Avaya Identity Engines Ignition Guest Manager REST APIs

Release 9.2.3
NN47280-505
Issue 02.01
December 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: https://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 a 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 IF YOU PURCHASE A 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. YOUR USE OF THE HOSTED SERVICE SHALL BE LIMITED BY THE NUMBER AND TYPE OF LICENSES PURCHASED UNDER YOUR CONTRACT FOR THE HOSTED SERVICE, PROVIDED, HOWEVER, THAT FOR CERTAIN HOSTED SERVICES IF APPLICABLE, YOU MAY HAVE THE OPPORTUNITY TO USE FLEX LICENSES, WHICH WILL BE INVOICED ACCORDING TO ACTUAL USAGE ABOVE THE CONTRACT LICENSE LEVEL. CONTACT AVAYA OR AVAYA'S CHANNEL PARTNER FOR MORE INFORMATION ABOUT THE LICENSES FOR THE APPLICABLE HOSTED SERVICE, THE AVAILABILITY OF ANY FLEX LICENSES (IF APPLICABLE), PRICING AND BILLING INFORMATION, AND OTHER IMPORTANT INFORMATION REGARDING 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. 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 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. 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: 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 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 G.729 CODEC, 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 G.729 CODEC IS LICENSED BY SIPRO LAB TELECOM INC. SEE WWW.SIPRO.COM/CONTACT.HTML. 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

Customer acknowledges and agrees that it is responsible 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: https://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

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

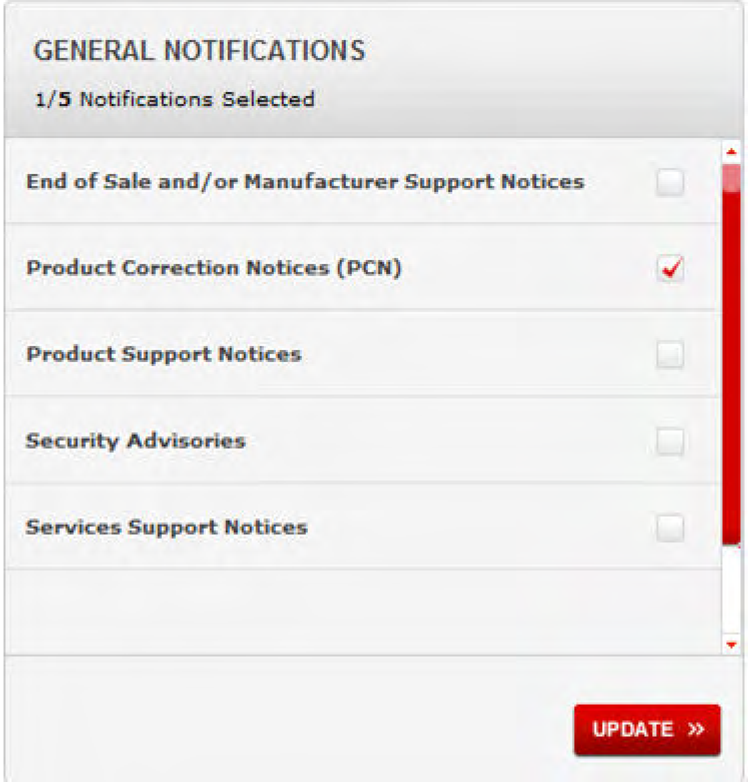
Subscribe to e-notifications to receive an email notification when documents are added to or changed on the Avaya Support website.

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 Virtual Services Platform 7000.

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**.



GENERAL NOTIFICATIONS

1/5 Notifications Selected

End of Sale and/or Manufacturer Support Notices □

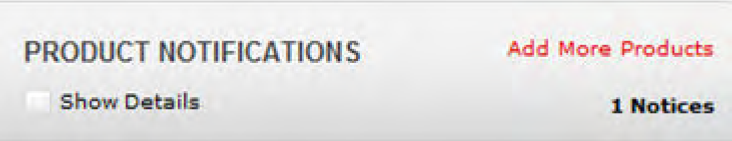
Product Correction Notices (PCN) ☑

Product Support Notices □

Security Advisories □

Services Support Notices □

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

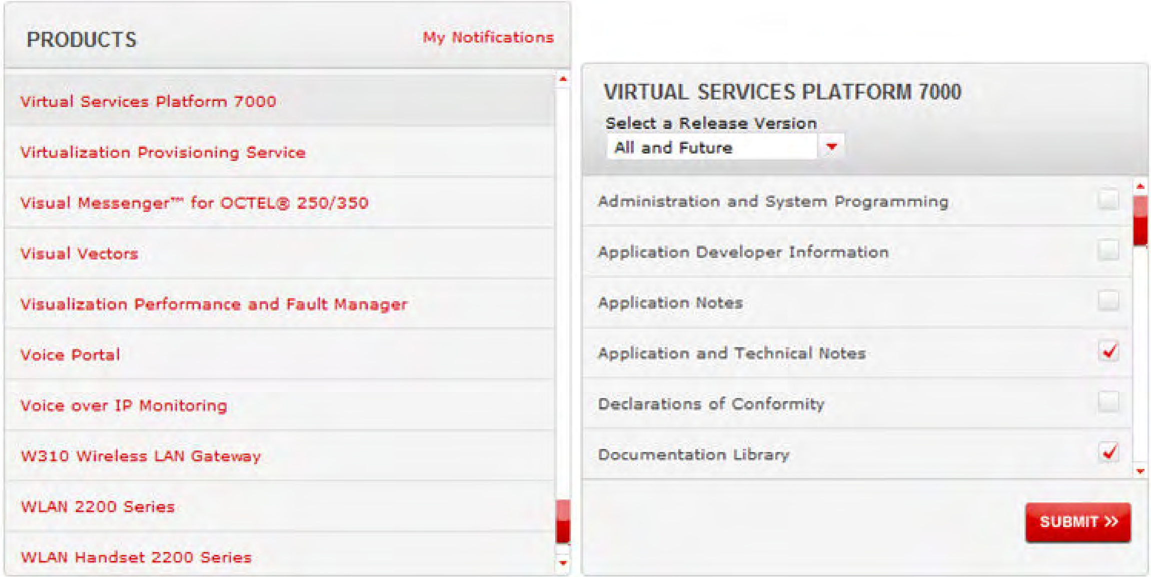


PRODUCT NOTIFICATIONS

□ Show Details

1 Notices

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.


PRODUCTS
My Notifications
Virtual Services Platform 7000
Virtualization Provisioning Service
Visual Messenger™ for OCTEL® 250/350
Visual Vectors
Visualization Performance and Fault Manager
Voice Portal
Voice over IP Monitoring
W310 Wireless LAN Gateway
WLAN 2200 Series
WLAN Handset 2200 Series
VIRTUAL SERVICES PLATFORM 7000
SUBMIT »

Select a Release Version: All and Future

- Administration and System Programming □
- Application Developer Information □
- Application Notes □
- Application and Technical Notes ☑
- Declarations of Conformity □
- Documentation Library ☑

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

The following section details what is new in *Avaya Identity Engines Ignition Guest Manager REST APIs* for Release 9.2.3.

API Version

In this release the Guest Manager is compatible with REST API versions v1.0.0 and v1.1.0

It is recommended to use the new version v1.1.0.

Other Changes

Two parameters firstName and lastName are added in Guest User details response.

Two parameters vlanId and vlanLabel are added for create device and device details response.

Supported APIs for Release 9.2.3

See the following table for the supported APIs for release 9.2.3:

API	URI	Reference	API Status
Fetching Provisioning groups for a Provisioner	/api/provisioningGroups	For more information, see Fetching Provisioning group for a Provisioner on page 28.	No Changes
Fetching Provisioning Group details for Group name	/api/ provisioningGroupDetails /{groupName}	For more information, see Fetching Provisioning Group details for Group name on page 29.	No Changes
Device Registration	/api/devices	For more information, see Device Registration REST API on page 37.	Updated
Update a device	/api/devices/{MAC}	For more information, see Update a device on page 41.	New
Delete a device	/api/devices/{MAC}	For more information, see Delete a device on page 42.	New

Table continues…

API	URI	Reference	API Status
Deleting multiple devices	/api/devices	For more information, see Deleting multiple devices on page 43.	New
Bulk Delete of devices for a Provisioner	/api/devices/ bulkDelete?hideDeleteDetails=true	For more information, see Bulk Delete of devices for a Provisioner on page 45.	New
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 46.	Updated
Fetching Devices iteratively for a Provisioner		For more information, see Fetching Devices iteratively for a Provisioner on page 48.	No Changes
GET Cursor Id of device	/api/devices	For more information, see GET Cursor Id on page 49.	No Changes
GET next N devices	/api/devices/next/{N}/{cursorId}	For more information, see GET next N devices on page 50.	Updated
GET first N devices	/api/devices/first/{N}/{cursorId}	For more information, see GET first N devices on page 51.	Updated
GET last N devices	/api/devices/last/{N}/{cursorId}	For more information, see GET last N devices on page 52.	Updated
GET count of total available device records	/api/devices/count/{cursorId}	For more information, see GET count of total available device records on page 53.	No Changes
Close Cursor Id	/api/ devices/close/{cursorId}	For more information, see Close Cursor Id on page 54.	No Changes
Fetching devices with filter	/api/devices?filterCriteria=<field>&op=<op value>&val=<value>	For more information, see Fetching devices with filter on page 55.	New
Fetching devices with filter and without details		For more information, see Fetching devices with filter and without details on page 59.	New

