Avaya Identity Engines Ignition Guest Manager REST APIs

Release 9.2
NN47280-505
Issue 01.01
August 2015

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 by Avaya in varying mediums which may include product information, operating instructions and performance specifications that Avaya may generally make available to users of its products and Hosted Services. 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 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 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.

Licenses

THE SOFTWARE LICENSE TERMS AVAILABLE ON THE AVAYA WEBSITE, HTTP://SUPPORT.AVAYA.COM/LICENSEINFO 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. 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 Designated Processor that hosts 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 on a number of Designated Processors up to the number indicated in the order. 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.

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 currently available for license from Avaya is the software contained within the list of Heritage Nortel Products located at http://support.avaya.com/LicenseInfo under the link “Heritage Nortel Products” or such successor site as designated by Avaya. For Heritage Nortel Software, Avaya grants You 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

Each product has its own ordering code and license types. Note that 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: http://support.avaya.com/Copyright or such successor site as designated by Avaya. You agree to the Third Party Terms for any such Third Party Components.

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: http://support.avaya.com or such successor site as designated by Avaya. Suspected security vulnerabilities with Avaya products should be reported to Avaya by sending mail to: securityalerts@avaya.com.

Downloading Documentation

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

Contact Avaya Support

See the Avaya Support website: http://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, 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: Introduction

Purpose

The *Avaya Identity Engines Ignition Guest Manager REST APIs* allow developers to integrate Guest Manager with other standalone or web applications to create Guest Users and Devices for a Provisioner in Ignition Guest Manager or to get Guest Users and Devices of a Provisioner from Ignition Guest Manager.

Related resources

Training

Ongoing product training is available. For more information or to register, you can access the Web site at http://avaya-learning.com/.

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 http://support.avaya.com and perform one of the following actions:
  - 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 perform one of the following actions:
  - 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.

Subscribing to e-notifications

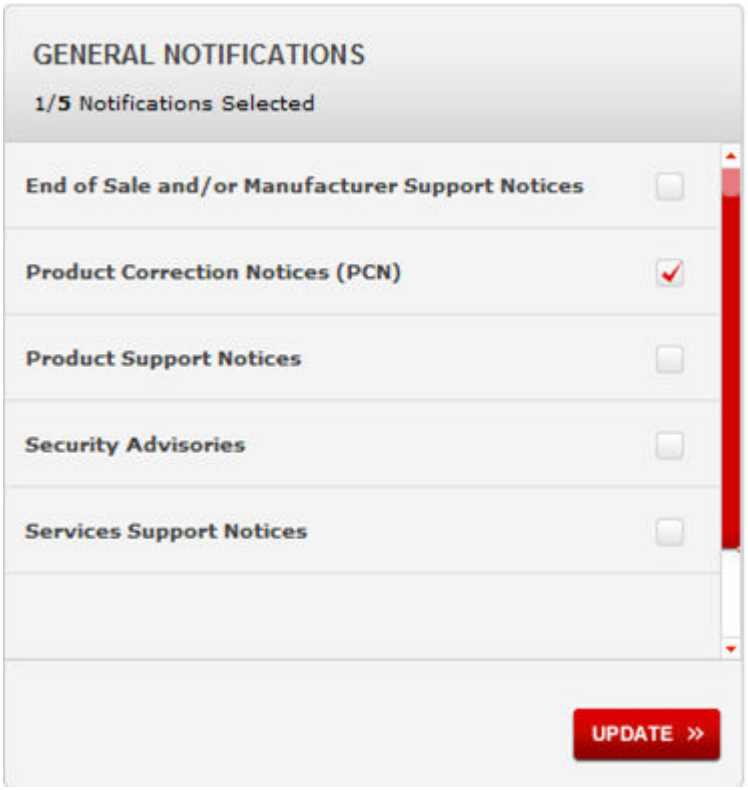
Subscribe to e-notifications to receive an email notification when documents are added to or changed on the Avaya Support website.

About this task

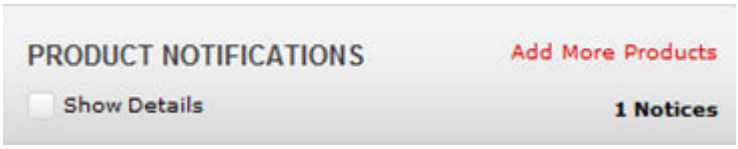
You can subscribe to different types of general notifications, for example, Product Correction Notices (PCN), which apply to any product or a specific product. You can also subscribe to specific types of documentation for a specific product, for example, Application & Technical Notes for Ethernet Routing Switch 8800.

Procedure

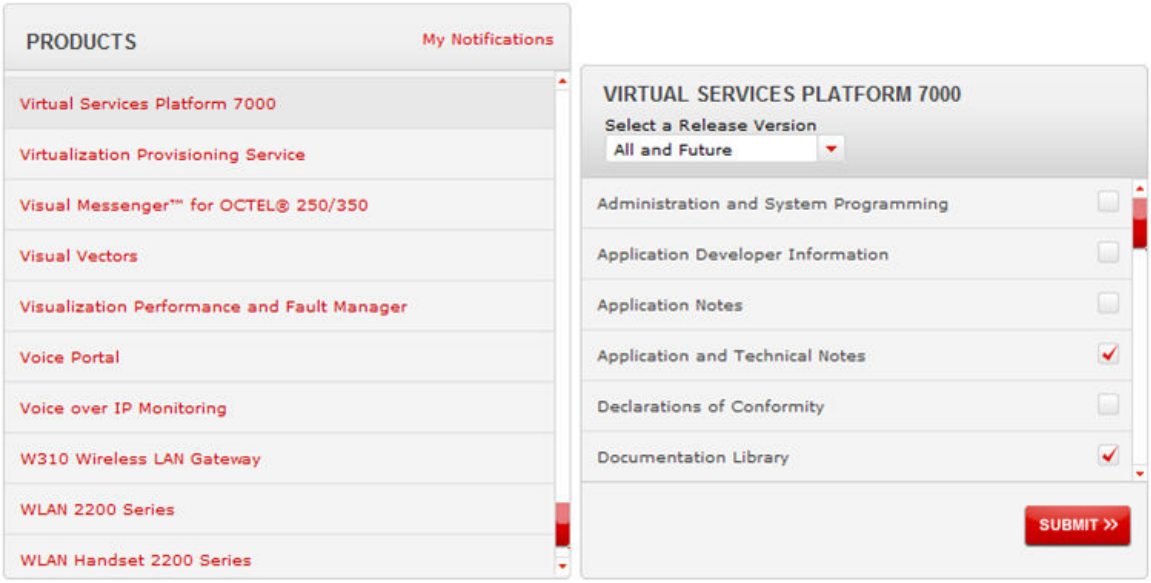
1. In an Internet browser, go to https://support.avaya.com.
2. Type your username and password, and then click **Login**.
3. Under **My Information**, select **SSO login Profile**.
4. Click **E-NOTIFICATIONS**.
5. In the GENERAL NOTIFICATIONS area, select the required documentation types, and then click **UPDATE**.



6. Click **OK**.
7. In the PRODUCT NOTIFICATIONS area, click **Add More Products**.



8. Scroll through the list, and then select the product name.
9. Select a release version.
10. Select the check box next to the required documentation types.



11. Click **Submit**.

Searching a documentation collection

On the Avaya Support website, you can download the documentation library for a specific product and software release to perform searches across an entire document collection. For example, you can perform a single, simultaneous search across the collection to quickly find all occurrences of a particular feature. Use this procedure to perform an index search of your documentation collection.

Before you begin

- Download the documentation collection zip file to your local computer.
- You must have Adobe Acrobat or Adobe Reader installed on your computer.

Procedure

1. Extract the document collection zip file into a folder.
2. Navigate to the folder that contains the extracted files and open the file named <*product_name_release*>.pdx.
3. In the Search dialog box, select the option **In the index named <*product_name_release*>.pdx**.
4. Enter a search word or phrase.
5. Select any of the following to narrow your search:
   - Whole Words Only
   - Case-Sensitive
   - Include Bookmarks

- Include Comments

6. Click **Search**.

   The search results show the number of documents and instances found. You can sort the search results by Relevance Ranking, Date Modified, Filename, or Location. The default is Relevance Ranking.

Support

Go to the Avaya Support website at http://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.

Chapter 2: New in this release

Avaya Identity Engines Ignition Guest Manager REST APIs, NN47280-505 is a new document for IDE Release 9.2, so all the features are new in this release. See *Avaya Identity Engines Release Notes*, NN47280–400 for the complete list of features.

Supported APIs for Release 9.2

Following are the supported APIs for release 9.2:

API	URI	Reference
Fetching Provisioning groups for a Provisioner	/api/provisioningGroups	For more information, see Fetching Provisioning group for a Provisioner on page 26.
Fetching Provisioning Group details for Group name	/api/provisioningGroupDetails/{groupName}	For more information, see Fetching Provisioning Group details for Group name on page 27.
Device Registration	/api/devices	For more information, see Device Registration REST API on page 35.
Fetching Device details by MAC for a Provisioner	/api/devices/deviceDetails/{MAC}	For more information, see Fetching Device details by MAC for a Provisioner on page 38.
Fetching Devices iteratively for a Provisioner		For more information, see Fetching Devices iteratively for a Provisioner on page 40.
GET Cursor Id of device	/api/devices	For more information, see GET Cursor Id on page 40.
GET next N devices	/api/devices/next/{N}/{cursorId}	For more information, see GET next N devices on page 41.
GET first N devices	/api/devices/first/{N}/{cursorId}	For more information, see GET first N devices on page 42.
GET last N devices	/api/devices/last/{N}/{cursorId}	For more information, see GET last N devices on page 43.
GET count of total available device records	/api/devices/count/{cursorId}	For more information, see GET count of total available device records on page 45.

Table continues…

API	URI	Reference
Close Cursor Id	/api/ devices/close/{cursorId}	For more information, see Close Cursor Id on page 45.
Guest User Registration	/api/guestUsers	For more information, see Guest User Registration REST API on page 46.
Re-send Credentials through EMAIL/SMS to Guest User by Username	/api/guestUsers/ resendCredentials/{username}	For more information, see Re-send Credentials through EMAIL/SMS to Guest User by Username on page 52.
Fetching Guest User details by username for a Provisioner	/api/guestUsers/guestUserDetails/ {username}	For more information, see Fetching Guest User details by username for a Provisioner on page 53.
Fetching Guest Users iteratively for a Provisioner		For more information, see Fetching Guest Users iteratively for a Provisioner on page 55.
GET Cursor Id	/api/guestUsers	For more information, see GET Cursor Id on page 55.
GET next N Guest Users	/api/guestUsers/next/{N}/{cursorId}	For more information, see GET next N Guest Users on page 56.
GET first N Guest Users	/api/ guestUsers/first/{N}/{cursorId}	For more information, see GET first N Guest Users on page 57.
GET last N Guest Users	/api/ guestUsers/last/{N}/{cursorId}	For more information, see GET last N Guest Users on page 58.
GET count of total available Guest Users records	/api/ guestUsers/count/{cursorId}	For more information, see GET count of total available Guest User records on page 59.
Close Cursor Id	/api/ guestUsers/close/{cursorId}	For more information, see Close Cursor Id on page 60.

