manager instance.

POST /instance/{id}:{attribute}={value}: This endpoint updates a specific Session Manager instance.

DELETE /instance/{id}: This endpoint deletes a specific Session Manager instance.

The following sections details the usages of each endpoint.

Get all SM/BSM instances

Request	`GET https://{fqdn}/ASM/ws/instances`
Request Content	None.
Response Content	See SM Instance response content section.
Response Errors	404: Not Found

Get a single SM/BSM instance by ID

Request	`GET https://{fqdn}/ASM/ws/instances/{id}` id is the numeric surrogate id, which is the last value of the self-reference URI or "id" element.
Request Content	None.

Response Content	See SM Instance response content section.
Response Errors	404: Not Found

Create SM/BSM Instance

Request	`POST https://{fqdn}/ASM/ws/asminstance`
Request Content	See SM instance request body sample section
Response Content	See SM instance response body sample section
Response Errors	404: Not found 500: Various validation errors when attributes are missing or invalid. The response contains a specific attribute that is in error.

Edit SM/BSM instance

Once a SM/BSM instance is created, individual attributes can be edited. The parameters editable are based on the content used for creating an instance.

Request	`POST https://{fqdn}/ASM/ws/asminstances/{id}:{parameter}={value}`
Request Content	None.
Response Content	See SM instance response body sample section
Response Errors	404: Not Found

Delete SM instance

Request	`DELETE https://{fqdn}/ASM/ws/asminstance/{id}`

Request Content	None
Response Content	200: OK – if successfully deleted
Response Errors	404: Not Found - if doesn't exist 500: Instance is referenced in Communication Profiles

SM instance response content sample

The response contains a URI to the Session Manager instance and the underlying SIP entity.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
    <asminstances count="1" limit="1000" offset="0" query="" totalcount="1">
            <asminstance>
                <assetDefaultGateway>140.129.50.1</assetDefaultGateway>
                <assetNetMask>255.255.255.0</assetNetMask>
                <callControlPHB>46</callControlPHB>
                <cdrFileFormat>standard_flat</cdrFileFormat>
                <cdrIncompleteCalls>false</cdrIncompleteCalls>
                <cdrUserToUserCalls>false</cdrUserToUserCalls>
                <crlfKeepAliveSecs>0</crlfKeepAliveSecs>
                <errorOnNotifyFailure>false</errorOnNotifyFailure>
                <honorPhaseEqualEndpoint>true</honorPhaseEqualEndpoint>
                <isCdrEnabled>false</isCdrEnabled>
                <isMonitoring>true</isMonitoring>
                <isPpmPktRateLimiting>true</isPpmPktRateLimiting>
                <maintenanceMode>true</maintenanceMode>
                <managementAccessPointHostname>140.129.50.111</managementAccessPointHostname>
                <managementAccessPointPortNumber>11099</managementAccessPointPortNumber>
                <maxConnsPerPpm>3</maxConnsPerPpm>
                <monitorOkResponses>1</monitorOkResponses>
                <monitorProactiveSecs>900</monitorProactiveSecs>
                <monitorReactiveSecs>120</monitorReactiveSecs>
                <monitorRetries>1</monitorRetries>
                <name>my_sm</name>
                <network_topology>enterprise</network_topology> ← only when Edge Topology enabled
                <ppmConnTimeoutMins>5</ppmConnTimeoutMins>
                <ppmPktRateLimitThres>25</ppmPktRateLimitThres>
                <sipFirewallName>BSM 6.3.8.0</sipFirewallName>
                <smType>SM</smType>
                <link href="https://192.168.2.200/ASM/ws/asminstances/12" rel="self"/>
                <sipentity>
                    <link href=https://192.168.2.200/NRP/admin/sipentities/32768
                         hrefName="my_sm" rel="reference" title=""/>
                </sipentity>
            </asminstance>
        </asminstances>
```

SM instance request body sample

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<asminstance>
        <smType>SM</smType>  ← For branch SMs use BSM
        <sipentity>
          <link href="https://192.168.2.200/NRP/admin/sipentities/32768" rel="self"/>
        </sipentity>
        <name>edge_sm</name>
        <remoteAccessConfig>test1</remoteAccessConfig>   ← for Edge Topology BSM only
        <assetDefaultGateway>140.129.50.1</assetDefaultGateway>
        <managementAccessPointHostname>140.129.50.111</managementAccessPointHostname>
        <assetNetMask>255.255.255.0</assetNetMask>
        <callControlPHB>46</callControlPHB>
        <cdrFileFormat>standard_flat</cdrFileFormat>
        <cdrIncompleteCalls>false</cdrIncompleteCalls>
        <cdrUserToUserCalls>false</cdrUserToUserCalls>
        <crlfKeepAliveSecs>0</crlfKeepAliveSecs>
        <errorOnNotifyFailure>false</errorOnNotifyFailure>
        <honorPhaseEqualEndpoint>true</honorPhaseEqualEndpoint>
        <isCdrEnabled>false</isCdrEnabled>
        <isMonitoring>true</isMonitoring>
        <isPpmPktRateLimiting>true</isPpmPktRateLimiting>
        <maintenanceMode>true</maintenanceMode>
        <managementAccessPointPortNumber>11099</managementAccessPointPortNumber>
        <maxConnsPerPpm>3</maxConnsPerPpm>
        <monitorOkResponses>1</monitorOkResponses>
        <monitorProactiveSecs>900</monitorProactiveSecs>
        <monitorReactiveSecs>120</monitorReactiveSecs>
        <monitorRetries>1</monitorRetries>
        <ppmConnTimeoutMins>5</ppmConnTimeoutMins>
        <ppmPktRateLimitThres>25</ppmPktRateLimitThres>
        <sipFirewallName>BSM 6.3.8.0</sipFirewallName>
</asminstance>
```

