Extreme™
Customer-Driven Networking

Identity Engines Guest & IoT Manager REST APIs

Release 9.5.0
9035377 Rev 01
October 2018

Legal Notice

Extreme Networks, Inc. reserves the right to make changes in specifications and other information contained in this document and its website without prior notice. The reader should in all cases consult representatives of Extreme Networks to determine whether any such changes have been made.

The hardware, firmware, software or any specifications described or referred to in this document are subject to change without notice.

Trademarks

Extreme Networks and the Extreme Networks logo are trademarks or registered trademarks of Extreme Networks, Inc. in the United States and/or other countries.

All other names (including any product names) mentioned in this document are the property of their respective owners and may be trademarks or registered trademarks of their respective companies/owners.

For additional information on Extreme Networks trademarks, please see: www.extremenetworks.com/company/legal/trademarks

Software Licensing

Some software files have been licensed under certain open source or third-party licenses. End-user license agreements and open source declarations can be found at: www.extremenetworks.com/support/policies/software-licensing

Support

For product support, phone the Global Technical Assistance Center (GTAC) at 1-800-998-2408 (toll-free in U.S. and Canada) or +1-408-579-2826. For the support phone number in other countries, visit: http://www.extremenetworks.com/support/contact/

For product documentation online, visit: https://www.extremenetworks.com/documentation/

Contents

Chapter 1: About this Document

Purpose

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

Conventions

This section discusses the conventions used in this guide.

Text Conventions

The following tables list text conventions that can be used throughout this document.

Table 1: Notice Icons

Icon	Alerts you to...
Important:	Key information that does not carry with it the risk of personal injury, death, system failure, service interruption, loss of data, damage to equipment, or electrostatic discharge.
Note:	Important features or instructions.
Tip:	Helpful tips and notices for using the product.
Warning:	A potential hazard exists that, if not avoided, can result in harm to hardware or equipment.
Caution:	Practices that are not safe or are potential hazards not covered by danger or warning messages.

Table 2: Text Conventions

Convention	Description
Angle brackets (< >)	Angle brackets (< >) indicate that you choose the text to enter based on the description inside the brackets. Do not type the brackets when you enter the command. If the command syntax is `cfm maintenance-domain maintenance-level <0-7>` , you can enter `cfm maintenance-domain maintenance-level 4`.
Bold text	Bold text indicates the GUI object name you must act upon. Examples: • Click **OK**. • On the **Tools** menu, choose **Options**.
Italic Text	Italics emphasize a point or denote new terms at the place where they are defined in the text. Italics are also used when referring to publication titles that are not active links.
`Plain Courier Text`	Plain Courier text indicates command names, options, and text that you must enter. Plain Courier text also indicates command syntax and system output, for example, prompts and system messages. Examples: • `show ip route` • `Error: Invalid command syntax [Failed][2018-09-12 13:37:03.303 -04:00]`
Separator (>)	A greater than sign (>) shows separation in menu paths. For example, in the Navigation tree, expand the **Configuration** > **Edit** folders.

Documentation and Training

To find Extreme Networks product guides, visit our documentation pages at:

Current Product Documentation	www.extremenetworks.com/documentation/
Archived Documentation (for earlier versions and legacy products)	www.extremenetworks.com/support/documentation-archives/
Release Notes	www.extremenetworks.com/support/release-notes
Hardware/Software Compatibility Matrices	https://www.extremenetworks.com/support/compatibility-matrices/
White papers, data sheets, case studies, and other product resources	https://www.extremenetworks.com/resources/

Open Source Declarations

Some software files have been licensed under certain open source licenses. More information is available at: www.extremenetworks.com/support/policies/open-source-declaration/.

Training

Extreme Networks offers product training courses, both online and in person, as well as specialized certifications. For more information, visit www.extremenetworks.com/education/.

Getting Help

If you require assistance, contact Extreme Networks using one of the following methods:

- **GTAC (Global Technical Assistance Center) for Immediate Support**
  - **Phone:** 1-800-998-2408 (toll-free in U.S. and Canada) or +1 408-579-2826. For the support phone number in your country, visit: www.extremenetworks.com/support/contact
  - **Email:** support@extremenetworks.com. To expedite your message, enter the product name or model number in the subject line.
- **Extreme Portal** — Search the GTAC knowledge base, manage support cases and service contracts, download software, and obtain product licensing, training, and certifications.
- **The Hub** — A forum for Extreme Networks customers to connect with one another, answer questions, and share ideas and feedback. This community is monitored by Extreme Networks employees, but is not intended to replace specific guidance from GTAC.

Before contacting Extreme Networks for technical support, have the following information ready:

- Your Extreme Networks service contract number and/or serial numbers for all involved Extreme Networks products
- A description of the failure
- A description of any action(s) already taken to resolve the problem
- A description of your network environment (such as layout, cable type, other relevant environmental information)
- Network load at the time of trouble (if known)
- The device history (for example, if you have returned the device before, or if this is a recurring problem)
- Any related RMA (Return Material Authorization) numbers

Subscribing to Service Notifications

You can subscribe to email notifications for product and software release announcements, Vulnerability Notices, and Service Notifications.

1. Go to www.extremenetworks.com/support/service-notification-form.

2. Complete the form with your information (all fields are required).
3. Select the products for which you would like to receive notifications.

   Note:
   You can modify your product selections or unsubscribe at any time.
4. Click **Submit**.

Providing Feedback to Us

Quality is our first concern at Extreme Networks, and we have made every effort to ensure the accuracy and completeness of this document. We are always striving to improve our documentation and help you work better, so we want to hear from you! We welcome all feedback but especially want to know about:

- Content errors or confusing or conflicting information.
- Ideas for improvements to our documentation so you can find the information you need faster.
- Broken links or usability issues.

If you would like to provide feedback to the Extreme Networks Information Development team, you can do so in two ways:

- Use our short online feedback form at https://www.extremenetworks.com/documentation-feedback/.
- Email us at documentation@extremenetworks.com.

Please provide the publication title, part number, and as much detail as possible, including the topic heading and page number if applicable, as well as your suggestions for improvement.

Chapter 2: New in this Document

There is no feature changes in *Identity Engines Guest and IoT Manager REST APIs document* for Release 9.5.0. A few images in the document are updated to reflect the latest UI changes.

Chapter 3: Guest and IoT 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. The Guest and IoT 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 and IoT Manager REST API uses JSON and XML as its communication format, and the standard HTTP methods like GET, PUT, POST and DELETE.

Chapter 4: Guest and IoT Manager REST API Initial Setup

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

Setting up Guest and IoT Manager REST API

Use this procedures in sequence to enable the Guest and IoT Manager REST APIs.

1. Create Provisioning Group in Guest and IoT Manager (GIM). For more information, see Creating Provisioning Group in Guest and IoT Manager on page 11
2. Create Provisioner in GIM. For more information, see Creating Provisioner and Associating it with Provisioning Groups on page 12
3. Download and Install RESTClient plugin in Firefox. For more information, see Downloading and Installing Firefox RESTClient Plugin on page 12

Creating Provisioning Group in Guest and IoT Manager

Use this procedure to create a Provisioning Group in Guest and IoT Manager.

Procedure

1. In a support web browser, enter the Guest and IoT 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 *Identity Engines Guest and IoT Manager Configuration document*.

Creating Provisioner and Associating it with Provisioning Groups

Use this procedure to create a Provisioner to associate it with the Provisioner Groups in Guest and IoT Manager.

Procedure

1. Login to Guest and IoT 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 *Identity Engines Guest and IoT Manager Configuration document*.