Chapter 3: Guest Manager RESTful web services introduction

RESTful web services are built to work best on the Web. Representational State Transfer (REST) is an architectural style that specifies constraints like uniform interface, if that is applied to a web service, it induces desirable properties such as performance, scalability, and modifiability to enable services to work best on the Web.

API can be accessed using any web development language as the REST Application Programming Interface (API) is based on open standards.

In the REST architectural style, data and functionality are considered as resources. Guest Manager REST APIs provides access to resources using URI paths. To use a REST API, your application makes an HTTP request and parses the response. The Guest Manger REST API uses JSON and XML as its communication format, and the standard HTTP methods like GET, PUT, POST and DELETE.

Chapter 4: Guest Manager REST API Initial Setup

This chapter describes the procedures to create Provisioning groups, Provisioner and to install the RESTClient plugin for Firefox.

Setting up Guest Manager REST API

Follow the below procedures in sequence to enable the Guest Manager REST APIs.

1. Create Provisioning Group in Guest Manager (GM). For more information, see Creating Provisioning Group in Guest Manager on page 14
2. Create Provisioner in GM. For more information, see Creating Provisioner and associating it with Provisioning Groups on page 15
3. Download and Install RESTClient plugin in Firefox. For more information, see Downloading and Installing Firefox RESTClient plugin on page 15

Creating Provisioning Group in Guest Manager

Use the following procedure to create a Provisioning Group in Guest Manager.

Procedure

1. In a support web browser, enter the Guest Manager IP address (https://<*Guest Manager IP address*>/GuestManager/admin).
2. Enter the **User Name** and **Passsword**. The default **User Name** and **Passsword** is `admin` and `admin`.
3. Click **Provisioning Group** to create new groups.

   For more information on how to create Provisioning Groups, see *Configuring Avaya Identity Engines Ignition Guest Manager*, NN47280-501.

Creating Provisioner and associating it with Provisioning Groups

Use the following procedure to create a Provisioner to associate it with the Provisioner Groups in Guest manager.

Procedure

1. Login to Guest Manager using the default credentials.
2. Click **Provisioners** to create a Provisioner and associate Provisioning Groups that are created. For more information on how to create and associate Provisioner to Provisioning Groups, see *Configuring Avaya Identity Engines Ignition Guest Manager*, NN47280-501.

Downloading and Installing Firefox RESTClient plugin

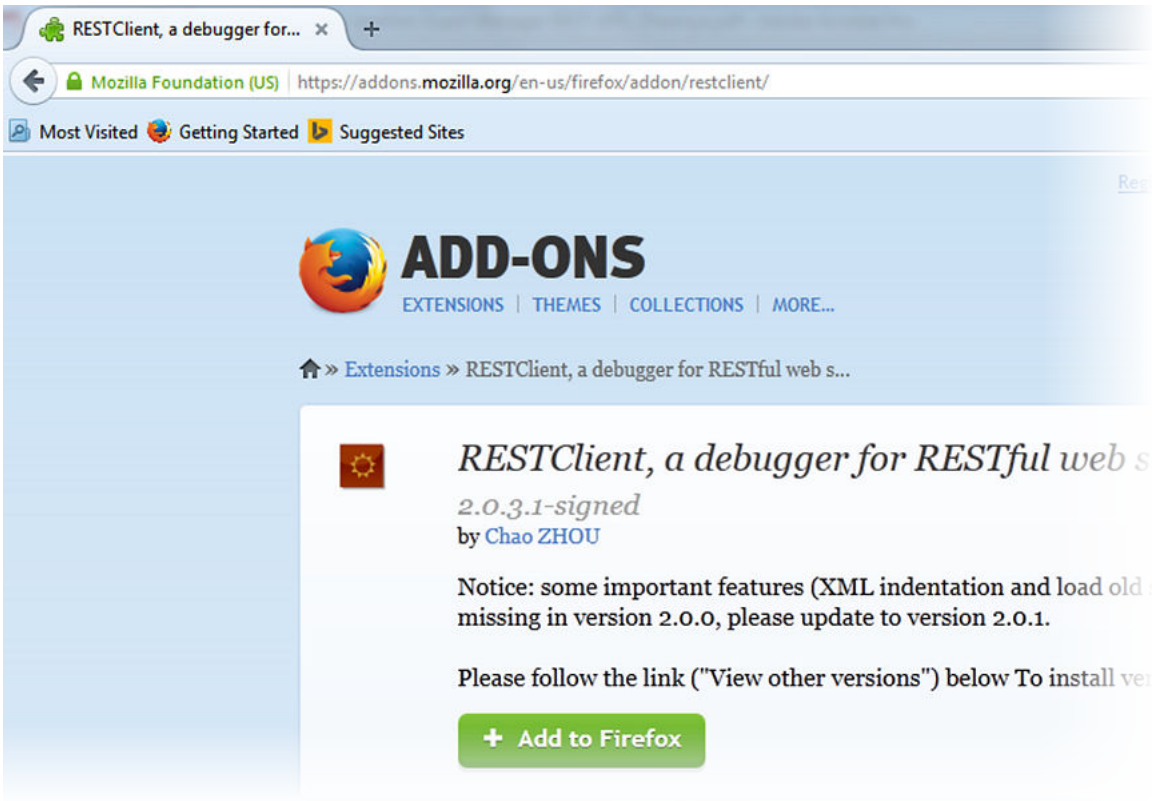
This section describes the procedure to download and install the Firefox RESTClient plugin.

Procedure

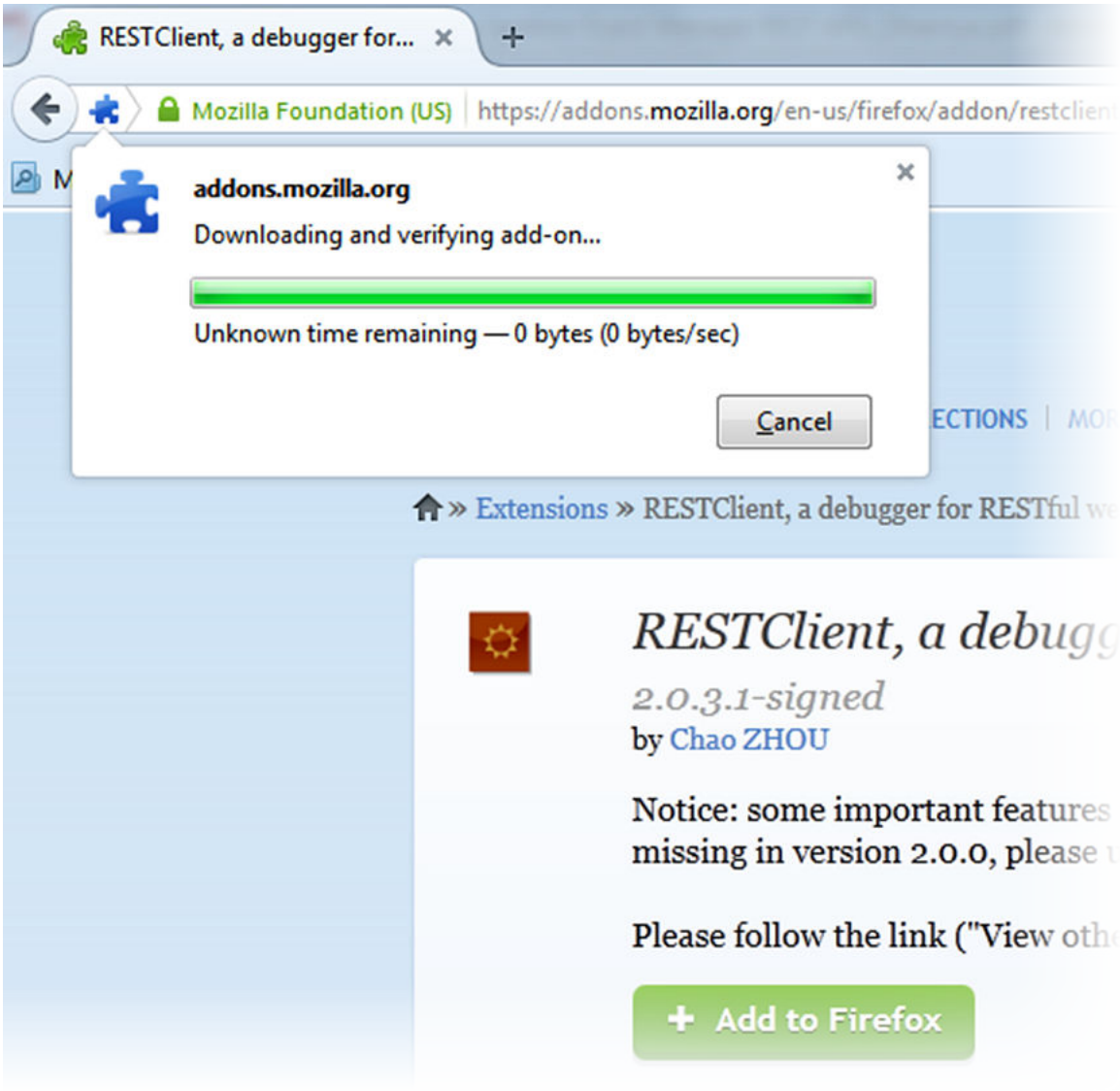
1. Download and Install the Firefox RESTClient plugin from the following URL:

   https://addons.mozilla.org/en-us/firefox/addon/restclient/

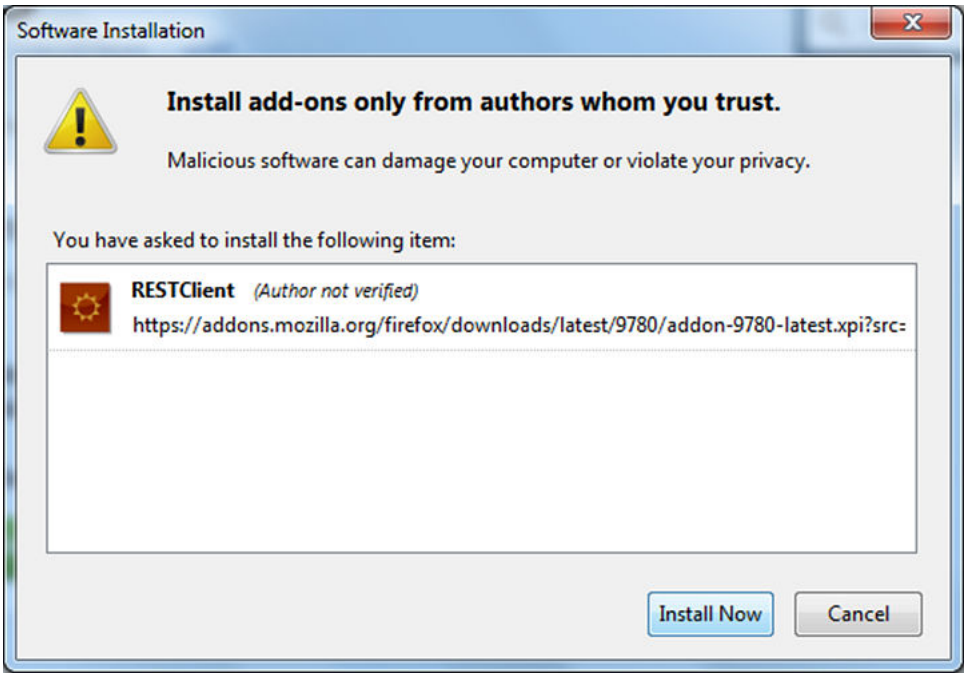
2. Click **+ Add to Firefox**.