Table continues…

API	URI	Reference	API Status
API to query the status of single device	/api/devices/ deviceStatusQuery/ {MAC}	For more information, see API to query the status of single device on page 61.	New
API to query the status of multiple devices	/api/devices/ deviceStatusQuery? macs=mac1|mac2|mac3	For more information, see API to query the status of multiple devices on page 62.	New
Guest User Registration	/api/guestUsers	For more information, see Guest User Registration REST API on page 63.	Updated
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 69.	No Changes
Update a Guest User	/api/guestUsers/ {username}	For more information, see Update a Guest User on page 70.	New
Delete a Guest User	/api/ guestUsers/ {username}	For more information, see Delete a Guest User on page 72.	New
Deleting multiple Guest Users	/api/guestUsers	For more information, see Deleting multiple Guest Users on page 73.	New
Bulk Delete of Guest Users for a Provisioner	/api/guestUsers/ bulkDelete? hideDeleteDetails=true	For more information, see Bulk Delete of Guest Users for a Provisioner on page 75.	New
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 76.	Updated
Fetching Guest Users iteratively for a Provisioner		For more information, see Fetching Guest Users iteratively for a Provisioner on page 78.	No Changes

Table continues…

API	URI	Reference	API Status
GET Cursor Id	/api/guestUsers	For more information, see GET Cursor Id on page 79.	No Changes
GET next N Guest Users	/api/guestUsers/next/{N}/{cursorId}	For more information, see GET next N Guest Users on page 79.	Updated
GET first N Guest Users	/api/ guestUsers/first/{N}/{cursorId}	For more information, see GET first N Guest Users on page 80.	Updated
GET last N Guest Users	/api/ guestUsers/last/{N}/{cursorId}	For more information, see GET last N Guest Users on page 81.	Updated
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 83.	No Changes
Close Cursor Id	/api/ guestUsers/close/{cursorId}	For more information, see Close Cursor Id on page 83.	No Changes
Fetching Guest User with filter	/api/guestUsers?filterCriteria=<field>&op=<op value>&val=<value>	For more information, see Fetching Guest User with filter on page 84.	New
Fetching Guest Users with filter and without details		For more information, see Fetching Guest Users with filter and without details on page 88.	New
API to query the status of single user	/api/guestUsers/userStatusQuery/{userName}	For more information, see API to query the status of single user on page 89.	New
API to query the status of multiple users	/api/guestUsers/userStatusQuery?userNames=username1|username2|username3	For more information, see API to query the status of multiple users on page 90.	New

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 17
2. Create Provisioner in GM. For more information, see Creating Provisioner and associating it with Provisioning Groups on page 18
3. Download and Install RESTClient plugin in Firefox. For more information, see Downloading and Installing Firefox RESTClient plugin on page 18

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

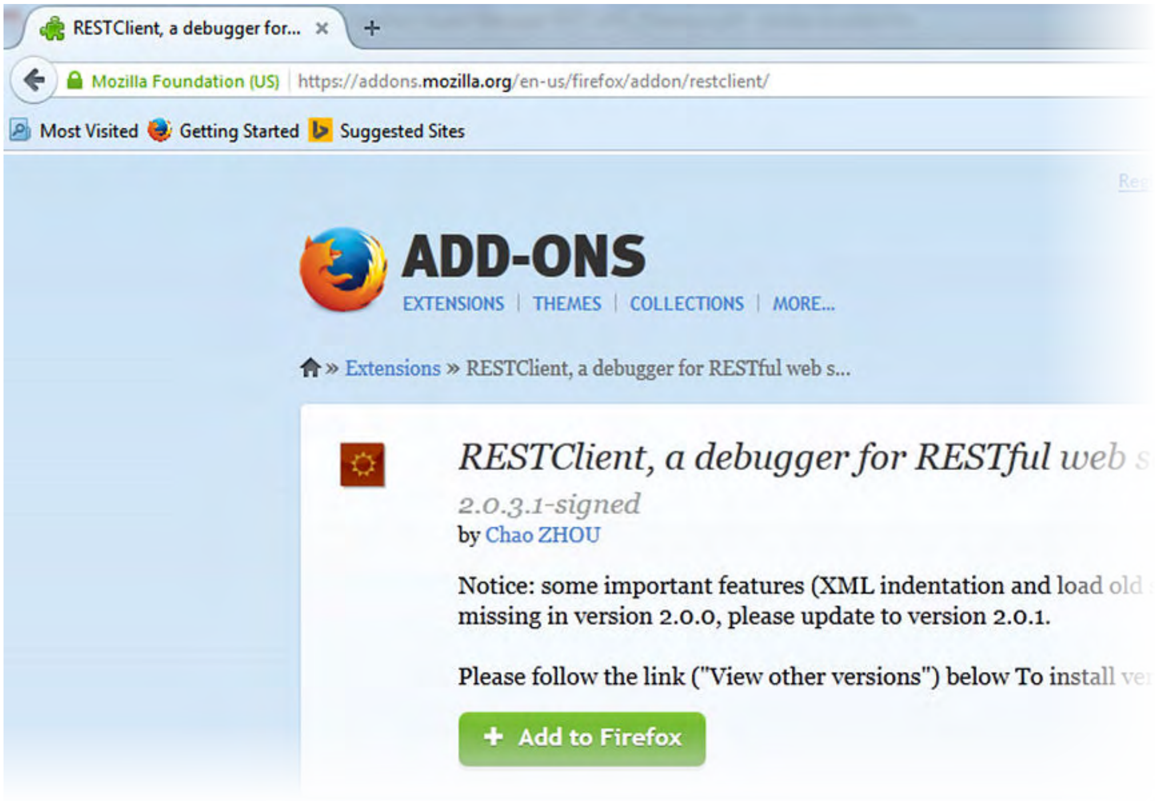
This section describes the procedure to download and install the Firefox RESTClient plugin.

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**.



RESTClient, a debugger for...
Mozilla Foundation (US)
https://addons.mozilla.org/en-us/firefox/addon/restclient/
Most Visited
Getting Started
Suggested Sites
ADD-ONS
EXTENSIONS
THEMES
COLLECTIONS
MORE...
» Extensions » RESTClient, a debugger for RESTful web s...
+ Add to Firefox

RESTClient, a debugger for RESTful web s

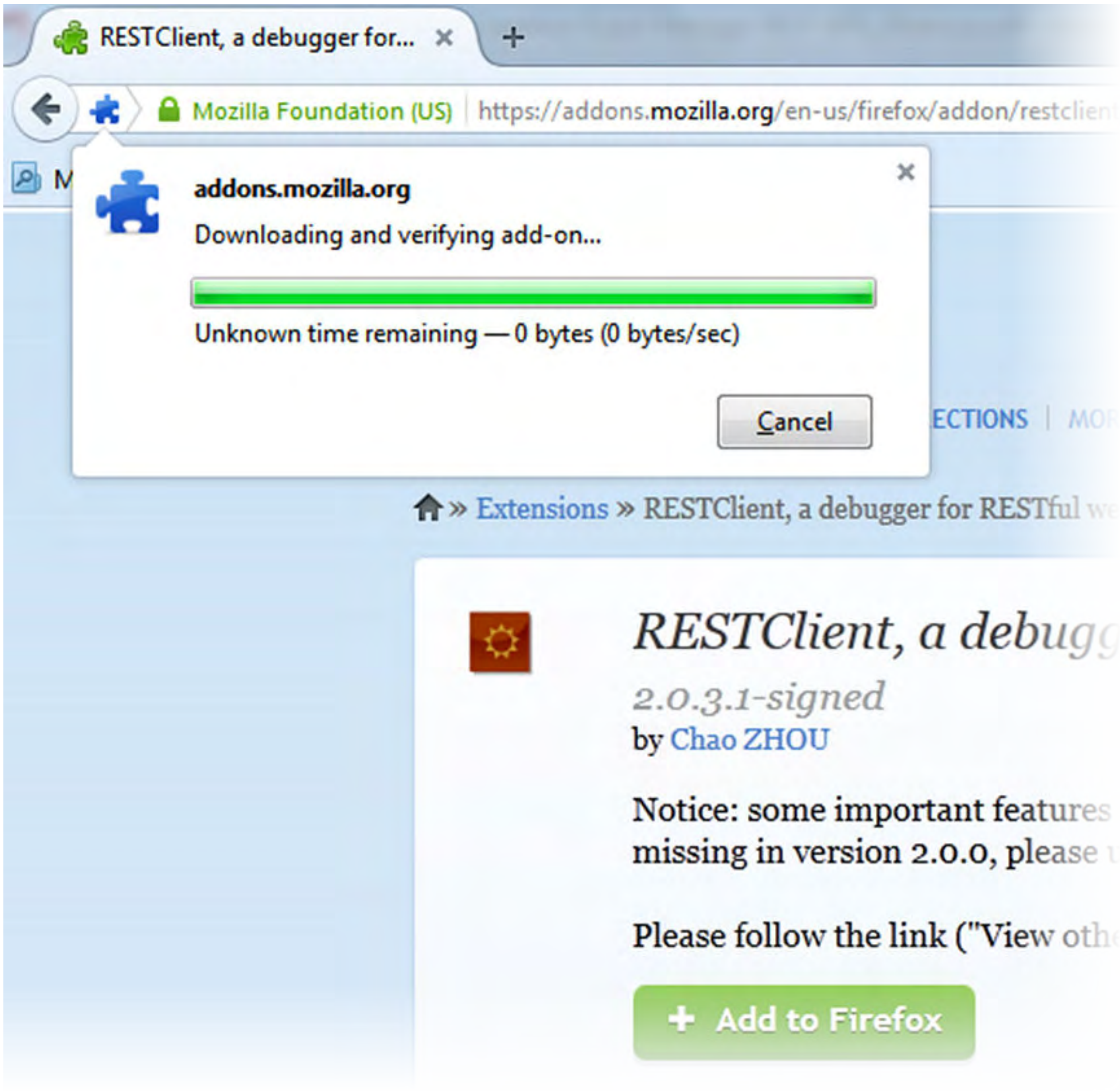
2.0.3.1-signed

by Chao ZHOU

Notice: some important features (XML indentation and load old missing in version 2.0.0, please update to version 2.0.1.