Downloading and Installing Firefox RESTClient Plugin

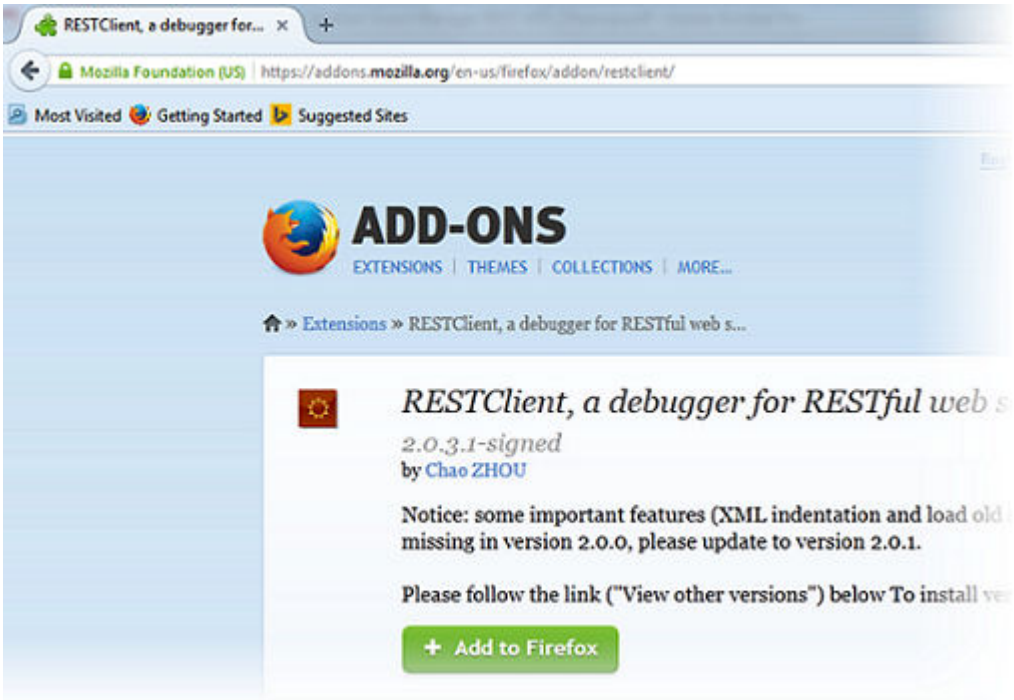
This section describes the procedure to download and install the Firefox RESTClient plugin.

Procedure

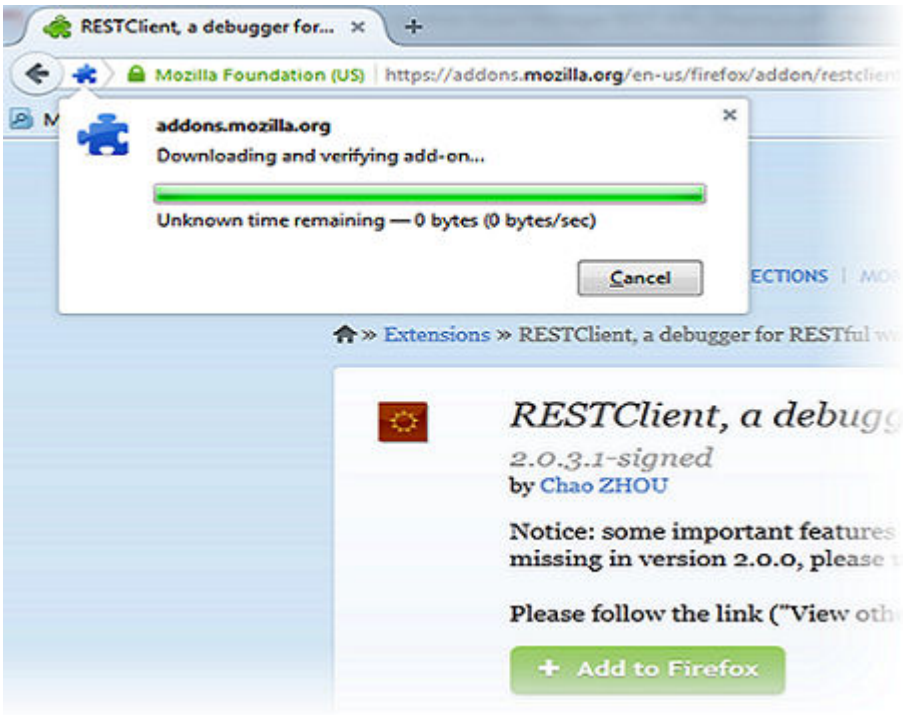
1. Download and Install the Firefox RESTClient plugin from the following URL:

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

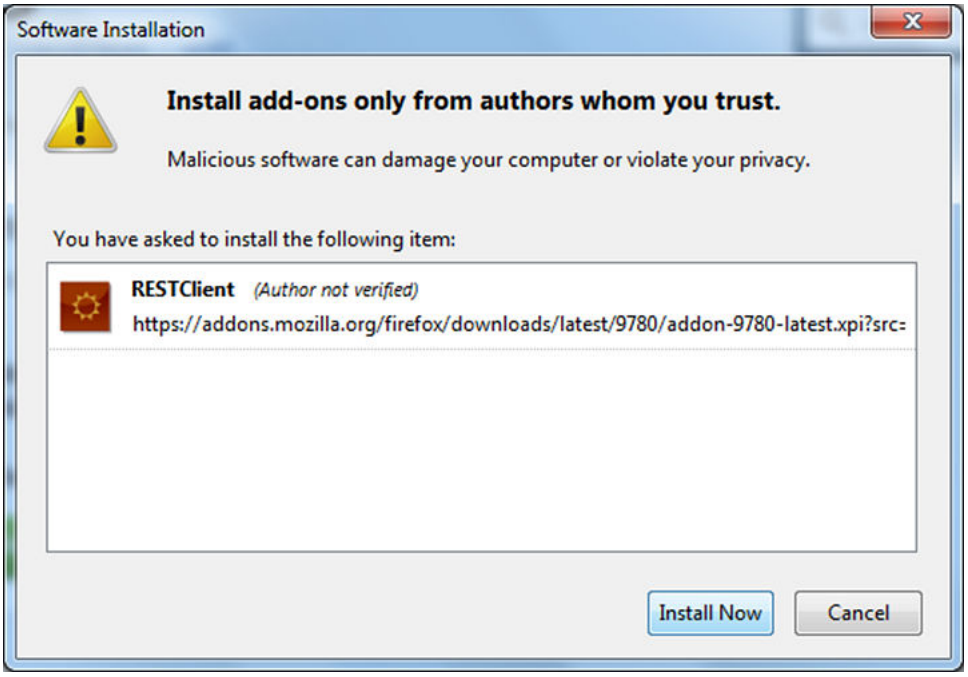
2. Click **+ Add to Firefox**.



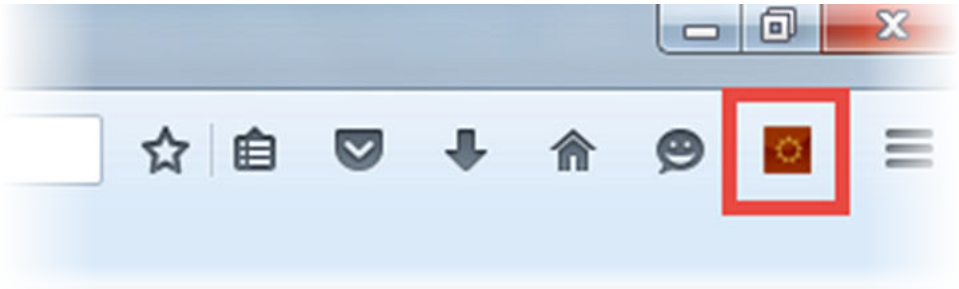
The plugin gets downloaded and verified and Software Installation window is displayed.



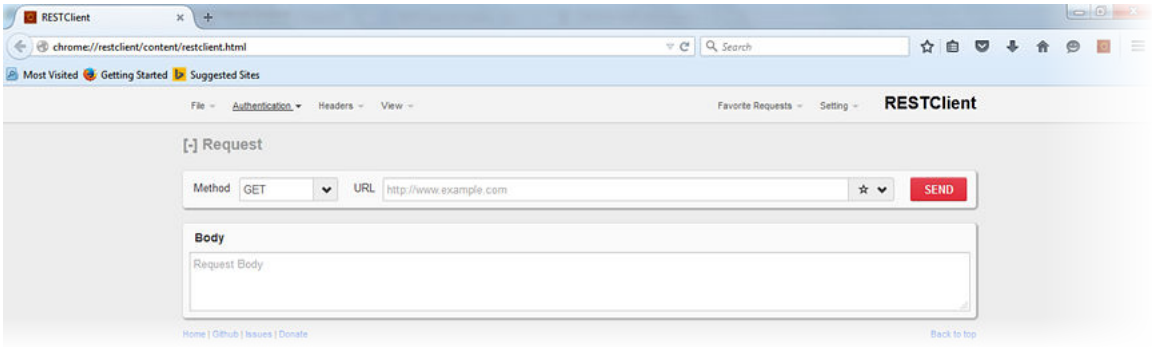
3. Click **Install Now**.