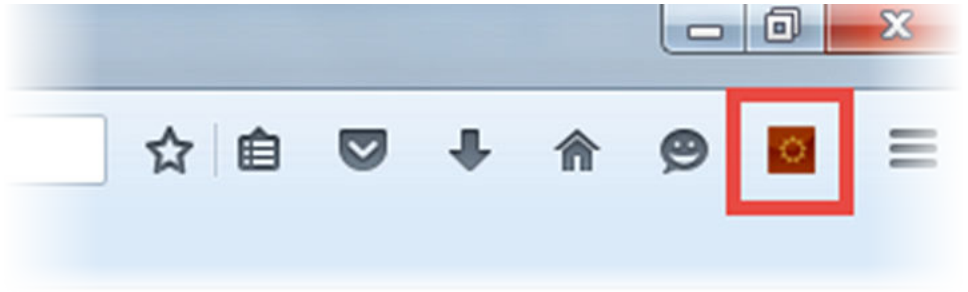
The plugin gets downloaded and verified and Software Installation window appears.


RESTClient, a debugger for...
Mozilla Foundation (US)
https://addons.mozilla.org/en-us/firefox/addon/restclient
addons.mozilla.org
Downloading and verifying add-on...
Unknown time remaining — 0 bytes (0 bytes/sec)
Cancel
» Extensions » RESTClient, a debugger for RESTful
RESTClient, a debugg
2.0.3.1-signed
by Chao ZHOU
Notice: some important features
missing in version 2.0.0, please
Please follow the link ("View oth
+ Add to Firefox

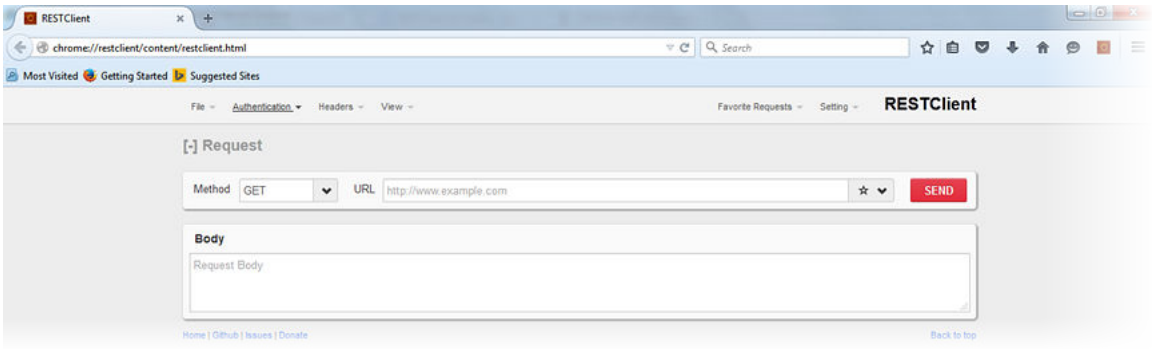
3. Click **Install Now**.



The plugin gets installed and the RESTClient icon appears as shown in the following image.



4. To launch RESTClient plugin, click the RESTClient icon.



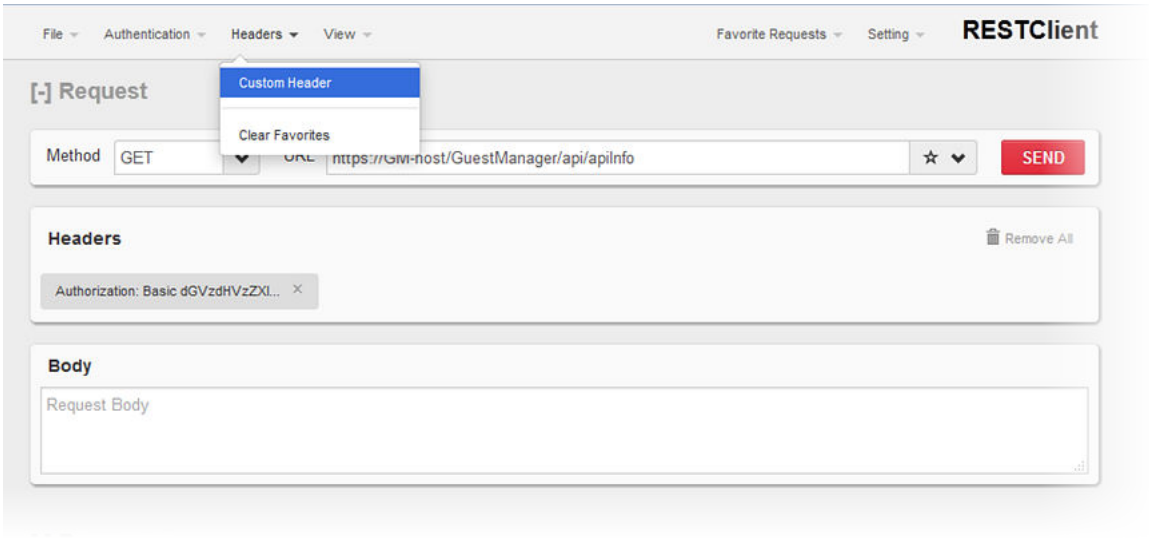
Guest Manager API Version

API versioning is maintained for the client to use the latest REST Web Services for the new features. We use **Request Header** to specify the API version. Current API Version is v1.0 (**api-version:v1.0**).

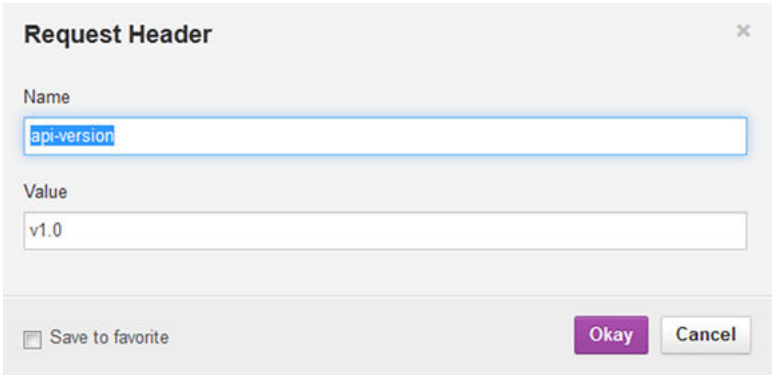
Follow the below procedure to add version in the **HTTP Headers**.

Procedure

1. In the RESTClient, click **Headers** > **Custom Headers**.

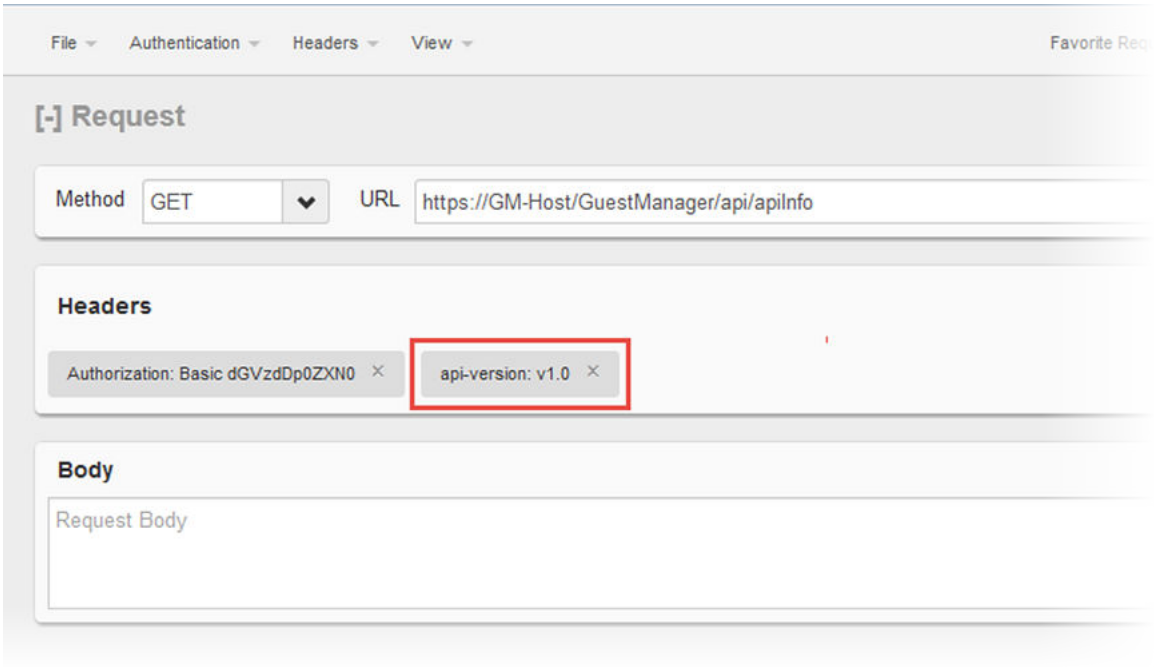


The Request Header window appears.



2. Enter the **Name** and **Value** for the version.
3. Click **Okay**.

   The version gets added to the **Headers**.



Authorization

Authorization HTTP header is required for each API for the Provisioner login credentials. The Provisioner login credentials must be Base64 encrypted with Basic Authorization Scheme.