Please follow the link ("View other versions") below To install ve

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



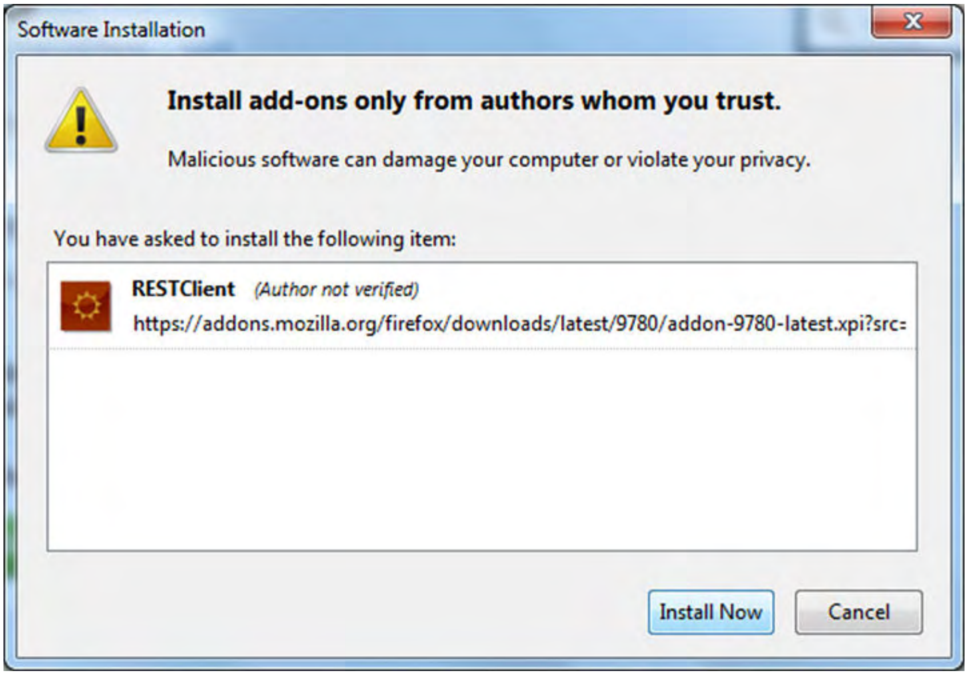
addons.mozilla.org

Downloading and verifying add-on...

Unknown time remaining — 0 bytes (0 bytes/sec)

RESTClient, a debugger for...
Mozilla Foundation (US)
https://addons.mozilla.org/en-us/firefox/addon/restclien
Cancel
ECTIONS
MOR
» Extensions » RESTClient, a debugger for RESTful we
RESTClient, a debug
2.0.3.1-signed
by Chao ZHOU
Notice: some important features
missing in version 2.0.0, please u
Please follow the link ("View oth
+ Add to Firefox

3. Click **Install Now**.



Software Installation

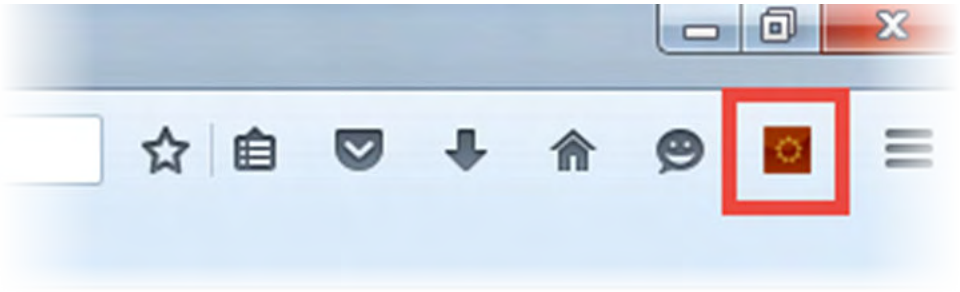
Install add-ons only from authors whom you trust.

Malicious software can damage your computer or violate your privacy.

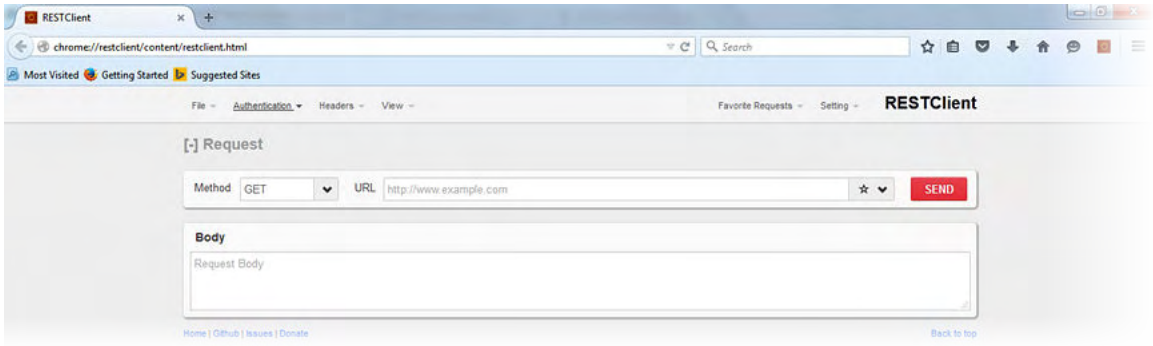
You have asked to install the following item:

RESTClient *(Author not verified)*
https://addons.mozilla.org/firefox/downloads/latest/9780/addon-9780-latest.xpi?src=

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



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



RESTClient

chrome://restclient/content/restclient.html

[-] Request

Method GET URL http://www.example.com

Body

Request Body

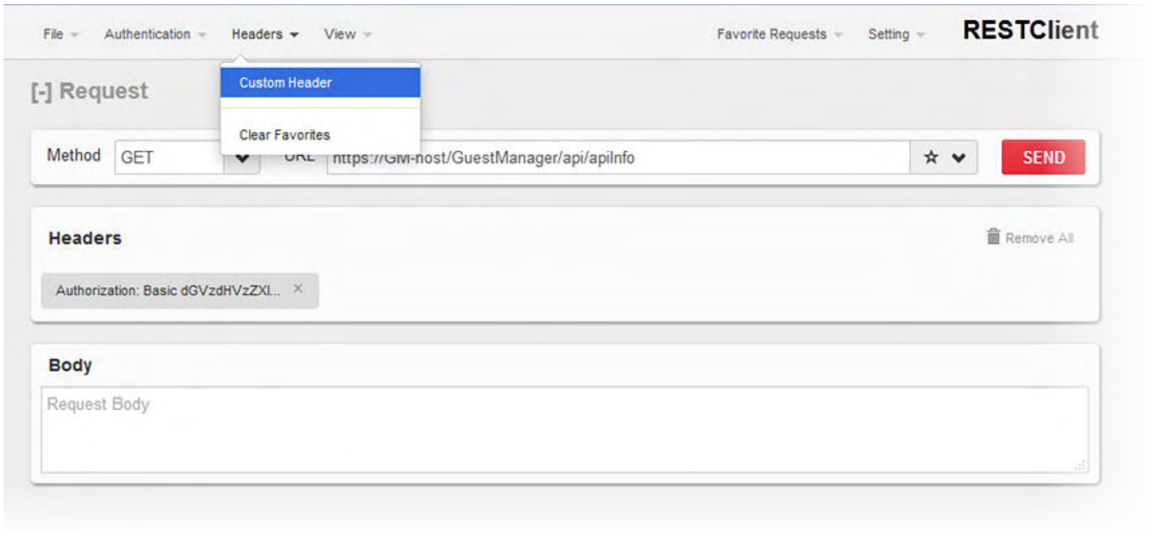
Guest Manager REST API Version

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 v1.1.0 (**api-version:v1.1.0**). In this Release the GM is compatible with REST API versions v1.0.0 and v1.1.0.

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

Procedure

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



Method: GET

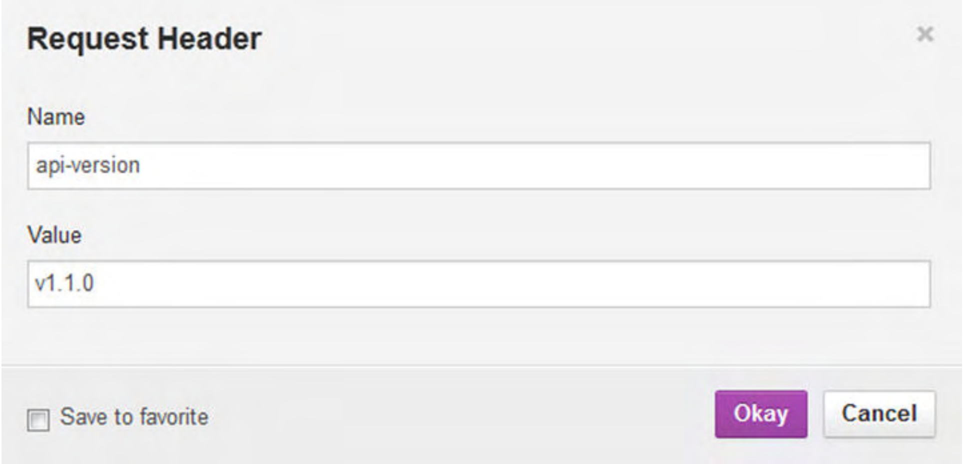
URL: https://GM-host/GuestManager/api/apiInfo

Headers

Authorization: Basic dGVzdHVzZXl...

Body

The Request Header window appears.