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



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



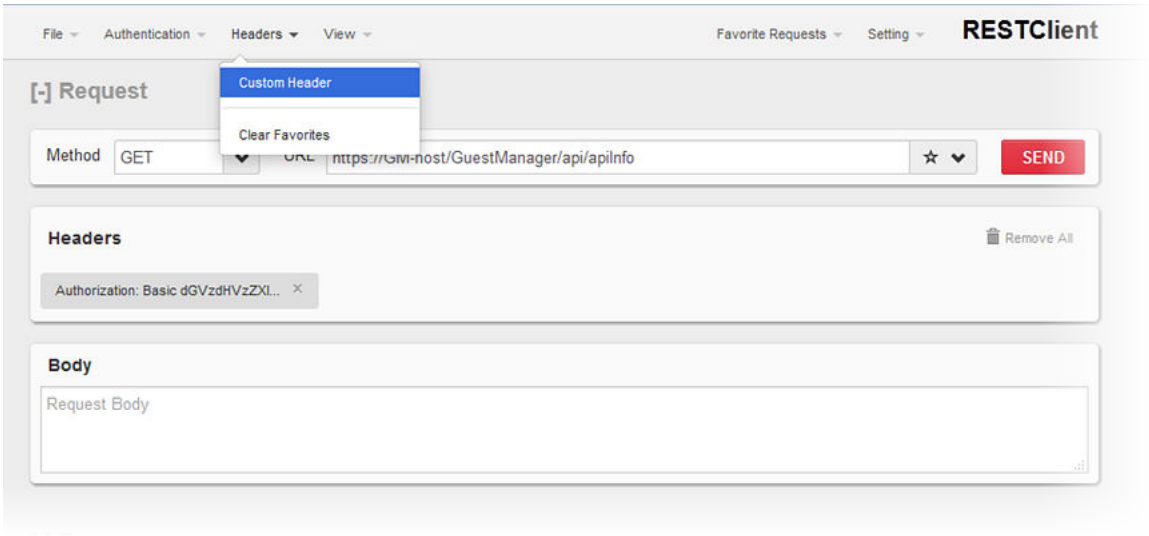
Guest and IoT Manager REST API Version

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 v2.0 (**api-version:v2.0**). In this Release the GM is compatible with REST API versions v1.0, v1.1.0 and v2.0.

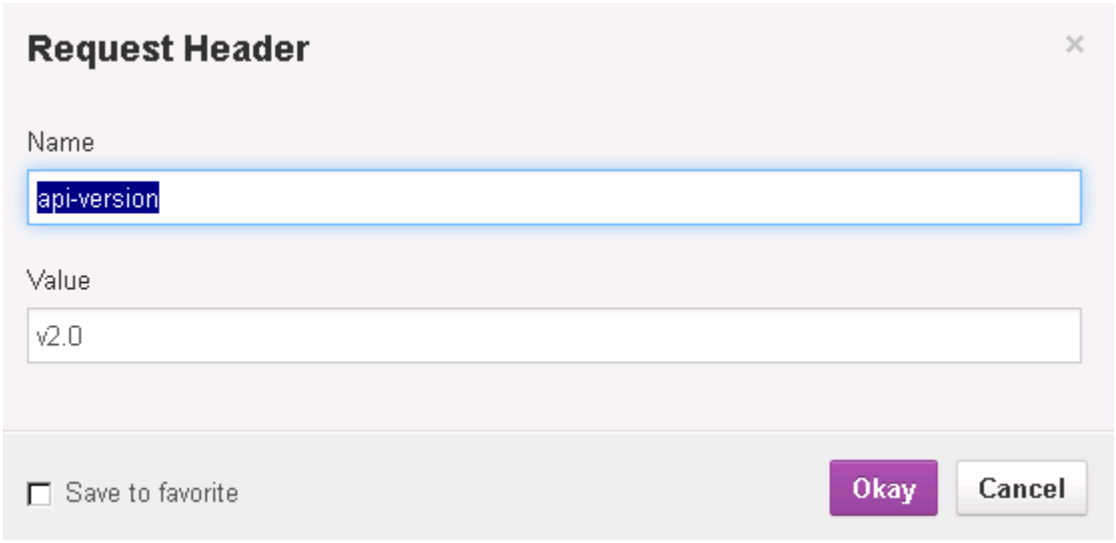
Use this procedure to add version in the **HTTP Headers**.

Procedure

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

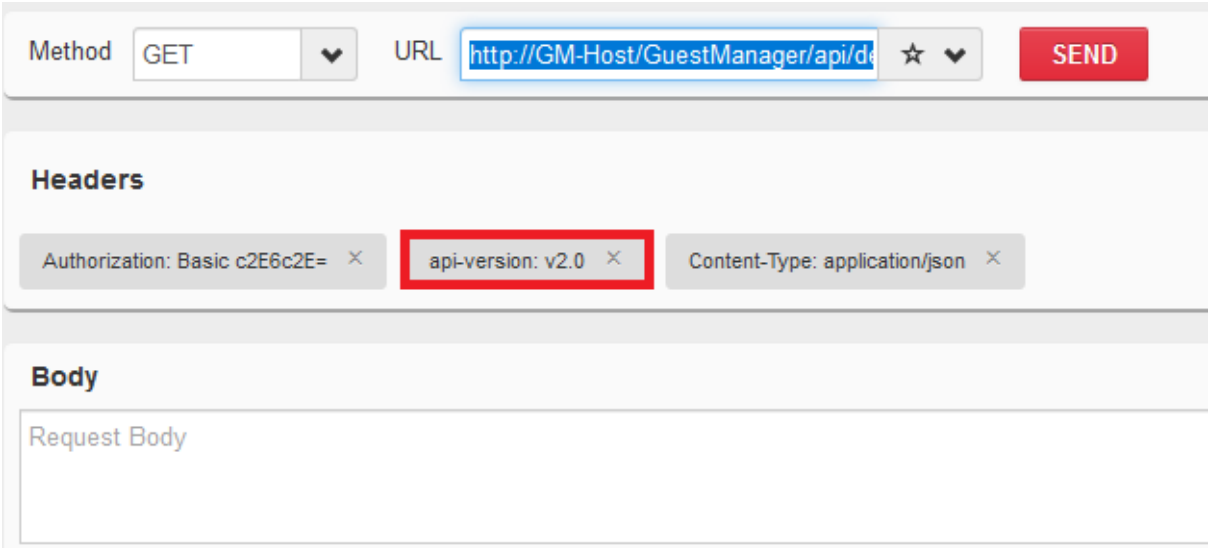


The Request Header window is displayed.