Authorization Scheme: **Basic (Base64 encryption)**

Authorization: **username:password**

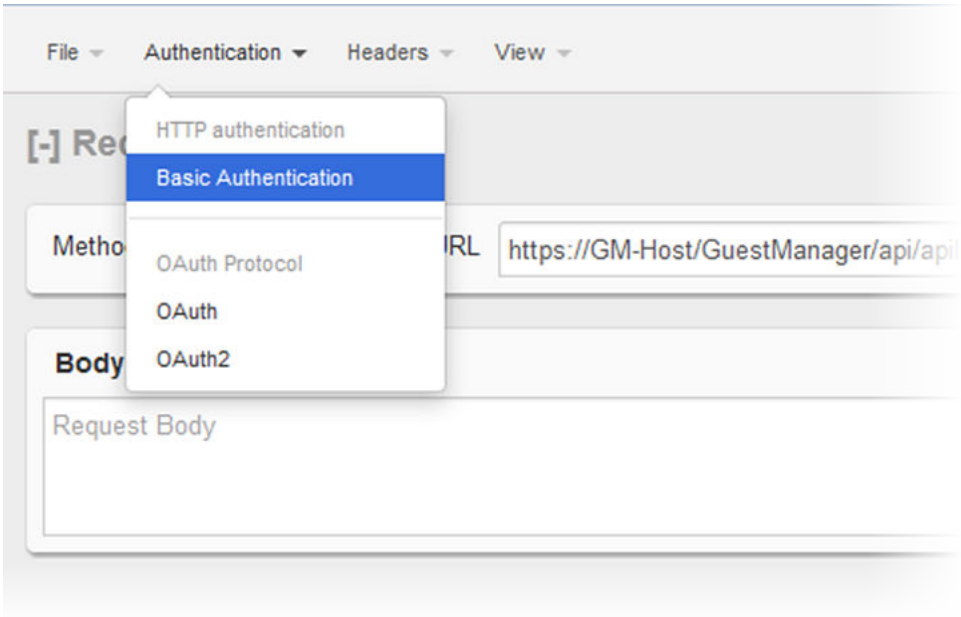


Figure 1: Basic Authentication

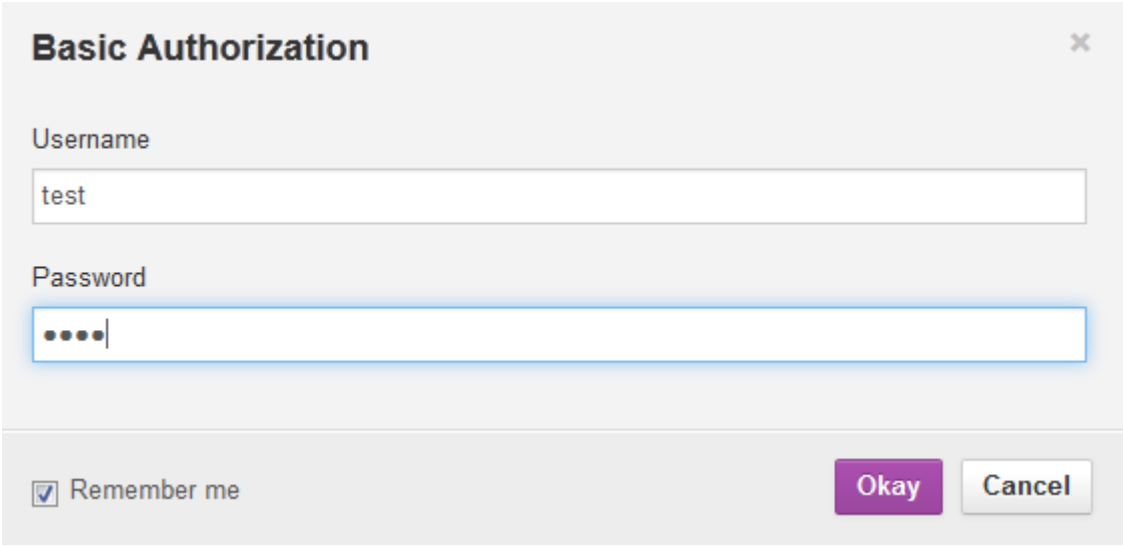


Figure 2: Basic Authorization

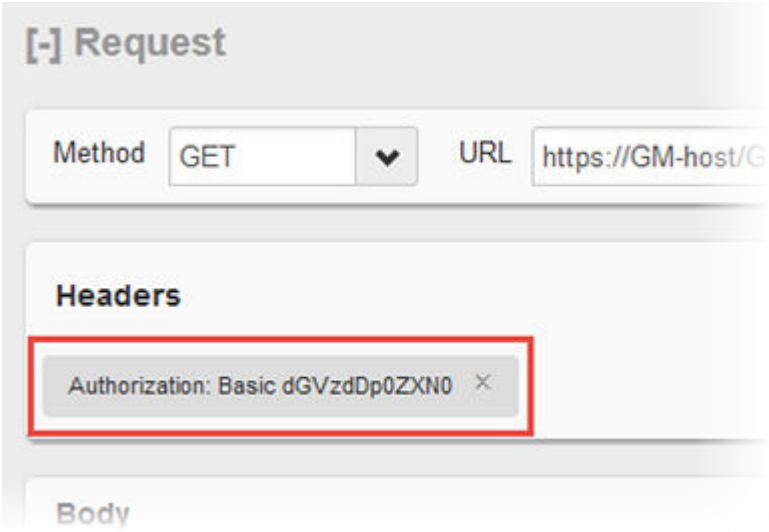


Figure 3: Authorization Header

Guest Manager API Info

Define the REST API to get the API information that contains the basic information about the API.

Note:

Authorization and api-version HTTP header is not required for API info.

Guest Manager API info	
URI	/api/apiInfo
HTTP Header	Accept: application/json
Response	The Format of response preview can be XML or JSON and Avaya focus on JSON primarily. The Response preview contains the following information. • apiPath: The base path used to fetch the API info. • name : Service Name. • vendor: Name of the Vendor. • product Name: Name of the product. • version : API version.

Example	**Request**
	`GET /GuestManager/api/apiInfo HTTP/1.1`
	`Host: 10.120.120.30`
	`Accept:application/json`
	`Cache-Control: no-cache`

Table continues…

Guest Manager API info	

	Response
	
	JSON Format
	
	`{`
	`"apiPath": "/api",`
	`"name": "Ignition Guest Manager REST API",`
	`"productName": "Avaya Identity Engines Ignition Guest Manager",`
	`"vendor": "Avaya Inc.",`
	`"version": "v1.0"`
	`}`

Note:

The Guest Manager APIs URL must be postfix with Guest Manager base URL.

For example: If Guest Manager base URL is https://10.10.10.10/ Guest Manager then API info URI is https://10.10.10.10/ Guest Manager/api/apiInfo.

Common Error Cases

The following table describes the Common Error Cases.

Error Case	Response Code	Error Respone
Authorization Header missing	401 Unauthorized	errorCode: AUTHORIZATION_REQUIRED msg: Authorization required.
Missing version	406 Not Acceptable	erorCode: VERSION_REQUIRED msg: API Version required, refer API doc for details.
Invalid version format	406 Not Acceptable	erorCode: INVALID_VERSION_FORMAT msg: API version is not a valid format, refer API doc for details.
Version not supported	406 Not Acceptable	erorCode: INVALID_VERSION_FORMAT msg: API version is not supported.
Guest Manager is not connected with Ignition Server	500 Internal Server Error	errorCode: IGM_NOT_CONNECTED_WITH_IGS msg: Ignition Guest Manager is not connected to the

Table continues…

Error Case	Response Code	Error Respone
		Ignition&#153 Server. Please contact the Administrator.
Radius configuration missing	500 Internal Server Error	errorCode: RADIUS_CONFIG_MISSING msg: Radius Configuration Missing, Please Contact Administrator.
Guest User or Device group association failure	401 Unauthorized	errorCode: PROVISIONING_ACESS_DENIED msg: Your account does not have permission to Provisioning the Guest User or Devices.
Invalid Credentials	401 Unauthorized	errorCode: INAVLID_CREDENTIALS msg: Invalid user name and Password.
Radius Error (Ignition server not reachable)	503 Service Unavailable	errorCode: RADIUS_ERROR msg: Radius server error <error msg>
Provisioning Group is not accessible/invalid	400 Bad Request	errorCode: PROVISIONING_GROUP_ACCESS_DENIED msg: Your account does not have permission to access the Provisioning Group: {group name}
Does not have permission to provision the device	400 Bad Request	errorCode: DEVICE_PROVISIONING_ACCESS_DENIED msg: You do not have the permission to create the device, Please contact Administrator
Does not have permission to provision the Guest User	400 Bad Request	errorCode: GUEST_USER_PROVISIONING_ACCESS_DENIED msg: You do not have the permission to create the guest user accounts, Please contact Administrator.
Invalid input data	400 Bad Request	errorCode: INVALID_RECORD

Table continues…

Error Case	Response Code	Error Respone
		msg: Invalid Fields: {Comma separated attribute name} Example: Invalid Fields: macAddress, name
Record already exist with same macAddress	400 Bad Request	errorCode: DUPLICATE_DEVICE_RECORD msg: The device you provided already exists. Please provide a different MAC address
Limit of enabled devices reached for the Provisioner	403 Forbidden	errorCode: PROVISIONING_DEVICE_LIMIT_EXCEED msg: Limit on Number of enabled devices has been reached. Delete/ Lock Devices to reach level below limit: {limit}
Error while fetching Provisioner Group	500 Internal Server Error	errorCode: PROVISIONING_GROUP_ERROR msg: Unable to get Provisioner Group. Error: <error msg>

Chapter 5: Guest Manager REST APIs

This chapter describes the GM REST Web Services for third-party, to fetch the list and details of Provisioning groups, Devices and Guest Users for a Provisioner.

Fetching Provisioning group for a Provisioner

This API is used to fetch the list of Provisioning groups that are associated with a Provisioner.

Fetching Provisioning group for a Provisioner	
URI	/api/provisioningGroups
Method	GET

HTTP Headers	`Authorization Scheme: Basic (Base64 encryption)`
	`Authorization: username:password`
	`api-version: {VERSION}`
	`Accept: application/json or application/xml`

Response Code	`200 OK`
Response Payload	`List of Provisioning Group name`

Example	**Request**
	`GET /GuestManager/api/provisioningGroups HTTP/1.1`
	`Host: 10.120.120.30`
	`api-version: v1.0`
	`Authorization: Basic dGVzdDp0ZXN0`
	`Accept:application/json`
	`Cache-Control: no-cache`
	Response Payload
	JSON Format
	`{`
	`"ProvisioningGroups": {`
	`"groupName": [`
	`"api-device-provGroup",`
	`"api-device-provGroup1",`
	`"api-device-provGroup2"`
	`]`

Table continues…

Fetching Provisioning group for a Provisioner

}
}
 	
XML Format
 	
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ProvisioningGroups>
<groupName>api-device-provGroup</groupName>
<groupName>api-device-provGroup1</groupName>
<groupName>api-device-provGroup2</groupName>
</ProvisioningGroups>
 	
Error Response
 	
{
"error": {
"errorCode": "AUTHORIZATION_REQUIRED",
"msg": "Authorization required."
}
}

Fetching Provisioning Group details for Group name

The API is used to query the Provisioning group details for a Provisioning group name.

Fetching Provisioning Group details for Group name	
URI	/api/provisioningGroupDetails/{groupName}
Method	GET
HTTP Headers	Authorization Scheme: Basic (Base64 encryption) Authorization: username:password api-version:{VERSION} Accept: application/json or application/xml
Response Code	200 OK
Response Payload	Provisioning Group details
Example	**Request** GET /GuestManager/api/provisioningGroupDetails/pg-api-user-device HTTP/1.1 Host: 10.120.120.30 api-version: v1.0 Authorization: Basic dGVzdDp0ZXN0 Accept:application/json **Guest User and Device Rights Provisioning** The following example contains Provisioning group details with the Guest Users and devices associated with the group name.