Chapter 8: SDK Java JAX-RS Examples

The examples given are very simple code snippets to provide examples of SM EM JAXB mapping examples for Java clients. They are only to provide Java analogs for the more complete XML and JSON reference examples. Before running any of these samples, make sure there are some Session Manager instances and SIP Users administered with Session Manager Communication Profiles on the target System Manager.

Working with the Java samples

The sample code in the SDK is tested with Jersey 2.25.1 RI.

Do the following:

1. Download Jersey
2. Download JUnit (include the hamcrest jar) into your desired development environment.
3. Copy in asmelementmgr_jaxb.jar and sample source from the SDK
4. Create CA trust store and configure into the test environment

The sample/Common.java file needs the following 3 statics modified for your URI configuration and login credentials:

```
static public String URLBASE = "https://turtlebay7.dr.avaya.com/ASM/ws";
       static public String ADMINID = "admin";
       static public String ADMINPASSWORD = "Admin123$";
```

5. Run the sample as JUnit tests.

JAX-RS Client Setup

The Client setup is very dependent on the underlying jax-rs provider.
Key aspects to consider for setting up a jax-rs Client:

- SSL/TLS configuration, all URLs need to begin with "https://"
- Basic Authentication mechanism and injecting login/password

A generalized Client creation routine might look as follows in Oracle Jersey 2.6 (Note: SSL setup differs between Jersey versions). JBoss RESTEasy and Apache CXF will require notably different approaches for configuring clients.

```
static public Client getClient() {
    SslConfigurator sslConfig =
SslConfigurator.newInstance().trustStoreFile("./resources/sample.jks").trustStorePassword
("sample").securityProtocol("TLS");
    SSLContext sslContext = sslConfig.createSSLContext();
    Client client =
ClientBuilder.newBuilder().sslContext(sslContext).register(getAuthFilter()).build();
    return client;
}
```

A very simple HTTP basic authentication filter that does not handle preemptive requests can be as follows:

```
public static class BasicAuth implements ClientRequestFilter {
    private final String user;
    private final String password;
    public BasicAuth(String user, String password) {
        this.user = user;
        this.password = password;
    }
    public void filter(ClientRequestContext requestContext) throws IOException {
        MultivaluedMap<String, Object> headers = requestContext.getHeaders();
        final String basicAuthentication = getBasicAuthentication();
        headers.add("Authorization", basicAuthentication);
    }
    private String getBasicAuthentication() {
        String token = this.user + ":" + this.password;
        try {
            return "BASIC " + DatatypeConverter.printBase64Binary(token.getBytes("UTF-
8"));
        } catch (Exception ex) {
            throw new IllegalStateException("Cannot encode with UTF-8", ex);
        }
    }
}
```