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

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



Authorization

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

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

Authorization: **username:password**

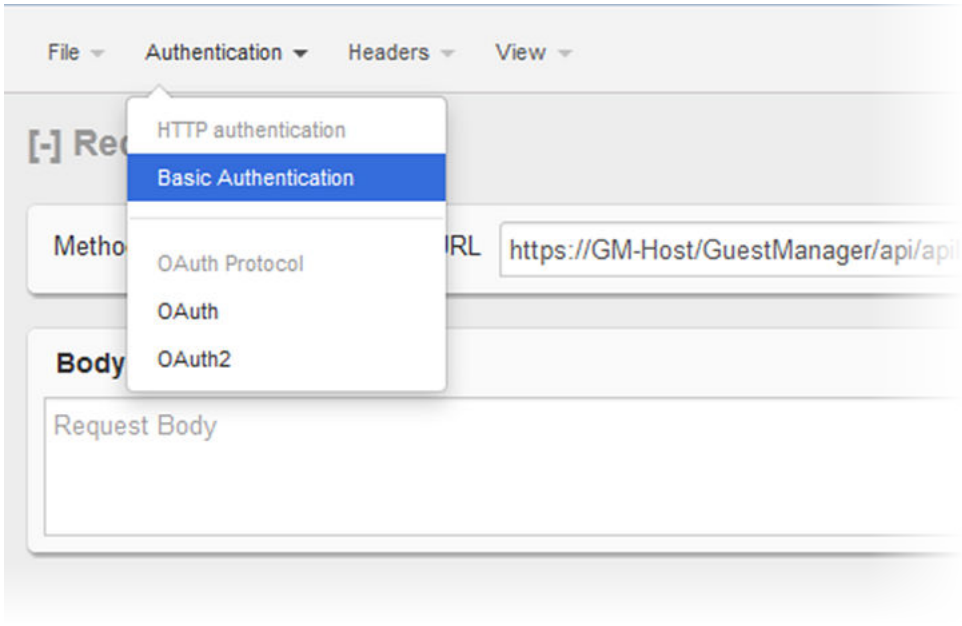


Figure 1: Basic Authentication

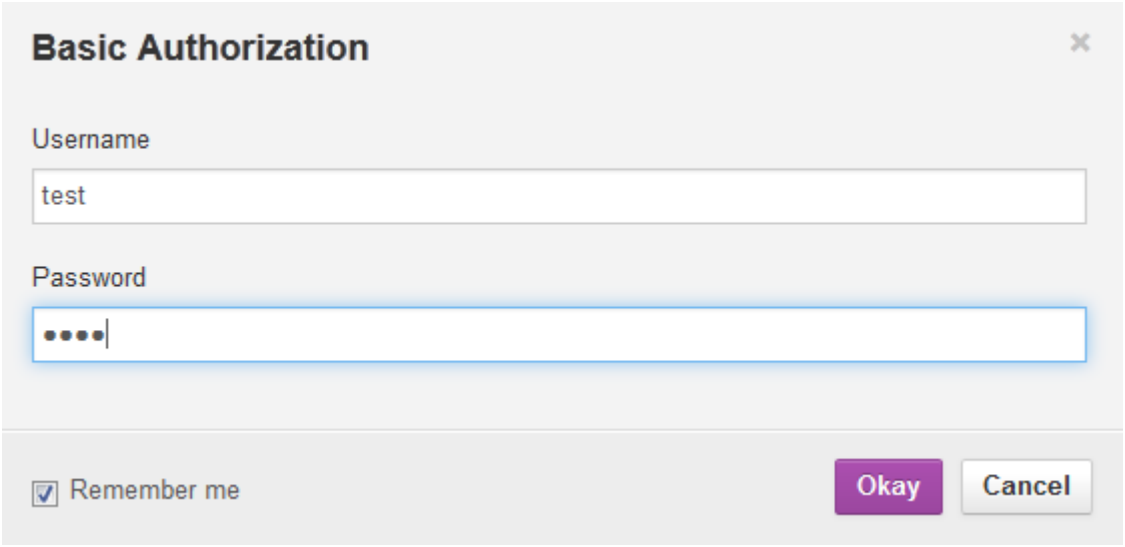


Figure 2: Basic Authorization

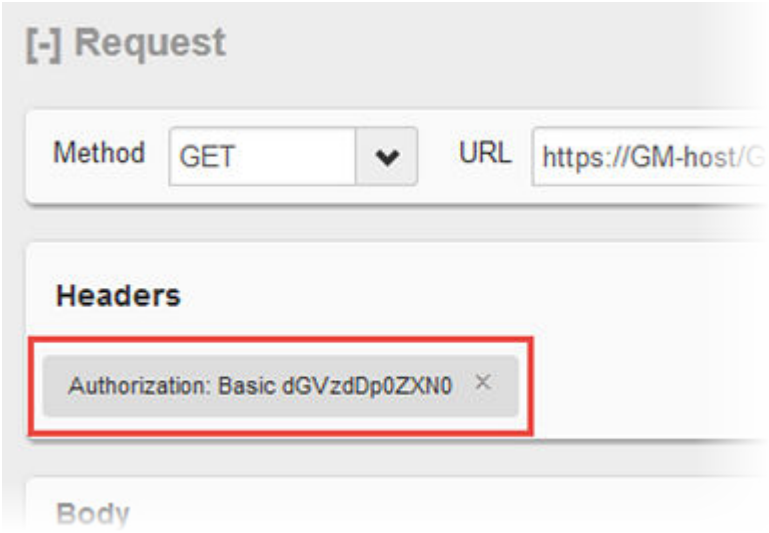


Figure 3: Authorization Header

Guest and IoT 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 and IoT Manager API info	
URI	/api/apiInfo
HTTP Header	Accept: application/json
Response	The Format of response preview can be XML or JSON. 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: 192.0.2.1`
	`Accept:application/json`
	`Cache-Control: no-cache`
	Response
	JSON Format
	`{`
	`"apiPath": "/api",`
	`"name": "Ignition Guest & IoT Manager REST API",`
	`"productName": "Identity Engines Ignition Guest & IoT Manager",`
	`"vendor": "Extreme Networks.",`
	`"version": "v2.0"`
	`}`

Note:

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

For example: If Guest and IoT Manager base URL is https://192.0.2.1/GuestManager then API info URI is https://192.0.2.1/ GuestManager/api/apiInfo.

Common Error Cases

The following table describes the Common Error Cases.

Error Case	Response Code	Error Respone
Authorization Header missing	401 Unauthorized	errorCode: AUTHORIZATION_REQUIRED msg: Authorization required.
Missing version	406 Not Acceptable	erorCode: VERSION_REQUIRED

Table continues…

Error Case	Response Code	Error Respone
		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 and IoT Manager is not connected with Ignition Server	500 Internal Server Error	errorCode: IGM_NOT_CONNECTED_WITH_IGS msg: Ignition Guest and IoT 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 (not reachable) Ignition Server not reachable	503 Service Unavailable	errorCode: RADIUS_ERROR msg: Radius server error <error msg>
Provisioning Group is not accessible/invalid	400 Bad Request	errorCode: PROVISIONING_GROUP_ACCESS_DENIED msg: Your account does not have permission to access the Provisioning Group: {group name}

Table continues…

Error Case	Response Code	Error Respone
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 and IoT Manager REST APIs

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

Fetching Provisioning Group for a Provisioner

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

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

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

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

Example	**Request**
	`GET /GuestManager/api/provisioningGroups HTTP/1.1`
	`Host: 192.0.2.1`
	`api-version: v2.0`
	`Authorization: Basic dGVzdDp0ZXN0`
	`Accept:application/json`
	`Cache-Control: no-cache`
	Response Payload
	JSON Format
	`{`
	`"ProvisioningGroups": {`
	`"groupName": [`
	`"api-device-provGroup",`
	`"api-device-provGroup1",`

Table continues…

Fetching Provisioning group for a Provisioner

```
                "api-device-provGroup2"
            ]
        }
}
```

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.

Note:

The Provisioning group name contains alphanumeric / special characters and space in between words. For example, use only these special characters: `# =()_-.! [].`

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`
New Parameters	networkRights, accessTypes, accessZones, deleteOnExpire, networkAccessRights, accessibleTypesSubTypes, type, subType, assetType, assetTypeDefault, deleteOnExpire, and customAttributes

Table continues…

Fetching Provisioning Group Details for Group name

Example

Request

```
GET /GuestManager/api/provisioningGroupDetails/pg-api-user-device
HTTP/1.1
Host: 192.0.2.1
api-version: v2.0
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.

JSON Format

```
{
    "ProvisioningGroup": {
        "groupName": "api-device!-provGroup#",
        "maxDuration": 8,
        "durationUnit": "HOURS",
        "timezone": "Asia/Calcutta",
        "guestUserAllowed": true,
        "devicesAllowed": true,
        "networkRights": "[IT,sales]",
        "accessTypes": "[wired, wireless]",
        "accessZones": "[Groundfloor, Firstfloor]",
        "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,
            "deleteOnExpire" : true,
            "networkAccessRights" : true,
        },
        "devicesDetails": {
            "nameAccessible": true,
            "nameRequired": false,
            "typeAccessible": true,
            "typeRequired": false,
            "subTypeAccessible": true,
            "subTypeRequired": false,
            "accessibleTypesSubtypes": [
             {
               "type": "FA client",
               "subTypes": [
                 "ONA-SDN",
                 "ONA-SPBoIP",
                 "wlan-9100",
                 "n/a"
               ]
             },
```

Table continues…

Fetching Provisioning Group Details for Group name

```
                {
                  "type": "fax machine",
                     "subTypes": "n/a"
                    }
                       ],
                 "assetType": true,
                 "assetTypeDefault": "TEMPORARY",
                 "deleteOnExpire": true,
                 "networkAccessRights": true,
                 "customAttributes": 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>
    <networkRights>[IT, sales]</networkRights>
    <accessTypes>[wired, wireless]</accessTypes>
    <accessZones>[Groundfloor, Firstfloor]</accessZones>
    <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>
            <deleteOnExpire>true</deleteOnExpire>
            <networkAccessRights>true</networkAccessRights>
    </guestUserDetails>
    <devicesDetails>
        <nameAccessible>true</nameAccessible>
        <nameRequired>false</nameRequired>
        <typeAccessible>true</typeAccessible>
        <typeRequired>false</typeRequired>
        <subTypeAccessible>true</subTypeAccessible>
        <subTypeRequired>false</subTypeRequired>

   <accessibleTypesSubtypes>
        <type>FA client</type>
        <subTypes>ONA-SDN</subTypes>
        <subTypes>ONA-SPBoIP</subTypes>
        <subTypes>wlan-9100</subTypes>
        <subTypes>n/a</subTypes>
```

Table continues…

Fetching Provisioning Group Details for Group name