Table continues…

Fetching Provisioning Group details for Group name

JSON Format

```
{
    "ProvisioningGroup": {
        "groupName": "api-device-provGroup",
        "maxDuration": 8,
        "durationUnit": "HOURS",
        "timezone": "Asia/Calcutta",
        "guestUserAllowed": true,
        "devicesAllowed": true,
        "guestUserDetails": {
            "userNameAccessible": false,
            "passwordAccessible": false,
            "firstAndLastNameAccessible": true,
            "firstAndLastNameRequired": true,
            "emailRequired": true,
            "cellPhoneRequired": true,
            "accountValidityDurationAccessible": true,
            "accountActivationAtFirstLogin": false,
            "guestDetailsAccessible": true,
            "guestEmailNotification": true,
            "guestSMSNotification": true,
            "displayUserName": false,
            "displayPassword": false
        },
        "devicesDetails": {
            "nameAccessible": true,
            "nameRequired": false,
            "typeAccessible": true,
            "typeRequired": false,
            "subTypeAccessible": true,
            "subTypeRequired": false
        }
    }
}
```

XML Format

```
<ProvisioningGroup>
    <groupName>api-device-provGroup</groupName>
    <maxDuration>8</maxDuration>
    <durationUnit>HOURS</durationUnit>
    <timezone>Asia/Calcutta</timezone>
    <guestUserAllowed>true</guestUserAllowed>
    <devicesAllowed>true</devicesAllowed>
    <guestUserDetails>
        <userNameAccessible>false</userNameAccessible>
        <passwordAccessible>false</passwordAccessible>
        <firstAndLastNameAccessible>true</firstAndLastNameAccessible>
        <firstAndLastNameRequired>true</firstAndLastNameRequired>
        <emailRequired>true</emailRequired>
        <cellPhoneRequired>true</cellPhoneRequired>
        <accountValidityDurationAccessible>true</
accountValidityDurationAccessible>
        <accountActivationAtFirstLogin>false</
accountActivationAtFirstLogin>
        <guestDetailsAccessible>true</guestDetailsAccessible>
        <guestEmailNotification>true</guestEmailNotification>
        <guestSMSNotification>true</guestSMSNotification>
        <displayUserName>false</displayUserName>
        <displayPassword>false</displayPassword>
    </guestUserDetails>
```

Table continues…

Fetching Provisioning Group details for Group name

```
        <devicesDetails>
            <nameAccessible>true</nameAccessible>
            <nameRequired>false</nameRequired>
            <typeAccessible>true</typeAccessible>
            <typeRequired>false</typeRequired>
            <subTypeAccessible>true</subTypeAccessible>
            <subTypeRequired>false</subTypeRequired>
        </devicesDetails>
    </ProvisioningGroup>
```

Guest User Rights Provisioning Group

The following example contains Provisioning group details with the Guest User rights associated with the group name.

```
{
    "ProvisioningGroup": {
        "groupName": "api-device-provGroup2",
        "maxDuration": 8,
        "durationUnit": "HOURS",
        "timezone": "Asia/Calcutta",
        "guestUserAllowed": true,
        "devicesAllowed": false,
        "guestUserDetails": {
            "userNameAccessible": true,
            "passwordAccessible": false,
            "firstAndLastNameAccessible": true,
            "firstAndLastNameRequired": true,
            "emailRequired": true,
            "cellPhoneRequired": true,
            "accountValidityDurationAccessible": true,
            "accountActivationAtFirstLogin": false,
            "guestDetailsAccessible": true,
            "guestEmailNotification": true,
            "guestSMSNotification": true,
            "displayUserName": true,
            "displayPassword": true
        }
    }
}
```

Device Rights Provisioning Group

The following example contains Provisioning group details with the device rights associated with the group name.

```
{
    "ProvisioningGroup": {
        "groupName": "api-device-provGroup1",
        "maxDuration": 8,
        "durationUnit": "HOURS",
        "timezone": "Asia/Calcutta",
        "guestUserAllowed": false,
        "devicesAllowed": true,
        "devicesDetails": {
            "nameAccessible": true,
            "nameRequired": false,
            "typeAccessible": true,
            "typeRequired": false,
            "subTypeAccessible": true,
            "subTypeRequired": false
        }
```

Table continues…

Fetching Provisioning Group details for Group name
} }

Variable definition

The following table describes the parameters for the Provisioning group, Guest User and Device details.

Provisioning Group

Attribute	Type/Value	Description
groupName	String	Specifies the group name. The maximum length of the group name must be 30 characters and the allowed characters are alphabets (upper and lower case), numbers (0 to 9) and special characters (hyphen and underscore).
maxDuration	Long	Specifies the maximum duration for which the guest user or device account is valid.
durationUnit	String	Specifies the duration in minutes, hours or days. Accepted value input format is [MINUTES:HOURS:DAYS]
timezone	String	Specifies the time zone. For example, Indian Standard Time (Asia/Calcutta).
guestUserAllowed	Boolean	Specifies whether the Provisioner can create Guest User. Parameters and values are: • if guestUserAllowed is true, then allow the Provisioner to create Guest User. • if guestUserAllowed is false, then does not allow the Provisioner to create Guest User.

Table continues…

Attribute	Type/Value	Description
devicesAllowed	Boolean	Specifies whether the Provisioner can create devices. Parameters and values are: • if devicesAllowed is true, then allow the Provisioner to create devices. • if devicesAllowed false, then does not allow the Provisioner to create devices.
guestUserDetails	Object	For more information, see Guest User details table below.
deviceDetails	Object	For more information, see Device details table below.

Guest User Details

Attribute	Type/Value	Description
userNameAccessible	Boolean	Specifies if the user name is required. Parameters and values are: • if userNameAccessible is true, then User Name value is used. • if userNameAccessible is false, then User Name is optional and value is ignored.
passwordAccessible	Boolean	Specifies if password is required. Parameters and values are: • if passwordAccessible is true, then password value is used. • if passwordAccessible is false, then optional and value is ignored.
firstAndLastNameAccessible	Boolean	Specifies whether the Provisioner can set the first and last name. Parameters and values are: • if firstAndLastNameAccessible is true, then Provisioner can set the first and last name. • if firstAndLastNameAccessible is false - Provisioner cannot set the first and last name.

Table continues…

Attribute	Type/Value	Description
firstAndLastNameRequired	Boolean	Specifies whether the first and last name is required. Parameters and values are: • if firstAndLastNameRequired is true, then first and last name are required. • if firstAndLastNameRequired is false, then first and last name are optional.
emailRequired	Boolean	Specifies whether the email address is mandatory. Parameters and values are: • if emailRequired is true, then email is mandatory. • if emailRequired is false, then email is optional.
cellPhoneRequired	Boolean	Specifies whether cell phone number is mandatory. Parameters and values are: • if cellPhoneRequired is true, then mobile number is mandatory. • if cellPhoneRequired is false, then mobile number is optional.
accountValidityDurationAccessible	Boolean	Specifies whether the Provisioner can change the duration. The duration cannot be more than the Max duration. Parameters and values are: • if accountValidityDurationAccessible is true, then Provisioner can change the account validity duration. • if accountValidityDurationAccessible is false, then Provisioner cannot change the account validity duration.

Table continues…

Attribute	Type/Value	Description
accountActivationAtFirstLogin	Boolean	Specifies the account activation. Parameters and values are: • if accountActivationAtFirstLogin is true, then Guest User account gets activated on first login and start time is calculated from the first time login. • if accountActivationAtFirstLogin is false, then Guest User account gets activated on the start time.
guestDetailsAccessible	Boolean	Specifies whether to allow the Provisioner to set the Guest details. Parameters and values are: • if guestDetailsAccessible is true, then Provisioner can set the Guest details. • if guestDetailsAccessible is false, then Provisioner cannot set the Guest details.
guestEmailNotification	Boolean	Specifies whether an email notification must be sent to the Guest. Parameters and values are: • if guestEmailNotification is true, then guest receives the email notification. • if guestEmailNotification is false, then guest does not receive email notification.
guestSMSNotification	Boolean	Specifies whether SMS notification must be sent to the Guest. Parameters and values are: • if guestSMSNotification is true, then guest receives the notification through SMS. • if guestSMSNotification is false, then guest does not receive notification through SMS.

Table continues…

Attribute	Type/Value	Description
displayUserName	Boolean	Specifies whether the user name must be sent in the response. Parameters and values are: • if displayUserName is true, then user name is sent. • if displayUserName is false, then user name is not sent.
displayPassword	Boolean	Specifies whether the password must be sent in the response. Parameters and values are: • if displayPassword is true, then password is sent. • if displayPassword is false, then password is not sent.

Device Details

Attributes	Type/Value	Description
nameAccessible	Boolean	Specifies whether to allow the Provisioner to configure the device name. Parameters and values are: • if nameAccessible is true, then Provisioner can configure device name. • if nameAccessible is false, then Provisioner cannot configure device name.
nameRequired	Boolean	Specifies whether the device name is mandatory. Parameters and values are: • if nameRequired is true, then device name is mandatory. • if nameRequired is false, then device name is optional.
typeAccessible	Boolean	Specifies whether the Provisioner can configure the device type. Parameters and values are: • if typeAccessible is true, then Provisioner can configure device type.

Table continues…

Attributes	Type/Value	Description
		• if typeAccessible is false, then Provisioner cannot configure device type.
typeRequired	Boolean	Specifies whether the device type is required. Parameters and values are: • if typeRequired is true, then device type is mandatory. • if typeRequired is false, then device type is optional.
subTypeAccessible	Boolean	Specifies whether the Provisioner can configure the device Sub Type. Parameters and values are: • if subTypeAccessible is true, then Provisioner can configure the device Sub Type. • if subTypeAccessible is false, then Provisioner cannot configure the device Sub Type.
subTypeRequired	Boolean	Specifies whether the device Sub Type is required. Parameters and values are: • if subTypeRequired is true, then device Sub Type is mandatory. • if subTypeRequired is false, then device Sub Type is optional.

Device Registration REST API

The API allows Provisioner to add devices to the Guest Manager.