Request Header

Name

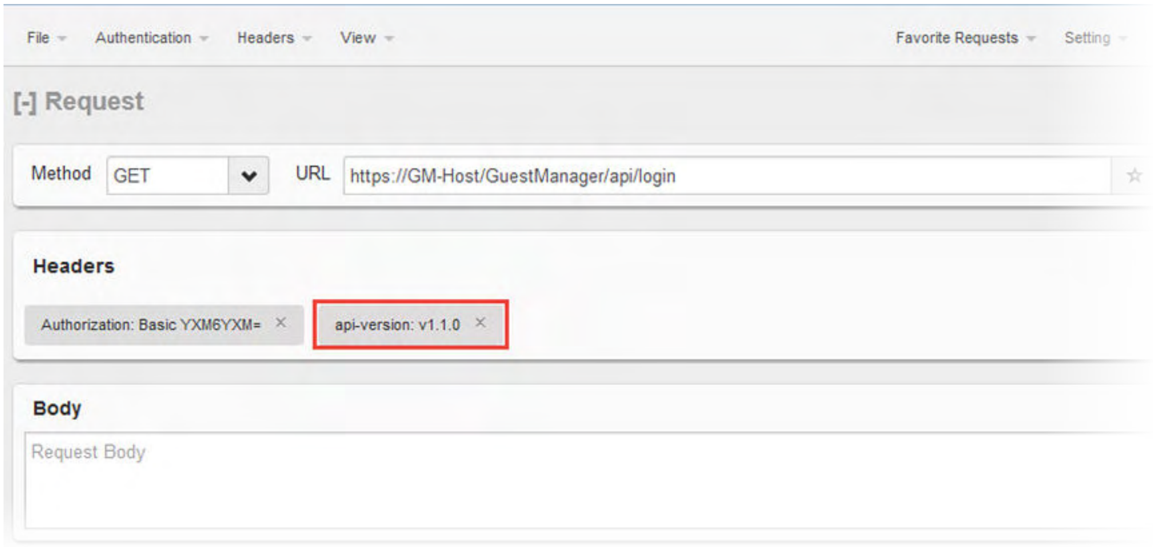
api-version

Value

v1.1.0

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

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



Method GET

URL https://GM-Host/GuestManager/api/login

Headers

Authorization: Basic YXM6YXM=

api-version: v1.1.0

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**

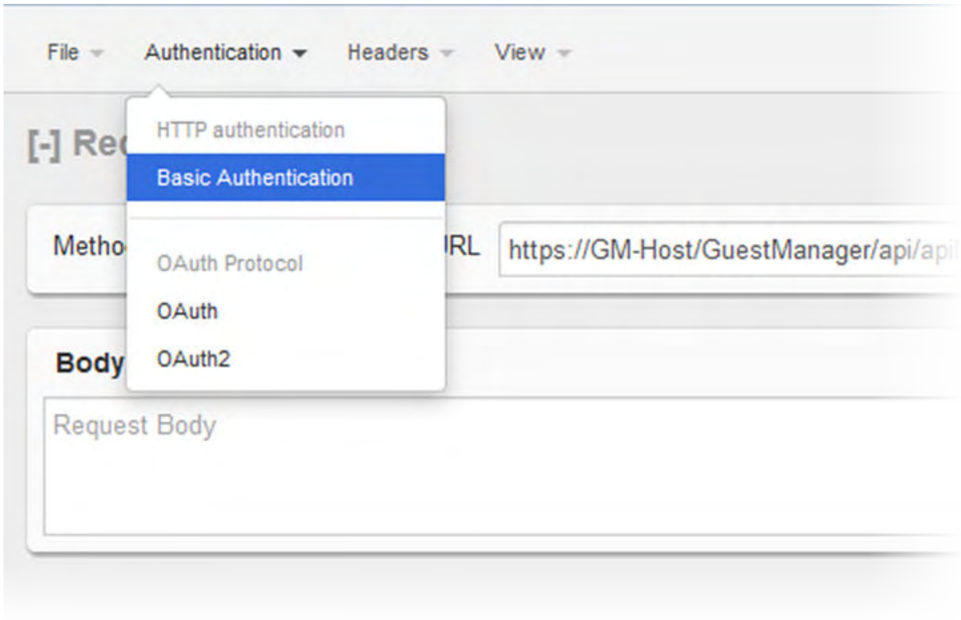
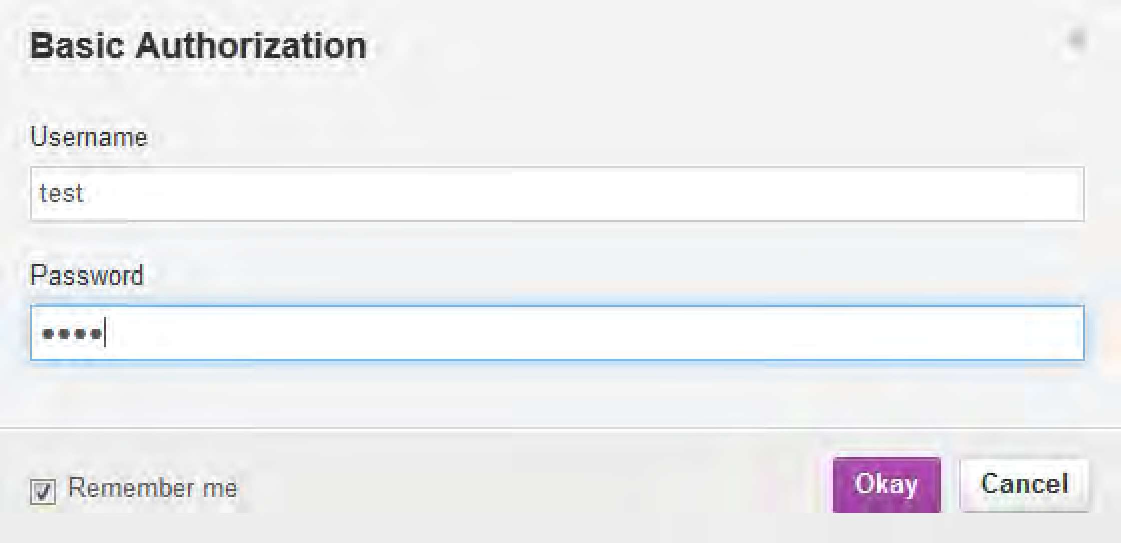


Figure 1: Basic Authentication



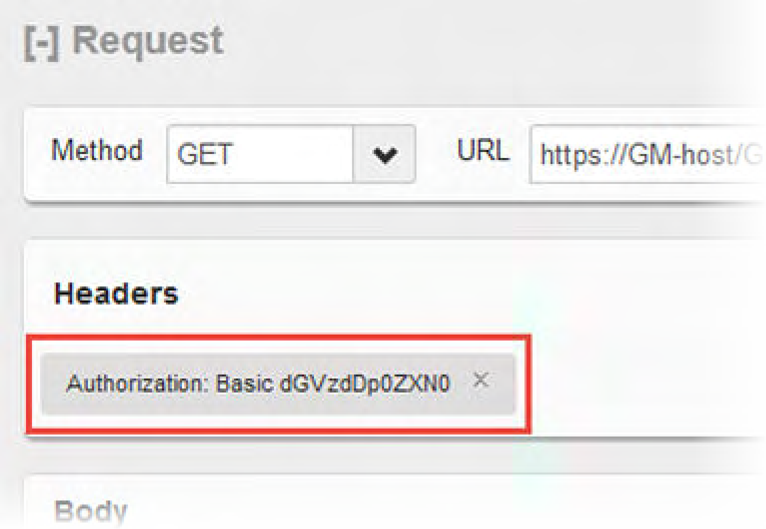
Basic Authorization

Username: test

Password: ••••

☑ Remember me

Figure 2: Basic Authorization



[-] Request

Method: GET

URL: https://GM-host/G

Headers

Authorization: Basic dGVzdDp0ZXN0

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`
	Response
	JSON Format
	`{`
	`    "apiPath": "/api",`
	`    "name": "Ignition Guest Manager REST API",`
	`    "productName": "Avaya Identity Engines Ignition Guest`
	`Manager",`
	`    "vendor": "Avaya Inc.",`
	`    "version": "v1.1.0"`
	`}`

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.

Table continues…

Error Case	Response Code	Error Respone
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 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

Table continues…

Error Case	Response Code	Error Respone
		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 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.

<table>
<tr><th colspan="2">Fetching Provisioning group for a Provisioner</th></tr>
<tr><td><b>URI</b></td><td>/api/provisioningGroups</td></tr>
<tr><td><b>Method</b></td><td>GET</td></tr>
<tr><td><b>HTTP Headers</b></td><td>

```
Authorization Scheme: Basic (Base64 encryption)

Authorization: username:password

api-version: {VERSION}

Accept: application/json or application/xml
```

</td></tr>
<tr><td><b>Response Code</b></td><td>

```
200 OK
```

</td></tr>
<tr><td><b>Response Payload</b></td><td>

```
List of Provisioning Group name
```

</td></tr>
<tr><td><b>Example</b></td><td>

<b>Request</b>

```
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
```

<b>Response Payload</b>

<b>JSON Format</b>

```
{
    "ProvisioningGroups": {
        "groupName": [
            "api-device-provGroup",
            "api-device-provGroup1",
            "api-device-provGroup2"
        ]
```

</td></tr>
</table>

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…

<table>
<tr><th colspan="2">Device Registration REST API</th></tr>
<tr><td></td><td><pre>Content-Type: application/json or application/xml</pre></td></tr>
<tr><td><b>Response Code</b></td><td><pre>201 (created)</pre></td></tr>
<tr><td><b>Response Payload</b></td><td>N/A</td></tr>
<tr><td><b>Example</b></td><td>
<b>Request</b>
<pre>
POST /GuestManager/api/devices HTTP/1.1
Host: 10.120.120.30
api-version: v1.0
Authorization: Basic dGVzdDp0ZXN0
Accept:application/json
Content-Type: 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", "vlanLabel": "vlan-100",
"vlanId": "100" } }
</pre>
<b>Response Header</b>
<pre>
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
</pre>
<b>Request Payload</b>