```
            </accessibleTypesSubTypes>
            <accessibleTypesSubTypes>
        <type>fax machine</type>
        <subTypes>n/a</subTypes>
            </accessibleTypesSubTypes>
     <assetType>true</assetType>
     <assetTypeDefault>PERMANENT</assetTypeDefault>
     <deleteOnExpire>true</deleteOnExpire>
     <networkAccessRights>true</networkAccessRights>
     <customAttributes>false</customAttributes>
   </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,
```

Table continues…

Fetching Provisioning Group Details for Group name

```
                "subTypeAccessible": true,
                "subTypeRequired": false
            }
        }
}
```

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.
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.
networkRights	String	Specifies the network rights configured for a provisioning group by an admin in the Ignition Server Dashboard.
accessTypes	String	Specifies the access type configured for a provisioning group by an admin in the Ignition Server Dashboard.
accessZones	String	Specifies the access zone configured for a provisioning group by an admin in the Ignition Server Dashboard.

Table continues…

Attribute	Type/Value	Description
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.
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.

Table continues…

Attribute	Type/Value	Description
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. • if accountValidityDurationAccessible is false, **Permanent** option is enabled. You can set the account validity duration either to default Provisioning group value (Max Validity Duration) or Permanent. You can enter the number of minutes, hours or days in the **Provisioners Groups** > **Common** > **Temporary accounts may be valid for up to** field. If Permanent is set, the Guest User account is not expired.
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.
deleteOnExpire	Boolean	Specifies whether the deleteOnExpire must be sent in the request and response. Parameters and values are: • if deleteOnExpire is true, then deleteOnExpire is sent. • if deleteOnExpire is false, then deleteOnExpire is not sent.
networkAccessRights	Boolean	Specifies whether the networkAccessRights must be provided in the request and response. Parameters and values are: • if networkAccessRights is true, then networkRights, accessTypes, and accessZones are sent. • if networkAccessRights is true, then networkRights, accessTypes, and accessZones are 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. • if typeAccessible is false, then Provisioner cannot configure device type.

Table continues…

Attributes	Type/Value	Description
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.
accessibleTypesSubTypes	Object	Specifies the array of type and subtype.
type	String	Specifies the type of device.
subType	Object	Specifies the array of device subtypes.
assetType	Boolean	Specifies whether the assetType must be sent in the request and response. Parameters and values are: • if assetType is true, then assetType is sent. • if assetType is false, then assetType is not sent.
assetTypeDefault	String	Specifies the value set in the provisioning group by admin. The default value is TEMPORARY.
deleteOnExpire	Boolean	Specifies whether the deleteOnExpire must be sent in the request and response. Parameters and values are: • if deleteOnExpire is true, then deleteOnExpire is sent. • if deleteOnExpire is false, then deleteOnExpire is not sent.
networkAccessRights	Boolean	Specifies whether the networkAccessRights must be provided in the request and response. Parameters and values are: • if networkAccessRights is true, then networkRights, accessTypes, and accessZones are sent. • if networkAccessRights is true, then networkRights, accessTypes, and accessZones are not sent.

Table continues…

Attributes	Type/Value	Description
customAttributes	Boolean	Specifies whether the customAttributes must be sent in the request and response. Parameters and values are: • if customAttributes is true, then custom 1 to custom 5 fields are sent. • if customAttributes is true, then custom 1 to custom 5 fields are not sent.

Device Operations

This section describes CRUD operations with respect to Device.

Device Registration

The API allows Provisioner to add devices to the Guest and IoT 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`
	`Content-Type: application/json or application/xml`

Response Code	`201 (created)`
Response Payload	N/A
New Parameters	enabled, assetType, startDate, endDate, durationUnit, duration, deleteOnExpire, networkRights, accessTypes, accessZones, custom1, custom2, custom3, custom4, custom5, and comments

Example	**Request**
	`POST /GuestManager/api/devices HTTP/1.1`
	`Host: 192.0.2.1`
	`api-version: v2.0`
	`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":`

Table continues…

Device Registration REST API

```
"mobile", "subType": "generic-android", "vlanLabel": "vlan-100",
"vlanId": "100", "enabled": "true", "assetType": "TEMPORARY",
"startDate" : "2016/11/10 10:30:41", "endDate" : "2016/11/10
15:30:41", "durationUnit" : "HOURS", "duration" : 5,
"deleteOnExpire":"true", "networkRights": IT, "accessTypes": "[Wired,
Wireless]", "accessZones": "[Ground-Floor-Left-Wing, Ground-Floor-
Right-Wing]", "custom1": "text1", "custom2": "text2", "custom3":
"text3", "custom4": "text4", "custom5": "text5", "comment": "test
device create"} }
```

Response Header

```
Content-Length: 0
Date: Thu, 25 Jun 2015 07:27:46 GMT
Location: http://192.0.2.1/GuestManager/api/devices/deviceDetails/
10:10:10:00:00:01
Server: Apache-Coyote/1.1
```

Request Payload

JSON Format

```
{
  "Device": {
        "provisioningGroupName": "api-device-provGroup",
        "macAddress": "10:10:10:00:00:01",
        "name": "device1",
        "type": "mobile",
        "subType": "generic-android",
        "vlanLabel": "vlan-100",
        "vlanId": "100",
        "enabled": "true",
        "assetType": "TEMPORARY",
        "startDate" : "2016/11/10 10:30:41",
        "endDate" : "2016/11/10 15:30:41",
        "durationUnit" : "HOURS",
        "duration" : 5,
        "deleteOnExpire":"true",
        "networkRights": IT,
        "accessTypes": "[Wired, Wireless]",
        "accessZones": "[Ground-Floor-Left-Wing, Ground-Floor-Right-
Wing]",
        "custom1": "text1",
        "custom2": "text2",
        "custom3": "text3",
        "custom4": "text4",
        "custom5": "text5",
        "comments": "test device create"}
}
```

XML Format

```
<Device>
  <provisioningGroupName>api-device-provGroup</provisioningGroupName>
  <macAddress>10:10:10:00:00:01</macAddress>
  <name>device1</name>
  <type>mobile</type>
  <subType>generic-android</subType>
  <vlanLabel>vlan-100</vlanLabel>
  <vlanId>100</vlanId>
  <enabled>true</enabled>,
  <assetType>TEMPORARY</assetType>,
```

Table continues…

Device Registration REST API