Device Registration REST API	
URI	/api/devices
Method	POST

HTTP Headers	`Authorization Scheme: Basic (Base64 encryption)`
	`Authorization: username:password`
	`api-version:{VERSION}`
	`Accept: application/json or application/xml`

Table continues…

Device Registration REST API	
	Content-Type: application/json or application/xml
Response Code	201 (created)
Response Payload	N/A

Example	**Request**
	POST /GuestManager/api/devices HTTP/1.1
	Host: 10.120.120.30
	api-version: v1.0
	Content-Type: application/json
	Authorization: Basic dGVzdDp0ZXN0
	Accept:application/json
	Cache-Control: no-cache
	{ "Device": { "provisiongGroupName": "api-device-provGroup",
	"macAddress": "10:10:10:00:00:01", "name": "device1", "type":
	"mobile", "subType": "generic-android" } }
	Response Header
	Content-Length: 0
	Date: Thu, 25 Jun 2015 07:27:46 GMT
	Location: http://10.120.120.30/GuestManager/api/devices/deviceDetails/
	10:10:10:00:00:01
	Server: Apache-Coyote/1.1
	Request Payload
	JSON Format
	{
	"Device": {
	"provisioningGroupName": "api-device-provGroup",
	"macAddress": "10:10:10:00:00:01",
	"name": "device1",
	"type": "mobile",
	"subType": "generic-android"
	}
	}
	XML Format
	<Device>
	<provisioningGroupName>api-device-provGroup</provisioningGroupName>
	<macAddress>10:10:10:00:00:01</macAddress>
	<name>device1</name>
	<type>mobile</type>
	<subType>generic-android</subType>
	</Device>
	Error Response
	{
	"error": {
	"errorCode": "AUTHORIZATION_REQUIRED",
	"msg": "Authorization required."
	}
	}

Variable definition

The following table describes the parameters for the Device registration REST API.

Attribute	Type/Value	Description
provisioningGroupName	String	Specifies the provisioning group name. The maximum length of the group name must be 30 characters and the allowed characters are alphabets (upper and lower case), numbers (0 to 9) and special characters (hyphen and underscore).
macAddress	String	Specifies MAC Address of the device. The format is xx:xx:xx:xx:xx:xx. For example, oa:00:01:ab:a0:10
name	String	Specifies the device name. The device name depends on the Provisioning Group settings. Parameters and values are: • if nameAccessible is true, then name value is used, otherwise ignored. • if nameRequired is true, then name is mandatory, otherwise optional. The maximum length of the name must be 40 characters and the allowed characters are alphabets (upper and lower case), numbers (0 to 9), and special characters (hyphen, underscore and space).
type	String	Specifies the device type. It must match exactly with the defined device types in Ignition Dashboard. The device type depends on the Provisioning Group settings. Parameters and values are: • if typeAccessible is true, then type value is used, otherwise ignored.

Table continues…

Attribute	Type/Value	Description
		• if typeRequired is true, then type is mandatory, otherwise optional.
subType	String	Specifies the device Sub Type. It must match exactly with the defined device Sub Type in Ignition Dashboard. The device Sub Type depends on the Provisioning Group settings. Parameters and values are: • if subTypeAccessible is true, then Sub Type value is used, otherwise ignored. • if subTypeRequired is true, then Sub Type is mandatory, otherwise optional.

Fetching Device details by MAC for a Provisioner

The API is used to query the Device details by MAC for a Provisioner.

Fetching Device details by MAC for a Provisioner	
URI	/api/devices/deviceDetails/{MAC}
Method	GET

HTTP Headers	`Authorization Scheme: Basic (Base64 encryption)`
	`Authorization: username:password`
	`api-version:{VERSION}`
	`Accept: application/json or application/xml`

Response Code	`200 OK`
Response Payload	`Device Details`

Example	**Request**
	`GET /GuestManager/api/devices/deviceDetails/10:10:10:00:00:02 HTTP/1.1`
	`Host: 10.120.120.30`
	`api-version: v1.0`
	`Authorization: Basic dGVzdDp0ZXN0`
	`Accept:application/json`
	`Cache-Control: no-cache`

Table continues…

Fetching Device details by MAC for a Provisioner

Response Payload

```
{
    "Device": {
        "macAddress": "10:10:10:00:00:02",
        "name": "device1",
        "type": "mobile",
        "subType": "generic-android",
        "source": "GM-p-api-user-device",
        "enabled": true,
        "assetType": "PERMANENT",
        "startTime": "2015/06/24 07:13:53 PM IST",
        "endTime": "-",
        "provisioningGroup": "api-device-provGroup",
        "provisioner": "Internal/pall"
    }
}
```

Variable definition

The following table describes the parameters for Device record details.

Attribute	Type/Value	Description
macAddress	String	Specifies the MAC address. The format is xx:xx:xx:xx:xx:xx
type	String	Specifies the device type.
subtype	String	Specifies the device Sub Type.
source	String	Specifies the device source.
enabled	Boolean	Specifies the device record status. Parameter and values are: • if true, then device record is active. • if false, then device record is inactive.
assetType	String	Specifies the asset type. The device record type can be PERMANENT or TEMPORARY.
startTime	String	Specifies the device start time. If the assetType is PERMANENT, then value is '-' The format is yyyy/MM/dd hh:mm:ss a z For example, 2015/06/06 11:10:00 AM IST.

Table continues…

Attribute	Type/Value	Description
endTime	String	Specifies the device end time. If the assetType is PERMANENT or end time is not enable (first login), then value is '-' The format is yyyy/MM/dd hh:mm:ss a z For example, 2015/06/06 18:10:00 PM IST.
provisioningGroup	String	Specifies the provisioning group.
provisioner	String	Specifies the user name of the Provisioner who registered the device.

Fetching Devices iteratively for a Provisioner

Follow the below procedure in sequence to fetch devices iteratively for a Provisioner.

1. Get Cursor Id. For more information, see GET Cursor Id on page 40.
2. Get next N devices. For more information, see GET next N devices on page 41.
3. Get first N devices. For more information, see GET first N devices on page 42.
4. Get last N devices. For more information, see GET last N devices on page 43.
5. Get count of total available device records. For more information, see GET count of total available device records on page 45.
6. Close Cursor Id. For more information, see Close Cursor Id on page 45.

GET Cursor Id

The GET Cursor Id is the first API call to get the paging info that contains cursorId, which is used for subsequent calls, to get the devices iteratively.

GET Cursor Id	
URI	/api/devices
Method	GET

HTTP Headers	`Authorization Scheme: Basic (Base64 encryption)`
	`Authorization: username:password`
	`api-version:{VERSION}`

Table continues…

GET Cursor Id	
	`Accept: application/json or application/xml`
Response Code	`200 OK`
Response Payload	`PagingInfo which contains cursorId and total device records`

Example	**Request**
	`GET /GuestManager/api/devices HTTP/1.1`
	`Host: 10.120.120.30`
	`api-version: v1.0`
	`Authorization: Basic dGVzdDp0ZXN0`
	`Accept:application/json`
	`Cache-Control: no-cache`
	Response Payload
	`{`
	`PagingInfo:{`
	`"cursorId": "12804370433607408411",`
	`"totalRecord": 4`
	`}`
	`}`

Variable definition

The following table describes the parameters of Paging Info.

Attributes	Description
cursorId	Unique number that is maintained in server to get the devices iteratively, for all subsequent request this cursorId should be sent as part of request.
N	Total number of available device records.

GET next N devices

The GET next N devices is the API call to get the next set of N devices.

GET next N devices	
URI	/api/devices/next/{N}/{cursorId}
Method	GET

HTTP Headers	`Authorization Scheme: Basic (Base64 encryption)`
	`Authorization: username:password`
	`api-version:{VERSION}`
	`Accept: application/json or application/xml`

Response Code	`200 OK`

Table continues…

GET next N devices	
Response Payload	`Device List`

Example	**Request**
	`GET /GuestManager/api/devices/next/2/12804370433607408411 HTTP/1.1`
	`Host: 10.120.120.30`
	`api-version: v1.0`
	`Authorization: Basic dGVzdDp0ZXN0`
	`Accept:application/json`
	`Cache-Control: no-cache`
	Response Payload
	`{`
	`"DeviceList": {`
	`"Device": [`
	`{`
	`"macAddress": "10:0b:01:20:00:06",`
	`"name": "device-android",`
	`"type": "mobile",`
	`"subType": "generic-android",`
	`"source": "GM-p-api-user-device",`
	`"enabled": true,`
	`"assetType": "PERMANENT",`
	`"startTime": "2015/06/17 04:47:21 PM IST",`
	`"endTime": "-",`
	`"provisioningGroup": "api-device-provGroup",`
	`"provisioner": "Internal/pall"`
	`},`
	`{`
	`"macAddress": "10:10:10:00:00:02",`
	`"name": "device1",`
	`"type": "mobile",`
	`"subType": "generic-android",`
	`"source": "GM-p-api-user-device",`
	`"enabled": true,`
	`"assetType": "PERMANENT",`
	`"startTime": "2015/06/24 07:13:53 PM IST",`
	`"endTime": "-",`
	`"provisioningGroup": "api-device-provGroup",`
	`"provisioner": "Internal/pall"`
	`}`
	`]`
	`}`
	`}`

GET first N devices

The GET first N devices is the API call to get the first N devices.

GET first N devices	
URI	/api/devices/first/{N}/{cursorId}
Method	GET
HTTP Headers	`Authorization Scheme: Basic (Base64 encryption)`

Table continues…

GET first N devices	
	Authorization: username:password api-version:{VERSION} Accept: application/json or application/xml
Response Code	200 OK
Response Payload	Device List

Example	**Request**
	
	GET /GuestManager/api/devices/first/2/12804370433607408411 HTTP/1.1
	Host: 10.120.120.30
	api-version: v1.0
	Authorization: Basic dGVzdDp0ZXN0
	Accept:application/json
	Cache-Control: no-cache
	
	Response Payload
	
	{
	"DeviceList": {
	"Device": [
	{
	"macAddress": "10:0b:01:20:00:06",
	"name": "device-android",
	"type": "mobile",
	"subType": "generic-android",
	"source": "GM-p-api-user-device",
	"enabled": true,
	"assetType": "PERMANENT",
	"startTime": "2015/06/17 04:47:21 PM IST",
	"endTime": "-",
	"provisioningGroup": "api-device-provGroup",
	"provisioner": "Internal/pall"
	},
	{
	"macAddress": "10:10:10:00:00:02",
	"name": "device1",
	"type": "mobile",
	"subType": "generic-android",
	"source": "GM-p-api-user-device",
	"enabled": true,
	"assetType": "PERMANENT",
	"startTime": "2015/06/24 07:13:53 PM IST",
	"endTime": "-",
	"provisioningGroup": "api-device-provGroup",
	"provisioner": "Internal/pall"
	}
	]
	}
	}

GET last N devices

The GET last N devices is the API call to get the last N devices.

GET last N devices	
URI	/api/devices/last/{N}/{cursorId}
Method	GET
HTTP Headers	Authorization Scheme: Basic (Base64 encryption) Authorization: username:password api-version:{VERSION} Accept: application/json or application/xml
Response Code	200 OK
Response Payload	Device List

Example	**Request**
	GET /GuestManager/api/devices/last/2/12804370433607408411 HTTP/1.1
	Host: 10.120.120.30
	api-version: v1.0
	Authorization: Basic dGVzdDp0ZXN0
	Accept:application/json
	Cache-Control: no-cache
	Response Payload
	{
	"DeviceList": {
	"Device": [
	{
	"macAddress": "10:10:10:00:00:01",
	"name": "device1",
	"type": "mobile",
	"subType": "generic-android",
	"source": "GM-p-api-user-device",
	"enabled": true,
	"assetType": "PERMANENT",
	"startTime": "2015/06/25 12:52:13 PM IST",
	"endTime": "-",
	"provisioningGroup": "api-device-provGroup",
	"provisioner": "Internal/pall"
	},
	{
	"macAddress": "10:10:10:00:00:03",
	"name": "device1",
	"type": "mobile",
	"subType": "generic-android",
	"source": "GM-p-api-user-device",
	"enabled": true,
	"assetType": "PERMANENT",
	"startTime": "2015/06/24 07:46:11 PM IST",
	"endTime": "-",
	"provisioningGroup": "api-device-provGroup",
	"provisioner": "Internal/pall"
	}
	]
	}
	}

GET count of total available device records

The GET count of total available device records is the API call to get the count of device records of a Provisioner.