Session Manager Status

```
/*
               * Query all statuses.
               */

              WebTarget target = client.target(Common.URLBASE).path("asmstatuses");
              Response response = target.request(MediaType.APPLICATION_XML).get();
              Common.checkError(response);
              /*
               * In this case we are manually telling what entity type is to be
               * unmarshalled. Many clients allow registration of the JAXB mapping
               * classes so this is not needed.
               */
              AsmStatuses statuses = response.readEntity(AsmStatuses.class);
              System.out.println("Recieved " + statuses.getCount() + " of "
```

```
                        + statuses.getTotalcount() + " statuses");
            for (AsmStatus astatus : statuses.getStatus()) {
                  System.out.println("Recieved status for SM: " + astatus.getName());
            }
            /*
             * Now take first status found and GET based on reference URI.
             * This is not required, it is just a find of a particular SM status based
on
         * reference URI.
             */
            AsmStatus status = statuses.getList().get(0);
            target = client.target(status.getLink().getHref());
            response = target.request(MediaType.APPLICATION_XML).get();
            Common.checkError(response);
            AsmStatus status2 = response.readEntity(AsmStatus.class);
            System.out.println("Recieved status for SM: " + status2.getName());
            System.out.println("SM has release: " +  status2.getRelease());
            System.out.println("Deny New Service: " + status2.getDenyNewService());
```

Session Manger Operation Request

```
Client client = Common.getClient();
            /*
             * Query to get first SM URI
             */

            WebTarget target = client.target(Common.URLBASE).path("asmstatuses");
            Response response = target.request(MediaType.APPLICATION_XML).get();
            Common.checkError(response);
            /*
             * In this case we are manually telling what entity type is to be
             * unmarshalled. Many clients allow registration of the JAXB mapping
             * classes so this is not needed.
             */
            AsmStatuses statuses = response.readEntity(AsmStatuses.class);
            System.out.println("Recieved " + statuses.getCount() + " of "
                        + statuses.getTotalcount() + " statuses");
            /*
             * Now take first status found and GET based on reference URI.
             * Let's turn maintenance mode off, and set Accept New Service.
```

```
             * The later is a little odd in that control Accept New Service by
             * turning "off" Deny New Service.
             *
             * There is no request body nor reply body for these URIs
             */
            AsmStatus status = statuses.getList().get(0);
            String smuri = status.getLink().getHref().toString();
            target = client.target(smuri + "/maintenanceMode/false");
            response = target.request(MediaType.APPLICATION_XML).post(null);
            Common.checkError(response);

            // Give some time to allow for state change and network latency
            Thread.currentThread().sleep(10000);
            /*
             * The recommended approach for setting this state would be "set and test".
             * E.g. wait a bit and see if Accept New Service has been entered. There is
some delay
             * since this is an operation request sent through SMGR and then sent to the
particular SM.
             * Usually takes a few seconds for the SM to stablize the new state.
             */
            target = client.target(smuri + "/denyNewService/false");
            response = target.request(MediaType.APPLICATION_XML).post(null);
            Common.checkError(response);
```

Registration Status

```
Client client = Common.getClient();
            /*
             * Query all statuses.
             */
            WebTarget target = client.target(Common.URLBASE).path("registrations");
            Response response = target.request(MediaType.APPLICATION_XML).get();
            if (response.getStatus() == 503) {
                  // Registration cache not loaded, let's wait a bit.
                  Thread.currentThread().sleep(15000);
                  response = target.request(MediaType.APPLICATION_XML).get();
            }
            Common.checkError(response);
            /*
```

```
               * In this case we are manually telling what entity type is to be
               * unmarshalled. Many clients allow registration of the JAXB mapping
               * classes so this is not needed.
               */
              Registrations regs = response.readEntity(Registrations.class);
              System.out.println("Recieved " + regs.getCount() + " of " +
regs.getTotalcount() + " statuses");
              for (Registration reg : regs.getRegistration()) {
                     System.out.println("Recieved status for login: " + reg.getLogin());
              }
              /*
               * Now take first status found and GET based on reference URI.
               */
              Registration reg = regs.getList().get(0);
              target = client.target(reg.getLink().getHref());
              response = target.request(MediaType.APPLICATION_XML).get();
              Common.checkError(response);
              reg = response.readEntity(Registration.class);
              System.out.println("Recieved status for user login: " + reg.getLogin());
              System.out.println("Controller (no controller means not actice
registration): " + reg.getController());
              System.out.println("Primary controller: " + reg.getPrimName());
              /*
               * Now get the status via a query.
               */
              String login = reg.getLogin();
              target =
client.target(Common.URLBASE).path("registrations").queryParam("login", login)
                           .queryParam("limit", "1");
              response = target.request(MediaType.APPLICATION_XML).get();
              Common.checkError(response);
              regs = response.readEntity(Registrations.class);
              System.out.println("Recieved " + regs.getCount() + " of " +
regs.getTotalcount() + " statuses");
```