<b>JSON Format</b>
<pre>
{
  "Device": {
        "provisioningGroupName": "api-device-provGroup",
        "macAddress": "10:10:10:00:00:01",
        "name": "device1",
        "type": "mobile",
        "subType": "generic-android",
        "vlanLabel": "vlan-100",
        "vlanId": "100"
   }
}
</pre>
<b>XML Format</b>
<pre>
<Device>
  <provisioningGroupName>api-device-provGroup</provisioningGroupName>
  <macAddress>10:10:10:00:00:01</macAddress>
  <name>device1</name>
  <type>mobile</type>
  <subType>generic-android</subType>
  <vlanLabel>vlan-100</vlanLabel>
  <vlanId>100</vlanId>
</Device>
</pre>
<b>Error Response</b>
<pre>
{
    "error": {
        "errorCode": "AUTHORIZATION_REQUIRED",
        "msg": "Authorization required."
    }
}
</pre>
</td></tr>
</table>

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 150 characters and the allowed characters are a-z A-Z 0-9-_~$&+,.:;=?@#|`'<>.^*()%! []{} \ /.
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.
vlanLabel	String	Specifies the VLAN Label. The VLAN Label depends on the Provisioning Group settings. Parameters and value is: • if VLAN is accessible to provisioner in Provisioning Group then vlanLabel is used, otherwise ignored. The maximum length of the vlanLabel must be 150 characters and the allowed characters are a-z A-Z 0-9-_~$&+,:;=?@#|`'<>.^*()%! []{}\ /.
vlanId	Integer	Specifies the vlanId of a device. The VLAN ID depends on the Provisioning Group settings. Parameters and value is: • if VLAN is accessible to provisioner in Provisioning Group then vlanId is used, otherwise ignored. The vlanId must be in the range of 0-4095.

Update a device

The API is used to update a device for a MAC.

Note:

Error response is sent if device is expired.

Provisioning Group Name associated with this Device cannot be modified. If this information is passed in the API, it will be skipped.

Update a device	
URI	/api/devices/{MAC}
Method	PUT

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	`200 OK`
Response Payload	`Device record updated successfully`

Example	**Request**
	`PUT /GuestManager/api/devices/10:0b:01:20:00:06 HTTP/1.1`
	`Host: 10.120.120.30`
	`api-version: v1.1.0`
	`Authorization: Basic dGVzdDp0ZXN0`
	`Accept: application/json`
	`Content-Type: application/json`
	`Cache-Control: no-cache`
	Request Payload
	JSON
	`{`
	`    "Device": {`
	`        "name": "device1",`
	`        "type": "mobile",`
	`        "subType": "generic-android",`
	`        "vlanLabel": "vlan-100",`
	`        "vlanId": "100"`
	`        }`
	`}`
	XML
	`<Device>`
	`  <name>device1</name>`
	`  <type>mobile</type>`
	`  <subType>generic-android</subType>`
	`  <vlanLabel>vlan-100</vlanLabel>`
	`  <vlanId>100</vlanId>`
	`</Device>`

For more information about the variable definitions, see Variable definition on page 47.

Error Cases

The following table describes the error cases for updating device for a MAC.

Error Case	Response Code	Error Response
Device does not exists	404 not found	
Device access denied	400 Bad Request	errorCode: DEVICE_ACCESS_DENIED msg: Your account does not have permission to access the Device: {macAddress}. If Provisioner can access each other's record with this group is true then allow to edit the device by other provisioner and provisionedBy will be updated .
Device already expired	400 Bad Request	errorCode: DEVICE_EXPIRED msg: Device record already expired.

Delete a device

The API is used to delete a single device for a MAC.

Delete a device	
URI	/api/devices/{MAC}
Method	DELETE

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	`200 OK`
Response Payload	`Device record deleted successfully.`

Table continues…

Delete a device	

Example	**Request**
	`DELETE /GuestManager/api/devices/10:0b:01:20:00:06 HTTP/1.1`
	`Host: 10.120.120.30`
	`api-version: v1.1.0`
	`Authorization: Basic dGVzdDp0ZXN0`
	`Accept: application/json`
	`Content-type: application/json`
	`Cache-Control: no-cache`

Error Cases

The following table describes the error cases for deleting a device record for a MAC.

Error Case	Response Code	Error Response
Device does not exists	404 not found	
Device access denied	400 Bad Request	errorCode: DEVICE_ACCESS_DENIED msg: Your account does not have permission to delete the Device: {macAddress}. If Provisioner can access each other's record with this group is true then allow to delete the device.

Deleting multiple devices

The API is used to delete multiple devices.

Note:

Maximum 500 devices can be sent to delete.

Deleting multiple devices	
URI	/api/devices
Method	DELETE

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

Table continues…

Deleting multiple devices	
	`Content-Type: application/json or application/xml`
Response Code	`200 OK`
Response Payload	`List of device MAC addresses`
Example	**Request**

Request

```
DELETE /GuestManager/api/devices/10:0b:01:20:00:06 HTTP/1.1
Host: 10.120.120.30
api-version: v1.1.0
Authorization: Basic dGVzdDp0ZXN0
Accept: application/json
Content-type: application/json
Cache-Control: no-cache
```

Request Payload

```
{
    "DeviceList": {
        "Device": [
            {
                "macAddress": "10:0b:01:20:00:06"
            },
            {
                "macAddress": "10:10:10:00:00:02"
            }
        ]
    }
}
```

Response Payload

If all devices are deleted successfully.

```
{
   Message:” All Devices are deleted successfully.”
     successList:{ "Device": [
            {
                "macAddress": "10:0b:01:20:00:06"
            },
            {
                "macAddress": "10:10:10:00:00:02"
            } ]
    }
}
```

If Partial delete success

```
{
   Message:” Devices are deleted partially, please check the
successList and failedList for detail”
     successList:{ "Device": [
            {
                "macAddress": "10:0b:01:20:00:06"
            },
            {
                "macAddress": "10:10:10:00:00:02"
            } ]
    }
    failsList:{ "Device": [
            {
                "macAddress": "10:0b:01:20:00:0a",
```

Table continues…

Deleting multiple devices

```
                “reason”: ERROR-RecordNotFound
            },
            {
                "macAddress": "10:10:10:00:00:0b"
                “reason”: ERROR-AccessDenied
            } ]
    }
}
```

Bulk Delete of devices for a Provisioner

The API is used to delete all devices (bulk delete) of a provisioner.

Note:

At a time maximum 2000 records can be deleted, if records are more than 2000 then repeat the operation.

Bulk Delete of devices for a Provisioner	
URI	/api/devices/ bulkDelete?hideDeleteDetails=true **Note:** hideDeleteDetails is a optional parameter. If hideDeleteDetails is false, response will contain deleted device details. If hideDeleteDetails is true, response will not contain deleted device details. If hideDeleteDetails is not passed as argument, then response will contain deleted device details.
Method	DELETE

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	`200 OK`
Response Payload	`List of MAC Address of deleted devices`

Example	**Request**
	`DELETE /GuestManager/api/devices/bulkDelete?hideDeleteDetails=false HTTP/1.1`
	`Host: 10.120.120.30`
	`api-version: v1.1.0`
	`Authorization: Basic dGVzdDp0ZXN0`

Table continues…

Bulk Delete of devices for a Provisioner

```
Accept: application/json
Content-type: application/json
Cache-Control: no-cache
```

Response Payload

If devices are more than the limit (2000)

```
{
   Message:” First 2000 Devices are deleted successfully.
   repeatRequired: true,
   successList:{ "Device": [
            {
                "macAddress": "10:0b:01:20:00:06"
            },
            {
                "macAddress": "10:10:10:00:00:02"
            } ]
    }
}
```

If hidedeleteDetails is true

```
{
   Message:” First 2000 Devices are deleted successfully.
   repeatRequired: true
}
```

Note:

If repeatRequired is true, then more than 2000 records exist. Repeat the operation to delete all records.

If devices are less than or equal to the limit (2000)

```
{
   Message:” All Devices are deleted successfully.”
     successList:{ "Device": [
            {
                "macAddress": "10:0b:01:20:00:06"
            },
            {
                "macAddress": "10:10:10:00:00:02"
            } ]
    }
}
```

If hidedeleteDetails is true

```
{
   Message:” All Devices are deleted successfully.”
}
```

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
```

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",
        "vlanLabel": "vlan-100",
        "vlanId": "100"
    }
}
```

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.

Table continues…

Attribute	Type/Value	Description
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.
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.
vlanLabel	String	Specifies the VLAN Label.
vlanId	Integer	Specifies the vlanId of a 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 49.
2. Get next N devices. For more information, see GET next N devices on page 50.
3. Get first N devices. For more information, see GET first N devices on page 51.
4. Get last N devices. For more information, see GET last N devices on page 52.

5. Get count of total available device records. For more information, see GET count of total available device records on page 53.
6. Close Cursor Id. For more information, see Close Cursor Id on page 54.

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}`
	`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.