GET count of total available device records	
URI	/api/devices/count/{cursorId}
Method	GET

HTTP Headers	`Authorization Scheme: Basic (Base64 encryption)`
	`Authorization: username:password`
	`api-version:{VERSION}`
	`Accept: application/json or application/xml`

Response Code	`200 OK`
Response Payload	`Count of Device Records`

Example	**Request**
	`GET /GuestManager/api/devices/count/12804370433607408411 HTTP/1.1`
	`Host: 10.120.120.30`
	`api-version: v1.0`
	`Authorization: Basic dGVzdDp0ZXN0`
	`Accept:application/json`
	`Cache-Control: no-cache`
	Response Payload
	4

Close Cursor Id

The Close Cursor Id API is used to clean up cache for this cursor id and subsequent requests in the server. After cleaning up, the Cursor Id will not be valid.

Close Cursor Id	
URI	/api/ devices/close/{cursorId}
Method	GET

HTTP Headers	`Authorization Scheme: Basic (Base64 encryption)`
	`Authorization: username:password`
	`api-version:{VERSION}`
	`Accept: application/json or application/xml`

Response Code	`204 No Content OK`

Table continues…

Close Cursor Id	
Response Payload	`Device List`

Example	**Request**
	`GET /GuestManager/api/devices/close/12804370433607408411 HTTP/1.1`
	`Host: 10.120.120.30`
	`api-version: v1.0`
	`Authorization: Basic dGVzdDp0ZXN0`
	`Accept:application/json`
	`Cache-Control: no-cache`

Error Cases

The following table describes the Error cases for fetching devices iteratively for a Provisioner.

Error Case	Response Code	Error Response
Invalid cursor id	400 Bad Request	errorCode: INVALID_CURSOR_ID msg: Cursor Id is invalid or expired.
Invalid page size	400 Bad Request	errorCode: INVALID_PAGE_SIZE msg: Invalid page size. Please specify a value between 1 to 500.
No record found	204 No content	No content

Guest User Registration REST API

The API allows the Provisioner to add the Guest User to the Guest Manager.

Guest User Registration REST API	
URI	/api/guestUsers
Method	POST

HTTP Headers	`Authorization Scheme: Basic (Base64 encryption)`
	`Authorization: username:password`
	`api-version:{VERSION}`
	`Accept: application/json or application/xml`
	`Content-Type: application/json or application/xml`

Response Code	`201 Created`

Table continues…

Guest User Registration REST API	
Response Payload	Guest User Details
Example	**Request** `POST /GuestManager/api/guestUsers HTTP/1.1` `Host: 10.120.120.30` `api-version: v1.0` `Content-Type: application/json` `Authorization: Basic dGVzdDp0ZXN0` `Accept:application/json` `Cache-Control: no-cache` `{ "GuestUser" : { "provisioningGroupName" : "pg-api-user", "firstName" : "fName1", "lastName" : "lName1", "userName" : "guestUser1", "password" : "Abc@12", "email" : "test@avaya.com", "cellPhone" : "2991199112", "phoneCarrier" : "T-Mobile", "guestDetails" : "guest Details-DL", "startDate" : "2015/06/25 16:16:41", "durationUnit" : "HOURS", "duration" : 5 } }` **Request Payload** **JSON Format** `{` `"GuestUser" : {` `"provisioningGroupName" : "pg-api-user",` `"firstName" : "fName1",` `"lastName" : "lName1",` `"userName" : "guestUser1",` `"password" : "Abc@12",` `"email" : "test@avaya.com",` `"cellPhone" : "2991199112",` `"phoneCarrier" : "T-Mobile",` `"guestDetails" : "guest Details-DL",` `"startDate" : "2015/06/25 16:16:41",` `"durationUnit" : "HOURS",` `"duration" : 5` `}` `}` **XML Format** `<?xml version="1.0" encoding="UTF-8"?>` `<GuestUser>` `<provisioningGroupName>pg-api-user</provisioningGroupName>` `<userName>guestUser1</userName>` `<firstName>fName1</firstName>` `<lastName>lName1</lastName>` `<email>test@avaya.com</email>` `<password>Abc@12</password>` `<cellPhone>2991199112</cellPhone>` `<phoneCarrier>T-Mobile</phoneCarrier>` `<guestDetails>guest Details-DL</guestDetails>` `<startDate>2015/06/25 16:16:41</startDate>` `<durationUnit>HOURS</durationUnit>` `<duration>5</duration>` `</GuestUser>` **Response Header** `Connection → close` `Content-Type → application/json`

Table continues…

Guest User Registration REST API

```
Date → Thu, 25 Jun 2015 07:10:48 GMT
Location → https://10.120.120.30/GuestManager/api/guestUsers/
guestUserDetails/guestUser1
Transfer-Encoding → chunked
```

Response Payload

```
{
    "GuestUser": {
        "userName": "guestUser1",
        "password": "Abc@12",
        "email": "test@avaya.com",
        "smsAddress": "2991199112@tmomail.net"
}
```

Error Response

```
{
    "error": {
        "errorCode": "AUTHORIZATION_REQUIRED",
        "msg": "Authorization required."
    }
}
```

Variable definition

The following table describes the parameters of Request Guest User and Response Guest User.

Request Guest User

Attribute	Type/Value	Description
provisioningGroupName	String	Specifies the provisioning group name.
userName	String	Specifies the guest account user name. The maximum length of the user name can be 30 characters. The allowed characters are alphabets (upper and lower case), numbers (0 to 9), and special characters (hyphen and underscore). The guest account user name configuration depends on the following Provisioning Group settings. Parameter and value is: If the parameter values for userNameAccessible is true, then mandatory otherwise it is optional and value ignored.

Table continues…

Attribute	Type/Value	Description
firstName	String	Specifies first name of the guest user. The maximum length of the user name can be 30 characters. The allowed characters are alphabets (upper and lower case), numbers (0 to 9), and special characters (hyphen, underscore and space). The guest user first name configuration depends on the following Provisioning Group settings. Parameters and values are: • If firstAndLastNameAccessible is true, then first name value is used, otherwise it is ignored. • If firstAndLastNameRequired is true, then first name is mandatory, otherwise it is optional.
lastName	String	Specifies last name of the guest user. The maximum length of the user name can be 30 characters. The allowed characters are alphabets (upper and lower case), numbers (0 to 9), and special characters (hyphen, underscore and space). The guest user last name configuration depends on the following Provisioning Group settings. Parameters and values are: • If firstAndLastNameAccessible is true, then last name value is used, otherwise it is ignored. • If firstAndLastNameRequired is true, then last name is mandatory, otherwise it is optional.
Email	String	Specifies valid email address of the guest user. The guest user email address configuration depends on the

Table continues…

Attribute	Type/Value	Description
		following Provisioning Group settings. Parameter and value is: • If emailRequired is true, then email is mandatory, otherwise it is optional.
password	String	Specifies the password of the guest user. It is Base64 encrypted. The guest password configuration depends on the following Provisioning Group settings. Parameter and value is: • If passwordAccessible is true, then mandatory, otherwise it is optional and value ignored. **Note:** Password must follow password complexity which defined in Provisioning Group Setting.
cellphone	String (Valid Cell Phone)	Specifies the valid cellphone of the guest user. The maximum length of the user name is 12 digits. The guest cellphone configuration depends on the following Provisioning Group settings. Parameter and value is: If cellPhoneRequired is true, then cellphone is mandatory, otherwise it is optional.
phoneCarrier	String (Carrier Name)	Specifies valid carrier name, valid only if it is registered in SMS Gateways under Administrator notification setting in Guest Manager. Required if cellphone field is not empty and no default Gateway is configured in Guest Manager. If phoneCarrier is Empty and default Gateway is configured, then default Gateway is used.
guestDetails	String	Specifies the guest details and this field is optional.

Table continues…

Attribute	Type/Value	Description
		The maximum length is 48 characters.
startDate	String	Specifies start date. The format is yyyy/MM/dd HH:mm:ss (24 hours format). The start date configuration depends on the following Provisioning Group settings. Parameter and value is: • if firstLoginActivation is false, then it is ignored.
durationUnit	String	Specifies the duration in hours, minutes and days. Accepted value input format is [HOURS:MINUTES:DAYS]
duration	Long	Specifies the duration value. This is optional and must not be more than Provision Group Max duration. The duration configuration depends on the following Provisioning Group settings. Parameter and value is: • If accountValidityDurationAccessible is true, then duration and durationUnit value is used, otherwise it is ignored.

Response Guest User

Attribute	Type/Value	Description
userName	String	Specifies the user name of guest user account. If displayUserName is true, then username value appears, otherwise “-” appears.
password	String	Specifies the password of guest user account. If displayPassword is true, then password value appears, otherwise “-” appears.

Table continues…

Attribute	Type/Value	Description
Email	String	Specifies email address of the guest user account.
smsAddress	String	Specifies SMS Address of the guest user account. The format is cellphone@gateway. For example, 2991199112@tmomail.net.

Re-send Credentials through EMAIL/SMS to Guest User by Username

The API enables the Guest Manager to send EMAIL/SMS notification of login credential to the Guest User. The options EMAIL/SMS is enabled by checking the check box in Provisioning Group. NOTIFICATION_ERROR is sent if the EMAIL/SMS notifications are not enabled.

Re-send Credentials through EMAIL/SMS to Guest User by Username	
URI	/api/guestUsers/resendCredentials/{username} **Note**: username is Guest User name.
Method	GET

HTTP Headers	`Authorization Scheme: Basic (Base64 encryption)`
	`Authorization: username:password`
	`api-version:{VERSION}`
	`Accept: application/json or application/xml`

Response Code	`200 OK`
Response Payload	`Notification Sent Successfully`

Example	**Request**
	`GET /GuestManager/api/ guestUsers/resendCredentials/guestUser1 HTTP/1.1`
	`Host: 10.120.120.30`
	`api-version: v1.0`
	`Authorization: Basic dGVzdDp0ZXN0`
	`Accept:application/json`
	`Cache-Control: no-cache`

Error Cases

Error Case	Response Code	Error Response
Guest User does not exists	404 not found	

Table continues…

Error Case	Response Code	Error Response
Guest User exists but Email and SMS address empty	400 Bad Request	errorCode: NOTIFICATION_ERROR msg: Could not send notification. Cause: Guest User Email/SMS address empty.
Guest User exists but Email and SMS notification off in provisioning Group	400 Bad Request	errorCode: NOTIFICATION_ERROR msg: Could not send notification. Cause: Access Denied.
Email Notification disabled	400 Bad Request	errorCode: NOTIFICATION_ERROR msg: Error: Could not send the notification. Please contact your administrator.