Registration Notification Request

```
Client client = Common.getClient();
              /*
               * Query all statuses.
               */
```

```
            WebTarget target = client.target(Common.URLBASE).path("registrations");
            Response response = target.request(MediaType.APPLICATION_XML).get();
            if (response.getStatus() == 503) {
                  // Registration cache not loaded, let's wait a bit.
                  Thread.currentThread().sleep(15000);
                  response = target.request(MediaType.APPLICATION_XML).get();
            }
            Common.checkError(response);
            /*
             * In this case we are manually telling what entity type is to be
             * unmarshalled. Many clients allow registration of the JAXB mapping
             * classes so this is not needed.
             */
            Registrations regs = response.readEntity(Registrations.class);
            System.out.println("Recieved " + regs.getCount() + " of " +
regs.getTotalcount() + " statuses");
            for (Registration reg : regs.getRegistration()) {
                  System.out.println("Recieved status for login: " + reg.getLogin());
            }
            /*
             * Now take first status found send notification based on reference URI.
             */
            Registration reg = regs.getList().get(0);
            target = client.target(reg.getLink().getHref()).queryParam("notify",
"reboot");
            response = target.request(MediaType.APPLICATION_XML).post(null);
            Common.checkError(response);
            RegistrationNotifyStatuses statuses =
response.readEntity(RegistrationNotifyStatuses.class);
            System.out.println("Requested " + statuses.getRequested() + " and sent " +
statuses.getSent());
            /*
             * Now send notification based on query.
             */
            String login = reg.getLogin();
            target =
client.target(Common.URLBASE).path("registrations").queryParam("login", login)
                        .queryParam("notify", "reboot");
            response = target.request(MediaType.APPLICATION_XML).post(null);
            Common.checkError(response);
            statuses = response.readEntity(RegistrationNotifyStatuses.class);
            System.out.println("Requested " + statuses.getRequested() + " and sent " +
```

```
statuses.getSent());
              for (RegistrationNotifyStatus status : statuses.getList()) {
                     System.out.println(" For sm " +
status.getStatusLink().getLink().getHrefName() + " requested "
                                   + status.getRequested() + " sent " + status.getSent());
              }
```

Chapter 9: Related resources

Session Manager Documentation

The following table lists the documents related to Session Manager. Download the documents from the Avaya Support website at http://support.avaya.com

Title	Description	Audience
Overview		
Avaya Aura® Session Manager Overview and Specification	Describes the key features of Session Manager.	IT management System administrators
Avaya Aura® Session Manager Security Design	Describes the security considerations, features, and solutions for Session Manager.	Network administrators, services, and support personnel
Implementation		
Deploying Avaya Aura® Session Manager and Avaya Aura® Branch Session Manager in Virtualized Environment	Describes how to deploy the Session Manager virtual application in a virtualized environment.	Services and support personnel
Deploying Avaya Aura® Session Manager in Infrastructure as a Service Environment	Describes how to deploy the Session Manager in the Infrastructure as a Service (IaaS) environment.	Services and support personnel
Deploying Avaya Aura® Session Manager and Avaya Aura® Branch Session Manager in Software-Only Environment	Describes how to deploy the Session Manager in the Software-Only environment.	Services and support personnel
Deploying Avaya Aura® Session Manager and Avaya Aura® Branch Session Manager in Virtual Appliance	Describes how to deploy the Session Manager in Virtual Appliance.	Services and support personnel
Routing Web Service API Programming Reference	Describes how to use the System Manager Routing Web Service API for Session Manager.	Services and support personnel
Administration		
Administering Avaya Aura® Session Manager	Describes the procedures to administer Session Manager using System Manager.	System administrators
Avaya Aura® Session Manager Case Studies	Provides common administration scenarios.	System administrators
Upgrades		

Related Resources

Upgrading Avaya Aura® Session Manager	Describes the procedures to upgrade Session Manager to the latest software release.	Services and support personnel
Maintaining and Troubleshooting		
Maintaining Avaya Aura® Session Manager	*Maintaining Avaya Aura® Session Manager*	*Maintaining Avaya Aura® Session Manager*
Contains the procedures for maintaining Session Manager.	Contains the procedures for maintaining Session Manager.	Contains the procedures for maintaining Session Manager.

Finding documents on the Avaya Support website

Procedure

1. Go to https://support.avaya.com.
2. At the top of the screen, type your username and password and click **Login**.
3. Click **Support by Product** > **Documents**.
4. In **Enter your Product Here**, type the product name and then select the product from the list.
5. In **Choose Release**, select an appropriate release number.
6. In the **Content Type** filter, click a document type, or click **Select All** to see a list ofall available documents.

   For example, for user guides, click **User Guides** in the **Content Type** filter. The list displays the documents only from the selected category.
7. Click **Enter**.