<table>
<tr><th colspan="2">GET next N devices</th></tr>
<tr><td><b>URI</b></td><td>/api/devices/next/{N}/{cursorId}</td></tr>
<tr><td><b>Method</b></td><td>GET</td></tr>
<tr><td><b>HTTP Headers</b></td><td><code>Authorization Scheme: Basic (Base64 encryption)</code>
<code>Authorization: username:password</code>
<code>api-version:{VERSION}</code>
<code>Accept: application/json or application/xml</code></td></tr>
<tr><td><b>Response Code</b></td><td><code>200 OK</code></td></tr>
<tr><td><b>Response Payload</b></td><td><code>Device List</code></td></tr>
<tr><td><b>Example</b></td><td><b>Request</b>
<pre>
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
</pre>
<b>Response Payload</b>
<pre>
{
    "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",
                "vlanLabel": "vlan-100",
                "vlanId": "100"
            },
            {
                "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": "-",
</pre></td></tr>
</table>

Table continues…

GET next N devices
"provisioningGroup": "api-device-provGroup", "provisioner": "Internal/pall" "vlanLabel": "vlan-100", "vlanId": "100" }] } }

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) 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"
	"vlanLabel": "vlan-100",

Table continues…

GET first N devices

```
                "vlanId": "100"
            },
            {
                "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"
                "vlanLabel": "vlan-100",
                "vlanId": "100"
            }
        ]
    }
}
```

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": [
            {
```

Table continues…

GET last N devices

```
                                "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"
                                "vlanLabel": "vlan-100",
                                "vlanId": "100"
                            },
                            {
                                "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"
                                "vlanLabel": "vlan-100",
                                "vlanId": "100"
                            }
                        ]
                    }
                }
```

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`

Table continues…

GET count of total available 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`
Response Payload	NA

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

Table continues…

Error Case	Response Code	Error Response
		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

Fetching devices with filter

The API is used to fetch devices with filter iteratively.

Get cursor Id is the first API call to specify filter criteria and get the paging information that has cursor Id that are used in subsequent calls to get the devices iteratively.

The other API calls, **Get Next N devices**, **Get first N devices**, **Get last N devices**, **Get count of total available devices**, and **Close cursor Id** are same as in *Fetching Devices iteratively for a Provisioner* section. For more information, see Fetching Devices iteratively for a Provisioner on page 48.

GET Cursor Id	
URI	/api/devices?filterCriteria=<field>&op=<op value>&val=<value>
Method	GET

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	`200 OK`
Response Payload	`PagingInfo which contains cursorId and total device records`

Example	**Request**
	`GET /GuestManager/api/devices?`
	`filterCriteria=name&op=startWith&val=device HTTP/1.1`
	`Host: 10.120.120.30`
	`api-version: v1.1.0`
	`Authorization: Basic dGVzdDp0ZXN0`
	`Accept: application/json`
	`Content-Type:application/json`
	Response Payload

Table continues…

GET Cursor Id

GET Cursor Id

```
{
  "PagingInfo": {
    "cursorId": "16769248859629549495",
    "totalRecord": 11
  }
}
```

GET next N devices

URL: https://GuestManager/api/devices/next/2/16769248859629549495

```
{
  "DeviceList": {
    "Device": [
      {
        "macAddress": "11:11:11:11:11:12",
        "name": "Test1",
        "type": "mobile",
        "subType": "generic-android",
        "source": "GM-default",
        "enabled": true,
        "assetType": "PERMANENT",
        "startTime": "2015/11/09 09:24:45 AM GMT",
        "endTime": "-",
        "provisioningGroup": "default",
        "provisioner": "Internal/as"
        "vlanLabel": "vlan-100",
        "vlanId": "100"
      },
      {
        "macAddress": "11:11:11:11:11:13",
        "name": "Test2",
        "type": "mobile",
        "subType": "generic-android",
        "source": "GM-default",
        "enabled": true,
        "assetType": "TEMPORARY",
        "startTime": "2015/10/13 02:22:31 PM GMT",
        "endTime": "2015/10/13 10:22:31 PM GMT",
        "provisioningGroup": "default",
        "provisioner": "Internal/as"
        "vlanLabel": "vlan-100",
        "vlanId": "100"
      }
    ]
  }
}
```

GET first N devices

URL: https://GuestManager/api/devices/first/2/16769248859629549495

```
{
  "DeviceList": {
    "Device": [
      {
        "macAddress": "22:22:22:22:22:22",
        "name": "Test3",
        "type": "FA client",
        "subType": "ONA-SDN",
```

Table continues…

GET Cursor Id

```
        "source": "GM-default",
        "enabled": true,
        "assetType": "TEMPORARY",
        "startTime": "2015/10/12 06:48:05 AM GMT",
        "endTime": "2015/10/12 02:48:05 PM GMT",
        "provisioningGroup": "default",
        "provisioner": "Internal/as"
        "vlanLabel": "vlan-100",
        "vlanId": "100"
      },
      {
        "macAddress": "33:33:33:33:33:33",
        "name": "Test4",
        "type": "voip phone",
        "subType": "ONA-SDN",
        "source": "GM-default",
        "enabled": true,
        "assetType": "TEMPORARY",
        "startTime": "2015/10/11 10:00:54 AM GMT",
        "endTime": "2015/10/11 06:00:54 PM GMT",
        "provisioningGroup": "default",
        "provisioner": "Internal/as"
        "vlanLabel": "vlan-100",
        "vlanId": "100"
      }
    ]
  }
}
```

GET last N devices

URL: https://GuestManager/api/devices/last/2/16769248859629549495

```
{
  "DeviceList": {
    "Device": [
      {
        "macAddress": "00:0a:95:9d:68:45",
        "name": "Test5",
        "type": "Mobile",
        "subType": "generic-android",
        "source": "GM-default",
        "enabled": true,
        "assetType": "TEMPORARY",
        "startTime": "First Login Pending",
        "endTime": "-",
        "provisioningGroup": "karthik_group",
        "provisioner": "Internal/as"
        "vlanLabel": "vlan-100",
        "vlanId": "100"
      },
      {
        "macAddress": "00:0a:95:9d:68:34",
        "name": "Test6",
        "type": "",
        "subType": "n/a",
        "source": "GM-default",
        "enabled": true,
        "assetType": "PERMANENT",
        "startTime": "2015/10/30 07:04:27 AM GMT",
        "endTime": "-",
        "provisioningGroup": "default",
```

Table continues…

GET Cursor Id

```
                          "provisioner": "Internal/as"
                          "vlanLabel": "vlan-100",
                          "vlanId": "100"
                        }
                      ]
                    }
                  }
```

Filter details for a device

The following table describes the filter details for a device

Filter Criteria	Options	Value Type	Description
macAddress	equal notEqual startWith endsWith contains	String	MAC address
name	equal notEqual startWith endsWith contains	String	Device name
source	equal notEqual startWith endsWith contains	String	Device source
type	equal notEqual startWith endsWith contains	String	Device Type
startTime	greaterThan greaterThanEqual lessThan	Date Format: yyyy/MM/dd hh:mm:ss a z	Start Time of Device activation

Table continues…

Filter Criteria	Options	Value Type	Description
	lessThanEqual	ex: 2015/09/08 10:10:22 AM IST	
endTime		Date Format: yyyy/MM/dd hh:mm:ss a z ex: 2015/09/08 10:10:22 AM IST	Expiry time of device
provisionerGroup	Equal	String	Provisioning Group name Note: Provisioning group must be accessible to provisioner and have device rights.

Fetching devices with filter and without details

The API is used to fetch devices with filter and without device details.

The filter details are same as mentioned in the previous section. For more information, see Fetching devices with filter on page 55. Optional query parameter hideDetails=true is added to **Get next N devices**, **Get first N devices**, **Get last N devices**, **GET count of total available devices**, and **Close Cursor Id** API calls to fetch device record without device details.

Note:

if hideDetails = true then Response will not contains the device details.

if hideDetails=false then Response will contains the device details,

N maximum limit is 500.

Fetching devices with filter and without details	
URI	/api/devices?hideDetails=true
Method	GET

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	`200 OK`

Table continues…

Fetching devices with filter and without details	
Response Payload	PagingInfo which contains cursorId and total device records
Example	**Request**

```
GET /GuestManager/api/devices?
filterCriteria=name&op=startWith&val=device HTTP/1.1
Host: 10.120.120.30
api-version: v1.1.0
Authorization: Basic dGVzdDp0ZXN0
Accept:application/json
Content-Type: application/json
```

Response Payload

GET Cursor Id

```
{
  "PagingInfo": {
    "cursorId": "17303152153503406093",
    "totalRecord": 11
  }
}
```

GET next N devices

URL: https://GuestManager/api/devices/next/2/17303152153503406093?hideDetails=true

```
{
  "DeviceList": {
    "Device": [
      {
        "macAddress": "11:11:11:11:11:12"
      },
      {
        "macAddress": "11:11:11:11:11:13"
      }
    ]
  }
}
```

GET first N devices

URL: https://GuestManager/api/devices/first/2/17303152153503406093?hideDetails=true

```
{
  "DeviceList": {
    "Device": [
      {
        "macAddress": "22:22:22:22:22:22"
      },
      {
        "macAddress": "33:33:33:33:33:33"
      }
    ]
  }
}
```

GET last N devices

Table continues…

Fetching devices with filter and without details	

	URL: https://GuestManager/api/devices/last/2/17303152153503406093?hideDetails=true
	{
	  "DeviceList": {
	    "Device": [
	      {
	        "macAddress": "00:0a:95:9d:68:45"
	      },
	      {
	        "macAddress": "00:0a:95:9d:68:34"
	      }
	    ]
	  }
	}

API to query the status of single device

The API is used to query single device status to check whether the device exists, does not exists or expired.

API to query the status of single device	
URI	/api/devices/deviceStatusQuery/{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	`Status of a device` **Note:** The Status of a device could be as follows • FOUND - if device exists. • NOT_FOUND - if device does not exists. • FOUND_BUT_EXPIRED - if device exists but expired.

Example	**Request**
	`GET /GuestManager/api/devices/deviceStatusQuery/10:0b:01:20:00:06`
	`HTTP/1.1`
	`Host: 10.120.120.30`
	`api-version: v1.1.0`
	`Authorization: Basic dGVzdDp0ZXN0`

Table continues…

API to query the status of single device	

	Accept:application/json
	Cache-Control: no-cache
	
	Response Payload
	
	{
	        "Device": {
	                "macAddress": "10:0b:01:20:00:06",
	                “status”: “NOT_FOUND”
	        }
	}

API to query the status of multiple devices

The API is used to query single device status to check whether the device exists, does not exists or expired.

API to query the status of multiple devices	
URI	/api/devices/deviceStatusQuery?macs=mac1|mac2|mac3 **Note:** **|** is a separator between MAC Addresses, maximum 100 MAC addresses can be passed in a query parameter.
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 device status

Example	**Request**
	
	GET /GuestManager/api/devices/deviceStatusQuery?
	macs=12:00:00:00:00:01|12:00:00:00:00:02|12:00:00:00:00:03|
	12:00:00:00:00:04:00:00 HTTP/1.1
	Host: 10.120.120.30
	api-version: v1.1.0
	Authorization: Basic dGVzdDp0ZXN0
	Accept:application/json
	Cache-Control: no-cache
	
	Response Payload
	
	{
	    "DeviceList": {
	        "Device": [

Table continues…

API to query the status of multiple devices