Fetching Guest User details by username for a Provisioner

This API is used to get a particular Guest User details by username for a Provisioner.

Fetching Guest User details by username for a Provisioner	
URI	/api/guestUsers/guestUserDetails/{username} **Note**: username is Guest User name.
Method	GET

HTTP Headers	`Authorization Scheme: Basic (Base64 encryption)`
	`Authorization: username:password`
	`api-version:{VERSION}`
	`Accept: application/json or application/xml`

Response Code	`200 OK`
Response Payload	`Guest User Details`

Example	**Request**
	`GET /GuestManager/api/guestUsers/guestUserDetails/guestUser1 HTTP/1.1`
	`Host: 10.120.120.30`
	`api-version: v1.0`
	`Authorization: Basic dGVzdDp0ZXN0`
	`Accept:application/json`
	`Cache-Control: no-cache`
	Response Payload

Table continues…

Fetching Guest User details by username for a Provisioner

JSON Format

```
{
    "GuestUser": {
        "userName": "guestUser1",
        "email": "test@avaya.com",
        "smsAddress": "2991199112@ tmomail.net",
        "startTime": "2015/06/25 04:16:41 PM IST",
        "endTime": "2015/06/25 09:16:41 PM IST",
        "provisioningGroup": "pg-api-user",
        "provisioner": "Internal/pall",
        "guestDetails": "guest Details-DL"
    }
}
```

XML Format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<GuestUser>
    <userName>guestUser1</userName>
    <email>test@avaya.com</email>
    <smsAddress>2991199112@ tmomail.net</smsAddress>
    <startTime>2015/06/25 04:16:41 PM IST</startTime>
    <endTime>2015/06/25 09:16:41 PM IST</endTime>
    <provisioningGroup>pg-api-user</provisioningGroup>
    <provisioner>Internal/pall</provisioner>
    <guestDetails>guest Details-DL</guestDetails>
</GuestUser>
```

Variable definition

The following table describes the parameters of Guest User details.

Attribute	Type/Value	Description
userName	String	Specifies the username of guest user account.
email	String	Specifies the Email address of the guest user account.
smsAddress	String	Specifies the SMS Address of the guest user account. The format is cellphone@gateway. For example, 2991199112@tmomail.net
startTime	String	Specifies the activation time for guest user account. The format is yyyy/MM/dd hh:mm:ss a z For example, 2015/06/06 11:10:00 AM IST
endTime	String	Specifies the expiry time of guest user account

Table continues…

Attribute	Type/Value	Description
		The format is yyyy/MM/dd hh:mm:ss a z For example, 2015/06/06 18:10:00 PM IST.
provisioningGroup	String	Specifies the Provisioning group of the guest user account.
provisioner	String	Specifies the username of the Provisioner who registered the Guest User.
guestDetails	String	Guest User Details.

Fetching Guest Users iteratively for a Provisioner

Follow the below procedures in sequence to fetch guest users iteratively for a Provisioner.

1. Get Cursor Id. For more information, see GET Cursor Id on page 55.
2. Get next N Guest Users. For more information, see GET next N Guest Users on page 56.
3. Get first N Guest Users. For more information, see GET first N Guest Users on page 57.
4. Get last N Guest Users. For more information, see GET last N Guest Users on page 58.
5. Get count of total available Guest User records. For more information, see GET count of total available Guest User records on page 59.
6. Close Cursor Id. For more information, see Close Cursor Id on page 60.

GET Cursor Id

The GET Cursor Id is the first API call to get the paging info that contains cursorId, which is used for subsequent calls, to get the guest users iteratively.

GET Cursor Id	
URI	/api/guestUsers
Method	GET

HTTP Headers	`Authorization Scheme: Basic (Base64 encryption)`
	`Authorization: username:password`
	`api-version:{VERSION}`
	`Accept: application/json or application/xml`

Response Code	`200 OK`

Table continues…

GET Cursor Id	
Response Payload	PagingInfo which contains cursorId and total device records

Example	**Request**
	`GET /GuestManager/api/guestUsers HTTP/1.1`
	`Host: 10.120.120.30`
	`api-version: v1.0`
	`Authorization: Basic dGVzdDp0ZXN0`
	`Accept:application/json`
	`Cache-Control: no-cache`
	Response Payload
	`{`
	`"PagingInfo": {`
	`"cursorId": "13666304570298546472",`
	`"totalRecord": 10`
	`}`
	`}`

GET next N Guest Users

The GET next N devices is the API call to get the next N devices.

GET next N Guest Users	
URI	/api/guestUsers/next/{N}/{cursorId}
Method	GET
HTTP Headers	Authorization Scheme: Basic (Base64 encryption) Authorization: username:password api-version:{VERSION} Accept: application/json or application/xml
Response Code	200 OK
Response Payload	Guest User List

Example	**Request**
	`GET /GuestManager/api/guestUsers/next/2/13666304570298546472 HTTP/1.1`
	`Host: 10.120.120.30`
	`api-version: v1.0`
	`Authorization: Basic dGVzdDp0ZXN0`
	`Accept:application/json`
	`Cache-Control: no-cache`
	Response Payload
	`{`
	`"GuestUserList": {`
	`"GuestUser": [`
	`{`
	`"userName": "GuestUser1",`

Table continues…

GET next N Guest Users

<pre>"email": "test12@avaya.com",
"smsAddress": "9845342309@T-Mobile",
"startTime": "2015/06/24 04:16:41 PM IST",
"endTime": "2015/06/24 08:16:41 PM IST",
"provisioningGroup": "p-api-user-device",
"provisioner": "Internal/pall",
"guestDetails": "guest Details"
},
{
"userName": "GuestUser2",
"email": "test884@avaya.com",
"smsAddress": "9622000000@tmomail.net",
"startTime": "2015/06/24 04:16:41 PM IST",
"endTime": "2015/06/25 12:16:41 AM IST",
"provisioningGroup": "pg-fl-no",
"provisioner": "Internal/pall",
"guestDetails": "guest Details"
}
]
}
}

GET first N Guest Users

The GET first N Guest Users is the API call to get the first N devices.

GET first N Guest Users	
URI	/api/ guestUsers/first/{N}/{cursorId}
Method	GET
HTTP Headers	Authorization Scheme: Basic (Base64 encryption) Authorization: username:password api-version:{VERSION} Accept: application/json or application/xml
Response Code	200 OK
Response Payload	Guest User List

Example	**Request**
	GET /GuestManager/api/guestUsers/first/2/13666304570298546472 HTTP/1.1
	Host: 10.120.120.30
	api-version: v1.0
	Authorization: Basic dGVzdDp0ZXN0
	Accept:application/json
	Cache-Control: no-cache
	Response Payload
	{
	"GuestUserList": {
	"GuestUser": [

Table continues…

GET first N Guest Users

<pre>{
"userName": "GuestUser1",
"email": "testl2@avaya.com",
"smsAddress": "9845342309@T-Mobile",
"startTime": "2015/06/24 04:16:41 PM IST",
"endTime": "2015/06/24 08:16:41 PM IST",
"provisioningGroup": "p-api-user-device",
"provisioner": "Internal/pall",
"guestDetails": "guest Details"
},
{
"userName": "GuestUser2",
"email": "test884@avaya.com",
"smsAddress": "9622000000@tmomail.net",
"startTime": "2015/06/24 04:16:41 PM IST",
"endTime": "2015/06/25 12:16:41 AM IST",
"provisioningGroup": "pg-fl-no",
"provisioner": "Internal/pall",
"guestDetails": "guest Details"
}
]
}
}</pre>

GET last N Guest Users

The GET last N Guest Users is the API call to get the last N Guest Users of a Provisioner.

GET last N Guest Users	
URI	/api/ guestUsers/last/{N}/{cursorId}
Method	GET
HTTP Headers	Authorization Scheme: Basic (Base64 encryption) Authorization: username:password api-version:{VERSION} Accept: application/json or application/xml
Response Code	200 OK
Response Payload	Guest User List
Example	**Request** GET /GuestManager/api/guestUsers/last/2/1366630457029854647 2 HTTP/1.1 Host: 10.120.120.30 api-version: v1.0 Authorization: Basic dGVzdDp0ZXN0 Accept:application/json Cache-Control: no-cache

Table continues…

GET last N Guest Users	

	Response Payload
	<pre>{
	"GuestUserList": {
	"GuestUser": [
	{
	"userName": "GuestUser10",
	"email": "test10@avaya.com",
	"smsAddress": "9123456789@tmomail.net",
	"startTime": "2015/06/25 04:16:41 PM GMT+00:00",
	"endTime": "2015/06/25 09:16:41 PM GMT+00:00",
	"provisioningGroup": "pg-api-user",
	"provisioner": "Internal/pall",
	"guestDetails": "guest Details-DL"
	},
	{
	"userName": "GuestUser9",
	"email": "test9@avaya.com",
	"smsAddress": "9329393922@tmomail.net",
	"startTime": "2015/06/24 04:16:41 PM IST",
	"endTime": "2015/06/25 12:16:41 AM IST",
	"provisioningGroup": "pg-user-email-phone",
	"provisioner": "Internal/pall",
	"guestDetails": "guest Details"
	}
	]
	}
	}</pre>

GET count of total available Guest User records

The GET count of total available Guest User records is the API call to get the count of Guest User records of a Provisioner.

GET count of total available Guest User records	
URI	/api/ guestUsers/count/{cursorId}
Method	GET

HTTP Headers	`Authorization Scheme: Basic (Base64 encryption)`
	`Authorization: username:password`
	`api-version:{VERSION}`
	`Accept: application/json or application/xml`

Response Code	`200 OK`
Response Payload	`Count of Guest User`

Example	**Request**
	<pre>GET /GuestManager/api/guestUsers/count/13666304570298546472 HTTP/1.1
	Host: 10.120.120.30
	api-version: v1.0
	Authorization: Basic dGVzdDp0ZXN0</pre>

Table continues…

GET count of total available Guest User records	
	Accept:application/json Cache-Control: no-cache **Response** 10

Close Cursor Id

The Close Cursor Id API is used to clean up cache for this cursor id and subsequent requests in the server. After cleaning up, the Cursor Id will not be valid.

Close Cursor Id	
URI	/api/ guestUsers/close/{cursorId}
Method	GET
HTTP Headers	Authorization Scheme: Basic (Base64 encryption) Authorization: username:password api-version:{VERSION} Accept: application/json or application/xml
Response Code	204 No Content OK
Response Payload	Guest User List
Example	**Request** GET /GuestManager/api/guestUsers/close/13666304570298546472 HTTP/1.1 Host: 10.120.120.30 api-version: v1.0 Authorization: Basic dGVzdDp0ZXN0 Accept:application/json Cache-Control: no-cache