Avaya Documentation Portal navigation

Customer documentation for some programs is now available on the Avaya Documentation Portal at https://documentation.avaya.com.

Important:

For documents that are not available on the Avaya Documentation Portal, click **Support** on the top menu to open https://support.avaya.com.

Using the Avaya Documentation Portal, you can:

- Search for content in one of the following ways:
  - Type a keyword in the **Search** field.
  - Type a keyword in **Search**, and click **Filters** to search for content by product, release, and document type.
  - Select a product or solution and then select the appropriate document from the list.

Related Resources

- Find a document from the **Publications** menu.
- Publish a PDF of the current section in a document, the section and its subsections, or the entire document.
- Add content to your collection by using **My Docs** (☆).

  Navigate to the **My Content** > **My Docs** menu, and do any of the following:

  - Create, rename, and delete a collection.
  - Add content from various documents to a collection.
  - Save a PDF of selected content in a collection and download it to your computer.
  - Share content in a collection with others through email.
  - Receive content that others have shared with you.
- Add yourself as a watcher by using the **Watch** icon (👁).

  Navigate to the **My Content** > **Watch list** menu, and do the following:

  - Set how frequently you want to be notified, starting from every day to every 60 days.
  - Unwatch selected content, all content in a document, or all content on the Watch list page.

  As a watcher, you are notified when content is updated or deleted from a document, or the document is removed from the portal.
- Share a section on social media platforms, such as Facebook, LinkedIn, Twitter, and Google +.
- Send feedback on a section and rate the content.

Note:

Some functionality is only available when you log in to the portal. The available functionality depends on the role with which you are logged in.

Training

The following table contains courses that are available on https://www.avaya-learning.com. To search for the course, in the **Search** field, enter the course code and click **Go**.

New training courses are added periodically. Enter **Session Manager** in the **Search** field to display the inclusive list of courses related to Session Manager.

Course code	Course title
20980W	What's New with Avaya Aura® Release 8.1
71200V	Integrating Avaya Aura® Core Components
72200V	Supporting Avaya Aura® Core Components
20130V	Administering Avaya Aura® System Manager Release 8.1

Viewing Avaya Mentor videos

Avaya Mentor videos provide technical content on how to install, configure, and troubleshoot Avaya products.

About this task

Videos are available on the Avaya Support website, listed under the video document type, and on the Avaya-run channel on YouTube.

Procedure

- To find videos on the Avaya Support website, go to https://support.avaya.com/ and do one of the following:
  - In **Search**, type `Avaya Mentor Videos` to see a list of the available videos.
  - In **Search**, type the product name. On the Search Results page, select **Video** in the **Content Type** column on the left.
- To find the Avaya Mentor videos on YouTube, go to www.youtube.com/AvayaMentor and do one of the following:
  - Enter a key word or key words in the **Search Channel** to search for a specific product or topic.
  - Scroll down Playlists, and click the name of a topic to see the available list of videos posted on the website.

Note:

Videos are not available for all products.

Support

Go to the Avaya Support website at https://support.avaya.com for the most up-to-date documentation, product notices, and knowledge articles. You can also search for release notes, downloads, and resolutions to issues. Use the online service request system to create a service request. Chat with live agents to get answers to questions, or request an agent to connect you to a support team if an issue requires additional expertise.

Using the Avaya InSite Knowledge Base

The Avaya InSite Knowledge Base is a web-based search engine that provides:

- Up-to-date troubleshooting procedures and technical tips
- Information about service packs
- Access to customer and technical documentation
- Information about training and certification programs
- Links to other pertinent information

If you are an authorized Avaya Partner or a current Avaya customer with a support contract, you can access the Knowledge Base without extra cost. You must have a login account and a valid Sold-To number.

Use the Avaya InSite Knowledge Base for any potential solutions to problems.

1. Go to http://www.avaya.com/support.
2. Log on to the Avaya website with a valid Avaya user ID and password. The system displays the Avaya Support page.
3. **Click** Support by Product > Product Specific Support.
4. In **Enter Product Name**, enter the product, and press `Enter`.
5. Select the product from the list, and select a release.
6. Click the **Technical Solutions** tab to see articles.
7. Select relevant article

Chapter 9: Glossary

Keyword	Description
SM	Session Manager
AJAX	Asynchronous JavaScript and XML
DOM	Document Object Model
JSON	JavaScript Object Notation
XHR	XMLHttpRequest
MVC	Model View Controller
REST	Representational State Transfer
CSS	Cascading Style Sheet