```
            {
                "macAddress": "12:00:00:00:00:01",
                “status”: “NOT_FOUND”
            },
            {
                "macAddress": "12:00:00:00:00:02",
                “status”: “FOUND”
            },
          {
                "macAddress": "12:00:00:00:00:03",
                “status”: “FOUND_BUT_EXPIRED”
            },
          {
                "macAddress": "12:00:00:00:00:04:00:00",
                “status”: “INVALID_MACADDRESS”
            }
        ]
    }
}
```

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`
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",`

Table continues…

Guest User Registration REST API

```
"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
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"
}
```

Table continues…

Guest User Registration REST API

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.
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

Table continues…

Attribute	Type/Value	Description
		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 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

Table continues…

Attribute	Type/Value	Description
		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. 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

Table continues…

Attribute	Type/Value	Description
		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.
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	
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.

Table continues…

Error Case	Response Code	Error Response
Email Notification disabled	400 Bad Request	errorCode: NOTIFICATION_ERROR msg: Error: Could not send the notification. Please contact your administrator.

Update a Guest User

The API is used to update a Guest User for a user name.

Note:

Error response is sent if Guest User account is expired. If the Guest User account is expired, delete the account and recreate it.

Provisioning Group Name associated with this Guest User cannot be modified. If this information is passed in the API, it will be skipped.

Update a Guest User	
URI	/api/guestUsers/{username}
Method	PUT

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	`200 OK`
Response Payload	`Guest User record updated successfully`

Example	**Request**
	`PUT /GuestManager/api/guestUsers/guestUser1 HTTP/1.1`
	`Host: 10.120.120.30`
	`api-version: v1.1.0`
	`Authorization: Basic dGVzdDp0ZXN0`
	`Accept:application/json`
	`Content-type: application/json`
	Request Payload
	JSON
	`{`
	`   "GuestUser" : {`
	`      "firstName" : "fName1",`

Table continues…

Update a Guest User

```
      "lastName" : "lName1",
      "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

```
<?xml version="1.0" encoding="UTF-8"?>
<GuestUser>
   <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 Payload

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

For more information about the variable definitions, see Variable definition on page 77.

Error Cases

The following table describes the error cases for updating Guest User for a username.

Error Case	Response Code	Error Response
Guest User does not exists	404 not found	
Guest User access denied	400 Bad Request	errorCode: GUEST_USER_ACCESS_DENIED msg: Your account does not have permission to access the Guest User: <username>.

Table continues…

Error Case	Response Code	Error Response
		If Provisioner can access each other's record with this group is true then allow to edit the device by other provisioner and provisionedBy will be updated.
Guest User record already expired	400 Bad Request	errorCode: GUEST_USER_EXPIRED msg: Guest User already expired.

Delete a Guest User

The API is used to delete a single Guest User for a username.

Delete a Guest User	
URI	/api/ guestUsers/{username}
Method	DELETE

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	`200 OK`
Response Payload	`Guest User record deleted successfully`

Example	**Request**
	`DELETE /GuestManager/api/guestUsers/guestUser2 HTTP/1.1`
	`Host: 10.120.120.30`
	`api-version: v1.1.0`
	`Authorization: Basic dGVzdDp0ZXN0`
	`Accept: application/json`
	`Content-type: application/json`

Error Cases

The following table describes the error cases for deleting a Guest User for a username.

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

Table continues…

Error Case	Response Code	Error Response
Guest User access denied	400 Bad Request	errorCode: GUEST_USER_ACCESS_DENIED msg: Your account does not have permission to delete the Guest User: {0}. If Provisioner can access each other's record with this group is true then allow to delete the guest user.

Deleting multiple Guest Users

The API is used to delete multiple Guest Users.

Note:

Maximum 500 Guest Users can be sent to delete.

Deleting multiple Guest Users	
URI	/api/guestUsers
Method	DELETE

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	`200 OK`
Response Payload	`List of usernames`

Example	**Request**
	`DELETE /GuestManager/api/guestUsers HTTP/1.1`
	`Host: 10.120.120.30`
	`api-version: v1.1.0`
	`Authorization: Basic dGVzdDp0ZXN0`
	`Accept:application/json`
	`Content-type: application/json`
	Request Payload
	`{`
	`    "GuestUserList": {`

Table continues…

Deleting multiple Guest Users

```
        "GuestUser": [
            {
                "userName": "user1"
            },
            {
                "userName": "user2"
            },
            {
                "userName": "user3"
            },
            {
                "userName": "user4"
            }]
    }
}
```

Response Payload

If all Guest Users are deleted successfully

```
{
   Message:” All Guest Users are deleted successfully.”
     successList:{ "GuestUser": [
            {
                "userName": "user1"
            },
            {
                "userName": "user2"
            },
            {
                "userName": "user3"
            },
            {
                "userName": "user4"
            }]
    }
}
```

If Partial delete success

```
{
   Message:” Guest Users are deleted partially, please check the
successList and failedList for detail”
     successList:{ "GuestUser": [
            {
                "userName": "user1"
            },
            {
                "userName": "user2"
            } ]
    }
    failsList:{ "GuestUser": [
            {
                "userName": "user3"
                “reason”: ERROR-RecordNotFound
            },
            {
                "userName": "user4"
                “reason”: ERROR-AccessDenied
            } ]
    }
}
```

Bulk Delete of Guest Users for a Provisioner

The API is used to delete all Guest User (bulk delete) of a provisioner.

Note:

At a time maximum 2000 records can be deleted, if records are more than 2000 then repeat the operation.

Bulk Delete of Guest Users for a Provisioner	
URI	/api/guestUsers/bulkDelete?hideDeleteDetails=true **Note:** hideDeleteDetails is a optional parameter. If hideDeleteDetails is false, response will contain deleted Guest User details. If hideDeleteDetails is true, response will contain no deleted Guest User details. If hideDeleteDetails is not passed as argument, then response will contain deleted Guest User details.
Method	DELETE

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	`200 OK`
Response Payload	`List of user names for deleted Guest Users`

Example	**Request**
	`DELETE /GuestManager/api/guestUsers/bulkDelete?`
	`hideDeleteDetails=false HTTP/1.1`
	`Host: 10.120.120.30`
	`api-version: v1.1.0`
	`Authorization: Basic dGVzdDp0ZXN0`
	`Accept: application/json`
	`Content-type: application/json`
	`Cache-Control: no-cache`
	Response Payload
	If Guest Users are more than the limit (2000)
	`{`
	`   Message:” First 2000 Guest Users are deleted successfully.`
	`     successList:{ "GuestUser": [`
	`            {`
	`                "userName": "user1"`
	`            },`
	`            {`

Table continues…

Bulk Delete of Guest Users for a Provisioner

```
                    "userName": "user2"
                } ]
        }
}
```

If hidedeleteDetails is true

```
{
   Message:” First 2000 Guest Users are deleted successfully.
}
```

Note:

If repeatRequired is true, then more than 2000 records exist. Repeat the operation to delete all records.

If Guest Users are less than or equal to the limit (2000)

```
{
   Message:” All Guest Users are deleted successfully.”
     successList:{ "GuestUser": [
                {
                    "userName": "user1"
                },
                {
                    "userName": "user2"
                } ]
        }
}
```

If hidedeleteDetails is true

```
{
   Message:” All Guest Users are deleted successfully.”
}
```

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`

Table continues…

Fetching Guest User details by username for a Provisioner	
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

JSON Format

```
{
    "GuestUser": {
        "userName": "guestUser1",
        "firstName": "guest",
        "lastName": "User1",
        "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>
    <firstName>guest</firstName>
    <lastName>User1<lastName>
    <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 the guest user account.
firstName	String	Specifies the first name of the guest user.

Table continues…

Attribute	Type/Value	Description
lastName	String	Specifies the last name of the guest user.
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 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 79.
2. Get next N Guest Users. For more information, see GET next N Guest Users on page 79.
3. Get first N Guest Users. For more information, see GET first N Guest Users on page 80.
4. Get last N Guest Users. For more information, see GET last N Guest Users on page 81.
5. Get count of total available Guest User records. For more information, see GET count of total available Guest User records on page 83.
6. Close Cursor Id. For more information, see Close Cursor Id on page 83.

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`
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`

Table continues…

<table>
<tr><th colspan="2">GET next N Guest Users</th></tr>
<tr><td><b>Response Code</b></td><td><code>200 OK</code></td></tr>
<tr><td><b>Response Payload</b></td><td><code>Guest User List</code></td></tr>
<tr><td><b>Example</b></td><td>

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",
                "firstName": "guest",
                "lastName": "User1",
                "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",
                "firstName": "Guest",
                "lastName": "User2",
                "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"
            }
        ]
    }
}
```

</td></tr>
</table>

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)`

Table continues…

GET first N Guest Users	

	`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": [
            {
                "userName": "GuestUser1",
                "firstName": "Guest",
                "lastName": "User1",
                "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",
                "firstName": "Guest",
                "lastName": "User2",
                "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 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/2/13666304570298546472 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": "GuestUser10",
                "firstName": "Guest",
                "lastName": "User10",
                "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",
                "firstName": "Guest",
                "lastName": "User9",
                "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"
            }
        ]
    }
}
```

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**
	`GET /GuestManager/api/guestUsers/count/13666304570298546472 HTTP/1.1`
	`Host: 10.120.120.30`
	`api-version: v1.0`
	`Authorization: Basic dGVzdDp0ZXN0`
	`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`

Table continues…

Close Cursor Id	
Response Payload	NA

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`

Fetching Guest User with filter

The API is used to fetch guest users with filter iteratively.