```
  <startDate>2016/11/10 10:30:41</startDate>,
  <endDate>2016/11/10 15:30:41</endDate>,
  <durationUnit>HOURS</durationUnit>,
  <duration>5,</duration>
  <deleteOnExpire>true</deleteOnExpire>,
  <networkRights>IT</networkRights>,
  <accessTypes>[Wired, Wireless]</accessype>,
  <accessZones>[Ground-Floor-Left-Wing, Ground-Floor-Right-Wing]</
accessZones>,
  <custom1>text1</custom1>,
  <custom2>text2</custom2>,
  <custom3>text3</custom3>,
  <custom4>text4</custom4>,
  <custom5>text5</custom5>,
  <comments>test device create</comment>
</Device>
```

Error Response

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

Variable Definition

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

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

Table continues…

Attribute	Type/Value	Description
		The device type depends on the Provisioning Group settings. Parameters and values are: • if typeAccessible is true, then type value is used, otherwise ignored. • 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.
enabled	String	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.
startDate	String	Specifies start date. The format is yyyy/MM/dd HH:mm:ss (24 hours format). The start date configuration depends on the following Provisioning Group settings. Parameter and value is: • if firstLoginActivation is false, then it is ignored.

Table continues…

Attribute	Type/Value	Description
		For Example, 2016/11/10 10:30:41.
endDate	String	Specifies the expiry time of device. The format is yyyy/MM/dd HH:mm:ss (24 hours format). For Example, 2016/11/10 15:30:41. endDate is validated against the maximum duration set in the provisioning group. • If the endDate is not sent in the request, then the duration and durationUnit is considered for calculating the expiry time of the device. • If both the endDate and duration is not sent in the request, then the maximum duration set in the provisioning group is considered for calculating the expiry time of the device. • If both the endDate and duration are sent in the request, then the endDate is considered for calculating the expiry time of the device.
durationUnit	String	Specifies the duration in hours, minutes and days. Accepted value input format is [HOURS:MINUTES:DAYS]
duration	Int	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.
deleteOnExpire	String	Specifies whether the deleteOnExpire must be sent in the request and response. Parameters and values are: • if deleteOnExpire is true, then deleteOnExpire is sent. • if deleteOnExpire is false, then deleteOnExpire is not sent.
networkRights	String	Specifies the network rights selected from the provisioning group. You can select only one networkRights from the available list of networkRights in the provisioning group. **Note:** networkRights is mandatory if networkAccessRights are accessible to Provisioners as per the Provisioning Group configuration.
accessTypes	String	Specifies the access type selected from the provisioning group. You can select multiple accessTypes which is provided in the format separated by a comma (,) and enclosed in square brackets ([ ]). For Example, “[Wired, Wireless]”.

Table continues…

Attribute	Type/Value	Description
accessZones	String	Specifies the access zone selected from the provisioning group. You can select multiple accessZones which is provided in the format separated by a comma (,) and enclosed in square brackets ([ ]). For Example, “[Ground-Floor-Left-Wing, Ground-Floor-Right-Wing]”.
custom1	String	Specifies the text which is sent as a request. If the custom attributes is set as False in the provisioning group, then the custom1 field is not set.
custom2	String	Specifies the text which is sent as a request. If the custom attributes is set as False in the provisioning group, then the custom2 field is not set.
custom3	String	Specifies the text which is sent as a request. If the custom attributes is set as False in the provisioning group, then the custom3 field is not set.
custom4	String	Specifies the text which is sent as a request. If the custom attributes is set as False in the provisioning group, then the custom4 field is not set.
custom5	String	Specifies the text which is sent as a request. If the custom attributes is set as False in the provisioning group, then the custom5 field is not set.
comments	String	Specifies the comment which is sent as a request. This is a custom6 field.

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 is 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`

Table continues…

Update a device	
	`Content-Type: application/json or application/xml`
Response Code	`200 OK`
Response Payload	`Device record updated successfully`
New Parameters	enabled, assetType, startDate, endDate, durationUnit, duration, deleteOnExpire, networkRights, accessTypes, accessZones, custom1, custom2, custom3, custom4, custom5, and comments

Example	**Request**
	`PUT /GuestManager/api/devices/10:0b:01:20:00:06 HTTP/1.1`
	`Host: 192.0.2.1`
	`api-version: v2.0`
	`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",`
	`"enabled": "true",`
	`"assetType": "TEMPORARY",`
	`"startDate" : "2016/11/10 10:30:41",`
	`"endDate" : "2016/11/10 15:30:41",`
	`"durationUnit" : "HOURS",`
	`"duration" : 5,`
	`"deleteOnExpire":"true",`
	`"networkRights": IT,`
	`"accessTypes": "[Wired, Wireless]",`
	`"accessZones": "[Ground-Floor-Left-Wing, Ground-Floor-Right-Wing]",`
	`"custom1": "text1",`
	`"custom2": "text2",`
	`"custom3": "text3",`
	`"custom4": "text4",`
	`"custom5": "text5",`
	`"comments": "test device create"`
	`}`
	`}`
	XML
	`<Device>`
	`<name>device1</name>`
	`<type>mobile</type>`
	`<subType>generic-android</subType>`
	`<vlanLabel>vlan-100</vlanLabel>`
	`<vlanId>100</vlanId>`
	`<enabled>true</enabled>,`
	`<assetType>TEMPORARY</assetType>,`

Table continues…

Update a device	
	<startDate>2016/11/10 10:30:41</startDate>, <endDate>2016/11/10 15:30:41</endDate>, <durationUnit>HOURS</durationUnit>, <duration>5,</duration> <deleteOnExpire>true</deleteOnExpire>, <networkRights>IT</networkRights>, <accessTypes>[Wired, Wireless]</accessTypes>, <accessZones>[Ground-Floor-Left-Wing, Ground-Floor-Right-Wing]</accessZones>, <custom1>text1</custom1>, <custom2>text2</custom2>, <custom3>text3</custom3>, <custom4>text4</custom4>, <custom5>text5</custom5>, <comments>test device create</comments></Device>

For more information on the variable definitions, see Variable Definition on page 44.

Error Cases

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

Error Case	Response Code	Error Response
Device does not exist	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 is 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)

Table continues…

Delete a device	

	`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.`

Example	**Request**
	`DELETE /GuestManager/api/devices/10:0b:01:20:00:06 HTTP/1.1`
	`Host: 192.0.2.1`
	`api-version: v2.0`
	`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 exist	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}`

Table continues…

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

Example	**Request**
	
	DELETE /GuestManager/api/devices/10:0b:01:20:00:06 HTTP/1.1
	Host: 192.0.2.1
	api-version: v2.0
	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": [

Table continues…

Deleting multiple devices

<pre>{
"macAddress": "10:0b:01:20:00:0a",
“reason”: ERROR-RecordNotFound
},
{
"macAddress": "10:10:10:00:00:0b"
“reason”: ERROR-AccessDenied
}]
}
}</pre>

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 contains deleted device details. If hideDeleteDetails is true, response does not contain deleted device details. If hideDeleteDetails is not passed as argument, then response contains 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**
	<pre>DELETE /GuestManager/api/devices/bulkDelete?hideDeleteDetails=false HTTP/1.1
	Host: 192.0.2.1
	api-version: v2.0
	Authorization: Basic dGVzdDp0ZXN0</pre>

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}?viewAll=true **Note:** viewAll is a Boolean data type, which takes a value of either true or false. If the value is true and in the provisioning group if the **provisioner in this group can view all records** checkbox is selected, then the provisioner can view any record. If the value is false then the provisioner can only view the records created by the provisioner.
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`
New Parameters	deleteOnExpire, deviceUserName, networkRights, accessTypes, accessZones, custom1, custom2, custom3, custom4, custom5, and comments

Example	**Request**
	`GET /GuestManager/api/devices/deviceDetails/10:10:10:00:00:02 HTTP/1.1`
	`Host: 192.0.2.1`
	`api-version: v2.0`
	`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",`
	`"startDate": "2015/06/24 07:13:53 PM IST",`
	`"endDate": "-",`
	`"provisioningGroup": "api-device-provGroup",`
	`"provisioner": "Internal/pall",`
	`"vlanLabel": "vlan-100",`
	`"vlanId": "100",`
	`"deleteOnExpire": true,`
	`"deviceUserName: "admin"`
	`"networkRights": IT`
	`"accessTypes": "[Wired, Wireless]"`
	`"accessZones": "[Ground-Floor-Left-Wing, Ground-Floor-Right-`

Table continues…

Fetching Device Details by MAC for a Provisioner