Get cursor Id is the first API call to specify filter criteria and get the paging information that has cursor Id that are used in subsequent calls to get the Guest Users iteratively.

The other API calls, **Get Next N Guest Users**, **Get first N Guest Users**, **Get last N Guest Users**, **GET count of total available Guest Users**, and **Close Cursor Id** are same as in *Fetching Guest Users iteratively for a Provisioner* section. For more information, see Fetching Guest Users iteratively for a Provisioner on page 78.

Fetching Guest User with filter	
URI	/api/guestUsers?filterCriteria=<field>&op=<op value>&val=<value>
Method	GET

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	`200 OK`
Response Payload	`PagingInfo which contains cursorId and total user records`

Example	**Request**
	`GET /GuestManager/api/guestUsers?`
	`filterCriteria=userName&op=startWith&val=User HTTP/1.1`
	`Host: 10.120.120.30`
	`api-version: v1.1.0`
	`Authorization: Basic dGVzdDp0ZXN0`
	`Accept:application/json`
	`Content-type: application/json`
	Response Payload

Table continues…

Fetching Guest User with filter

GET Cursor Id

```
{
  "PagingInfo": {
    "cursorId": "9854343005721964640",
    "totalRecord": 10
  }
}
```

GET next N Guest Users

URL: https://GuestManager/api/guestUsers/next/2/9854343005721964640

```
{
    "GuestUserList": {
        "GuestUser": [
            {
                "userName": "GuestUser1",
                "firstName": "guest",
                "lastName": "User1",
                "email": "testl2@avaya.com",
                "smsAddress": "9845342309@T-Mobile",
                "startTime": "2015/11/09 04:16:41 PM IST",
                "endTime": "2015/11/90 08:16:41 PM IST",
                "provisioningGroup": "p-api-user-device",
                "provisioner": "Internal/pall",
                "guestDetails": "guest Details"
            },
            {
                "userName": "GuestUser2",
                "firstName": "Guest",
                "lastName": "User2",
                "email": "test884@avaya.com",
                "smsAddress": "9622000000@tmomail.net",
                "startTime": "2015/11/09 04:16:41 PM IST",
                "endTime": "2015/11/10 12:16:41 AM IST",
                "provisioningGroup": "pg-fl-no",
                "provisioner": "Internal/pall",
                "guestDetails": "guest Details"
            }
        ]
    }
}
```

GET first N Guest Users

URL: https://GuestManager/api/guestUsers/first/2/9854343005721964640

```
{
    "GuestUserList": {
        "GuestUser": [
            {
                "userName": "GuestUser1",
                "firstName": "Guest",
                "lastName": "User1",
                "email": "testl2@avaya.com",
                "smsAddress": "9845342309@T-Mobile",
                "startTime": "2015/11/09 04:16:41 PM IST",
                "endTime": "2015/11/09 08:16:41 PM IST",
                "provisioningGroup": "p-api-user-device",
                "provisioner": "Internal/pall",
                "guestDetails": "guest Details"
```

Table continues…

Fetching Guest User with filter

```
            },
            {
                "userName": "GuestUser2",
                "firstName": "Guest",
                "lastName": "User2",
                "email": "test884@avaya.com",
                "smsAddress": "9622000000@tmomail.net",
                "startTime": "2015/11/09 04:16:41 PM IST",
                "endTime": "2015/11/10 12:16:41 AM IST",
                "provisioningGroup": "pg-fl-no",
                "provisioner": "Internal/pall",
                "guestDetails": "guest Details"
            }
        ]
    }
}
```

GET last N Guest Users

URL: https://GuestManager/api/guestUsers/last/2/9854343005721964640

```
{
    "GuestUserList": {
        "GuestUser": [
            {
                "userName": "GuestUser10",
                "firstName": "Guest",
                "lastName": "User10",
                "email": "test10@avaya.com",
                "smsAddress": "9123456789@tmomail.net",
                "startTime": "2015/11/10 04:16:41 PM GMT+00:00",
                "endTime": "2015/11/10 09:16:41 PM GMT+00:00",
                "provisioningGroup": "pg-api-user",
                "provisioner": "Internal/pall",
                "guestDetails": "guest Details-DL"
            },
            {
                "userName": "GuestUser9",
                "firstName": "Guest",
                "lastName": "User9",
                "email": "test9@avaya.com",
                "smsAddress": "9329393922@tmomail.net",
                "startTime": "2015/11/10 04:16:41 PM IST",
                "endTime": "2015/11/10 12:16:41 AM IST",
                "provisioningGroup": "pg-user-email-phone",
                "provisioner": "Internal/pall",
                "guestDetails": "guest Details"
            }
        ]
    }
}
```

Filter details for a Guest User

The following table describes the filter details for a Guest User.

Filter Criteria	Options	Value Type	Description
userName	equal notEqual startWith endsWith contains	String	Username
firstName	equal notEqual startWith endsWith contains	String	First name
lastName	equal notEqual startWith endsWith contains	String	Last name
email	equal notEqual startWith endsWith contains	String	Email
startTime	greaterThan greaterThanEqual lessThan lessThanEqual	Date Format: yyyy/MM/dd hh:mm:ss a z ex: 2015/09/08 10:10:22 AM IST	Start Time of Guest User account
endTime		Date Format: yyyy/MM/dd hh:mm:ss a z ex: 2015/09/08 10:10:22 AM IST	Expiry time of Guest User account
smsAddress	Equal notEqual	String	SMS Address

Fetching Guest Users with filter and without details

The API is used to fetch Guest Users with filter and without details.

The filter details are same as mentioned in the previous section. For more information, see Fetching Guest User with filter on page 84. Optional query parameter hideDetails=true is added to **Get next N Guest Users**, **Get first N Guest Users**, **Get last N Guest Users**, **GET count of total available Guest Users**, and **Close Cursor Id** API calls to fetch user record without user details.

Note:

if hideDetails = true then Response will not contains the Guest User details.

if hideDetails=false then Response will contains the Guest User details,

N maximum limit is 500.

Fetching Guest Users with filter and without details	
URI	/api/guestUsers?hideDetails=true
Method	GET

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	`200 OK`
Response Payload	`PagingInfo which contains cursorId and total guest user records`

Example	**Request**
	`GET /GuestManager/api/guestUsers?`
	`filterCriteria=userName&op=startWith&val=User HTTP/1.1`
	`Host: 10.120.120.30`
	`api-version: v1.1.0`
	`Authorization: Basic dGVzdDp0ZXN0`
	`Accept:application/json`
	`Content-Type: application/json`
	Response Payload
	GET Cursor Id
	`{`
	`  "PagingInfo": {`
	`    "cursorId": "4490890776062129399",`
	`    "totalRecord": 5`
	`  }`
	`}`
	GET next N Guest Users

Table continues…

Fetching Guest Users with filter and without details

URL: https://GuestManager/api/guestUsers/next/2/4490890776062129399?hideDetails=true

```
{
  "GuestUserList": {
    "GuestUser": [
      {
        "userName": "GuestUser1"
      },
      {
        "userName": "GuestUser2"
      }
    ]
  }
}
```

GET first N Guest Users

URL: https://GuestManager/api/guestUsers/first/2/4490890776062129399?hideDetails=true

```
{
  "GuestUserList": {
    "GuestUser": [
      {
        "userName": "GuestUser1"
      },
      {
        "userName": "GuestUser2"
      }
    ]
  }
}
```

GET last N Guest Users

URL: https://GuestManager/api/guestUsers/last/2/4490890776062129399?hideDetails=true

```
{
  "GuestUserList": {
    "GuestUser": [
      {
        "userName": "GuestUser10"
      },
      {
        "userName": "GuestUser9"
      }
    ]
  }
}
```

API to query the status of single user

The API is used to query the status of single user to check whether the user exists, does not exists or expired.

API to query the status of single user	
URI	/api/guestUsers/userStatusQuery/{userName}
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	Status of a Guest User **Note:** The Status of a user could be as follows • FOUND - if user exists. • NOT_FOUND - if user does not exists. • FOUND_BUT_EXPIRED - if user exists but expired.

Example	**Request**
	GET /GuestManager/api/guestUsers/userStatusQuery/user1 HTTP/1.1
	Host: 10.120.120.30
	api-version: v1.1.0
	Authorization: Basic dGVzdDp0ZXN0
	Accept:application/json
	Cache-Control: no-cache
	Response Payload
	{
	        "User": {
	                "userName": "user1",
	                "status": "NOT_FOUND"
	        }
	}

API to query the status of multiple users

The API is used to query the status of multiple users to check whether the user exists, does not exists or expired.

API to query the status of multiple users	
URI	/api/guestUsers/userStatusQuery?userNames=username1|username2|username3 **Note:** **|** is a separator between usernames, maximum 100 usernames can be passed in a query parameter.

Table continues…

API to query the status of multiple users	
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 user status`

Example

Request

```
GET /GuestManager/api/guestUsers/userStatusQuery?userNames=user1|
user2|user3 HTTP/1.1
Host: 10.120.120.30
api-version: v1.1.0
Authorization: Basic dGVzdDp0ZXN0
Accept:application/json
Cache-Control: no-cache
```

Response Payload

```
{
    "UserList": {
        "User": [
            {
                "userName": "user1",
                 “status”: “NOT_FOUND”
            },
            {
                "userName": "user2",
                “status”: “FOUND”
            },
          {
                "userName": "user3",
                “status”: “FOUND_BUT_EXPIRED”
            },
        ]
    }
}
```