```
Wing]",
        "custom1": "text1",
        "custom2": "text2",
        "custom3": "text3",
        "custom4": "text4",
        "custom5": "text5",
        "comments": "test device create"              }
}
```

Variable Definition

The following table describes the parameters for Device record details.

Attribute	Type/Value	Description
macAddress	String	Specifies the MAC address. The format is xx:xx:xx:xx:xx:xx
type	String	Specifies the device type.
subtype	String	Specifies the device Sub Type.
source	String	Specifies the device source.
enabled	Boolean	Specifies the device record status. Parameter and values are: • if true, then device record is active. • if false, then device record is inactive.
assetType	String	Specifies the asset type. The device record type can be PERMANENT or TEMPORARY.
startDate	String	Specifies the device start date and 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.
endDate	String	Specifies the device end date and 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.

Table continues…

Attribute	Type/Value	Description
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.
deleteOnExpire	String	Specifies whether the deleteOnExpire must be sent in the response. Parameters and values are: • if deleteOnExpire is true, then deleteOnExpire is sent. • if deleteOnExpire is false, then deleteOnExpire is not sent.
deviceUserName	String	Specifies the User Name for the device.
networkRights	String	Specifies the network rights selected from the provisioning group. You can select only one networkRights from the available list of networkRights in the provisioning group.
accessTypes	String	Specifies the access type selected from the provisioning group. You can select multiple accessTypes which is provided in the format separated by a comma (,) and enclosed in square brackets ([ ]). For Example, “[Wired, Wireless]”.
accessZones	String	Specifies the access zone selected from the provisioning group. You can select multiple accessZones which is provided in the format separated by a comma (,) and enclosed in square brackets ([ ]). For Example, “[Ground-Floor-Left-Wing, Ground-Floor-Right-Wing]”.
custom1	String	Specifies the text sent as a response. If the custom attributes is set as False in the provisioning group, then the custom1 field is not sent.

Table continues…

Attribute	Type/Value	Description
custom2	String	Specifies the text sent as a response. If the custom attributes is set as False in the provisioning group, then the custom2 field is not sent.
custom3	String	Specifies the text sent as a response. If the custom attributes is set as False in the provisioning group, then the custom3 field is not sent.
custom4	String	Specifies the text sent as a response. If the custom attributes is set as False in the provisioning group, then the custom4 field is not sent.
custom5	String	Specifies the text sent as a response. If the custom attributes is set as False in the provisioning group, then the custom5 field is not sent.
comment	String	Specifies the comment which is sent as a response. This is a custom6 field.

Fetching Devices Iteratively for a Provisioner

Use this procedure in sequence to fetch devices iteratively for a Provisioner.

1. Get Cursor Id. For more information, see GET Cursor Id on page 46.
2. Get next N devices. For more information, see GET next N Devices on page 47.
3. Get first N devices. For more information, see GET first N Devices on page 49.
4. Get last N devices. For more information, see GET last N Devices on page 50.
5. Get count of total available device records. For more information, see GET count of Total Available Device Records on page 52.
6. Close Cursor Id. For more information, see Close Cursor Id on page 53.

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: 192.0.2.1`
	`api-version: v2.0`
	`Authorization: Basic dGVzdDp0ZXN0`
	`Accept:application/json`
	`Cache-Control: no-cache`
	Response Payload
	`{`
	`PagingInfo:{`
	`"cursorId": "12804370433607408411",`
	`"totalRecord": 4`
	`}`
	`}`

Variable Definition

The following table describes the parameters of Paging Info.

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

GET next N Devices

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

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

Table continues…

GET next N Devices	

	`api-version:{VERSION}`
	
	`Accept: application/json or application/xml`

Response Code	`200 OK`
Response Payload	`Device List`
New Parameters	deleteOnExpire, deviceUserName, networkRights, accessTypes, accessZones, custom1, custom2, custom3, custom4, custom5, and comments

Example	**Request**
	
	`GET /GuestManager/api/devices/next/2/12804370433607408411 HTTP/1.1`
	`Host: 192.0.2.1`
	`api-version: v2.0`
	`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",`
	`"startDate": "2015/06/17 04:47:21 PM IST",`
	`"endDate": "-",`
	`"provisioningGroup": "api-device-provGroup",`
	`"provisioner": "Internal/pall",`
	`"vlanLabel": "vlan-100",`
	`"vlanId": "100",`
	`"deleteOnExpire": true,`
	`"deviceUserName: "admin",`
	`"networkRights": IT,`
	`"accessTypes": "[Wired, Wireless]",`
	`"accessZones": "[Ground-Floor-Left-Wing, Ground-Floor-Right-Wing]",`
	`"custom1": "text1",`
	`"custom2": "text2",`
	`"custom3": "text3",`
	`"custom4": "text4",`
	`"custom5": "text5",`
	`"comments": "test device create" },`
	`{`
	`"macAddress": "10:10:10:00:00:02",`
	`"name": "device1",`
	`"type": "mobile",`
	`"subType": "generic-android",`
	`"source": "GM-p-api-user-device",`
	`"enabled": true,`
	`"assetType": "PERMANENT",`
	`"startDate": "2015/06/24 07:13:53 PM IST",`
	`"endDate": "-",`

Table continues…

GET next N Devices	

	<pre>"provisioningGroup": "api-device-provGroup",
	"provisioner": "Internal/pall"
	"vlanLabel": "vlan-100",
	"vlanId": "100",
	"deleteOnExpire": true,
	"deviceUserName: "admin",
	"networkRights": IT,
	"accessTypes": "[Wired, Wireless]",
	"accessZones": "[Ground-Floor-Left-Wing, Ground-Floor-Right-Wing]",
	"custom1": "text1",
	"custom2": "text2",
	"custom3": "text3",
	"custom4": "text4",
	"custom5": "text5",
	"comments": "test device create" }
	]
	}
	}

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`
New Parameters	deleteOnExpire, deviceUserName, networkRights, accessTypes, accessZones, custom1, custom2, custom3, custom4, custom5, and comments

Example	**Request**
	<pre>GET /GuestManager/api/devices/first/2/12804370433607408411 HTTP/1.1
	Host: 192.0.2.1
	api-version: v2.0
	Authorization: Basic dGVzdDp0ZXN0
	Accept:application/json
	Cache-Control: no-cache</pre>**Response Payload**
	<pre>{
	"DeviceList": {
	"Device": [
	{
	"macAddress": "10:0b:01:20:00:06",</pre>

Table continues…

GET first N devices


```
                    "name": "device-android",
                    "type": "mobile",
                    "subType": "generic-android",
                    "source": "GM-p-api-user-device",
                    "enabled": true,
                    "assetType": "PERMANENT",
                    "startDate": "2015/06/17 04:47:21 PM IST",
                    "endDate": "-",
                    "provisioningGroup": "api-device-provGroup",
                    "provisioner": "Internal/pall"
                    "vlanLabel": "vlan-100",
                    "vlanId": "100",
                    "deleteOnExpire": true,
                    "deviceUserName: "admin",
                    "networkRights": IT,
                    "accessTypes": "[Wired, Wireless]",
                    "accessZones": "[Ground-Floor-Left-Wing, Ground-Floor-
Right-Wing]",
                    "custom1": "text1",
                    "custom2": "text2",
                    "custom3": "text3",
                    "custom4": "text4",
                    "custom5": "text5",
                    "comments": "test device create"            },
                {
                    "macAddress": "10:10:10:00:00:02",
                    "name": "device1",
                    "type": "mobile",
                    "subType": "generic-android",
                    "source": "GM-p-api-user-device",
                    "enabled": true,
                    "assetType": "PERMANENT",
                    "startDate": "2015/06/24 07:13:53 PM IST",
                    "endDate": "-",
                    "provisioningGroup": "api-device-provGroup",
                    "provisioner": "Internal/pall"
                    "vlanLabel": "vlan-100",
                    "vlanId": "100",
                    "deleteOnExpire": true,
                    "deviceUserName: "admin",
                    "networkRights": IT,
                    "accessTypes": "[Wired, Wireless]",
                    "accessZones": "[Ground-Floor-Left-Wing, Ground-Floor-
Right-Wing]",
                    "custom1": "text1",
                    "custom2": "text2",
                    "custom3": "text3",
                    "custom4": "text4",
                    "custom5": "text5",
                    "comments": "test device create"            }
            ]
        }
}
```


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`
New Parameters	deleteOnExpire, deviceUserName, networkRights, accessTypes, accessZones, custom1, custom2, custom3, custom4, custom5, and comments

Example	**Request**
	`GET /GuestManager/api/devices/last/2/12804370433607408411 HTTP/1.1`
	`Host: 192.0.2.1`
	`api-version: v2.0`
	`Authorization: Basic dGVzdDp0ZXN0`
	`Accept:application/json`
	`Cache-Control: no-cache`
	Response Payload
	`{`
	`"DeviceList": {`
	`"Device": [`
	`{`
	`"macAddress": "10:10:10:00:00:01",`
	`"name": "device1",`
	`"type": "mobile",`
	`"subType": "generic-android",`
	`"source": "GM-p-api-user-device",`
	`"enabled": true,`
	`"assetType": "PERMANENT",`
	`"startDate": "2015/06/25 12:52:13 PM IST",`
	`"endDate": "-",`
	`"provisioningGroup": "api-device-provGroup",`
	`"provisioner": "Internal/pall"`
	`"vlanLabel": "vlan-100",`
	`"vlanId": "100"`
	`"deleteOnExpire": true,`
	`"deviceUserName: "admin",`
	`"networkRights": IT,`
	`"accessTypes": "[Wired, Wireless]",`
	`"accessZones": "[Ground-Floor-Left-Wing, Ground-Floor-Right-Wing]",`
	`"custom1": "text1",`
	`"custom2": "text2",`
	`"custom3": "text3",`
	`"custom4": "text4",`
	`"custom5": "text5",`
	`"comments": "test device create" },`
	`{`
	`"macAddress": "10:10:10:00:00:03",`
	`"name": "device1",`

Table continues…

GET last N Devices	

	<pre>"type": "mobile",
	"subType": "generic-android",
	"source": "GM-p-api-user-device",
	"enabled": true,
	"assetType": "PERMANENT",
	"startDate": "2015/06/24 07:46:11 PM IST",
	"endDate": "-",
	"provisioningGroup": "api-device-provGroup",
	"provisioner": "Internal/pall"
	"vlanLabel": "vlan-100",
	"vlanId": "100",
	"deleteOnExpire": true,
	"deviceUserName: "admin",
	"networkRights": IT,
	"accessTypes": "[Wired, Wireless]",
	"accessZones": "[Ground-Floor-Left-Wing, Ground-Floor-Right-Wing]",
	"custom1": "text1",
	"custom2": "text2",
	"custom3": "text3",
	"custom4": "text4",
	"custom5": "text5",
	"comments": "test device create" }
	]
	}
	}

GET count of Total Available Device Records

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

GET count of total available device records	
URI	/api/devices/count/{cursorId}
Method	GET
HTTP Headers	Authorization Scheme: Basic (Base64 encryption) Authorization: username:password api-version:{VERSION} Accept: application/json or application/xml
Response Code	200 OK
Response Payload	Count of Device Records

Example	**Request**
	<pre>GET /GuestManager/api/devices/count/12804370433607408411 HTTP/1.1
	Host: 192.0.2.1
	api-version: v2.0
	Authorization: Basic dGVzdDp0ZXN0
	Accept:application/json
	Cache-Control: no-cache</pre>**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 is not 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: 192.0.2.1`
	`api-version: v2.0`
	`Authorization: Basic dGVzdDp0ZXN0`
	`Accept:application/json`
	`Cache-Control: no-cache`

Error Cases

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

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

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 46.

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`
New Parameters	deleteOnExpire, deviceUserName, networkRights, accessTypes, accessZones, custom1, custom2, custom3, custom4, custom5, and comments

Example	**Request**
	`GET /GuestManager/api/devices?filterCriteria=name&op=startWith&val=device HTTP/1.1`
	`Host: 192.0.2.1`
	`api-version: v2.0`
	`Authorization: Basic dGVzdDp0ZXN0`
	`Accept: application/json`
	`Content-Type:application/json`
	Response Payload
	GET Cursor Id
	`{ "PagingInfo": { "cursorId": "1676924885962954949 5", "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", "startDate": "2015/11/09 09:24:45 AM GMT", "endDate": "-",`

Table continues…

GET Cursor Id


```
        "provisioningGroup": "default",
        "provisioner": "Internal/as"
        "vlanLabel": "vlan-100",
        "vlanId": "100",
        "deleteOnExpire": true,
        "deviceUserName: "admin",
        "networkRights": IT,
        "accessTypes": "[Wired, Wireless]",
        "accessZones": "[Ground-Floor-Left-Wing, Ground-Floor-Right-
Wing]",
        "custom1": "text1",
        "custom2": "text2",
        "custom3": "text3",
        "custom4": "text4",
        "custom5": "text5",
        "comments": "test device create"
      },
      {
        "macAddress": "11:11:11:11:11:13",
        "name": "Test2",
        "type": "mobile",
        "subType": "generic-android",
        "source": "GM-default",
        "enabled": true,
        "assetType": "TEMPORARY",
        "startDate": "2015/10/13 02:22:31 PM GMT",
        "endDate": "2015/10/13 10:22:31 PM GMT",
        "provisioningGroup": "default",
        "provisioner": "Internal/as"
        "vlanLabel": "vlan-100",
        "vlanId": "100",
        "deleteOnExpire": true,
        "deviceUserName: "admin",
        "networkRights": IT,
        "accessTypes": "[Wired, Wireless]",
        "accessZones": "[Ground-Floor-Left-Wing, Ground-Floor-Right-
Wing]",
        "custom1": "text1",
        "custom2": "text2",
        "custom3": "text3",
        "custom4": "text4",
        "custom5": "text5",
        "comments": "test device create"
               }
    ]
  }
}
```


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",
        "source": "GM-default",
        "enabled": true,
```


Table continues…

GET Cursor Id


```
        "assetType": "TEMPORARY",
        "startDate": "2015/10/12 06:48:05 AM GMT",
        "endDate": "2015/10/12 02:48:05 PM GMT",
        "provisioningGroup": "default",
        "provisioner": "Internal/as"
        "vlanLabel": "vlan-100",
        "vlanId": "100",
        "deleteOnExpire": true,
        "deviceUserName: "admin",
        "networkRights": IT,
        "accessTypes": "[Wired, Wireless]",
        "accessZones": "[Ground-Floor-Left-Wing, Ground-Floor-Right-
Wing]",
        "custom1": "text1",
        "custom2": "text2",
        "custom3": "text3",
        "custom4": "text4",
        "custom5": "text5",
        "comments": "test device create"
               },
      {
        "macAddress": "33:33:33:33:33:33",
        "name": "Test4",
        "type": "voip phone",
        "subType": "ONA-SDN",
        "source": "GM-default",
        "enabled": true,
        "assetType": "TEMPORARY",
        "startDate": "2015/10/11 10:00:54 AM GMT",
        "endDate": "2015/10/11 06:00:54 PM GMT",
        "provisioningGroup": "default",
        "provisioner": "Internal/as"
        "vlanLabel": "vlan-100",
        "vlanId": "100",
        "deleteOnExpire": true,
        "deviceUserName: "admin",
        "networkRights": IT,
        "accessTypes": "[Wired, Wireless]",
        "accessZones": "[Ground-Floor-Left-Wing, Ground-Floor-Right-
Wing]",
        "custom1": "text1",
        "custom2": "text2",
        "custom3": "text3",
        "custom4": "text4",
        "custom5": "text5",
        "comments": "test device create"
                }
    ]
  }
}
```


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",
```


Table continues…

GET Cursor Id


```
            "subType": "generic-android",
            "source": "GM-default",
            "enabled": true,
            "assetType": "TEMPORARY",
            "startDate": "First Login Pending",
            "endDate": "-",
            "provisioningGroup": "karthik_group",
            "provisioner": "Internal/as",
            "vlanLabel": "vlan-100",
            "vlanId": "100",
            "deleteOnExpire": true,
            "deviceUserName: "admin",
            "networkRights": IT,
            "accessTypes": "[Wired, Wireless]",
            "accessZones": "[Ground-Floor-Left-Wing, Ground-Floor-Right-
Wing]",
            "custom1": "text1",
            "custom2": "text2",
            "custom3": "text3",
            "custom4": "text4",
            "custom5": "text5",
            "comments": "test device create"
                    },
          {
            "macAddress": "00:0a:95:9d:68:34",
            "name": "Test6",
            "type": "",
            "subType": "n/a",
            "source": "GM-default",
            "enabled": true,
            "assetType": "PERMANENT",
            "startDate": "2015/10/30 07:04:27 AM GMT",
            "endDate": "-",
            "provisioningGroup": "default",
            "provisioner": "Internal/as",
            "vlanLabel": "vlan-100",
            "vlanId": "100",
            "deleteOnExpire": true,
            "deviceUserName: "admin",
            "networkRights": IT,
            "accessTypes": "[Wired, Wireless]",
            "accessZones": "[Ground-Floor-Left-Wing, Ground-Floor-Right-
Wing]",
            "custom1": "text1",
            "custom2": "text2",
            "custom3": "text3",
            "custom4": "text4",
            "custom5": "text5",
            "comments": "test device create"
                    }
        ]
      }
}
```


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
startDate	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 Device activation
endDate	greaterThan greaterThanEqual lessThan lessThanEqual	Date Format: yyyy/MM/dd hh:mm:ss a z ex: 2015/09/08 10:10:22 AM IST	Expiry time of device
provisioningGroup	Equal	String	Provisioning Group name Note: Provisioning group must be accessible to provisioner and have device rights.

Table continues…

Filter Criteria	Options	Value Type	Description
deviceUserName	equal notEqual startWith endsWith contains	String	Device user name

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 53. 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`
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: 192.0.2.1`
	`api-version: v2.0`
	`Authorization: Basic dGVzdDp0ZXN0`
	`Accept:application/json`
	`Content-Type: application/json`

Table continues…

Fetching devices with filter and without details

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

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 exist 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 exist. • 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: 192.0.2.1`
	`api-version: v2.0`
	`Authorization: Basic dGVzdDp0ZXN0`
	`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 exist 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: 192.0.2.1`
	`api-version: v2.0`
	`Authorization: Basic dGVzdDp0ZXN0`
	`Accept:application/json`
	`Cache-Control: no-cache`
	Response Payload
	`{ "DeviceList": { "Device": [{ "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 Operations

This section describes CRUD operations with respect to Guest Users.

Guest User Registration

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

Note:

If the Account Validity Duration is set to Permanent in the Provisioning group for the logged in Provisioner, Guest User created will be Permanent account and will not be expired. The Delete on Expire process will be set to false.

If the User is permanent and you provide the `end date, duration, duration units` in request pay load, then these attributes are ignored while creating / modifying the Guest User account.

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`
New Parameters	endDate, deleteOnExpire, enabled, networkRights, accessTypes, accessZones, and comments

Example	**Request**
	`POST /GuestManager/api/guestUsers HTTP/1.1`
	`Host: 192.0.2.1`
	`api-version: v2.0`
	`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@extremenetworks.com", "cellPhone" : "2991199112",`
	`"phoneCarrier" : "T-Mobile", "guestDetails" : "guest Details-DL",`
	`"startDate" : "2015/06/25 16:16:41", "durationUnit" : "HOURS",`

Table continues…

Guest User Registration REST API

```
"duration" : 5, "endDate" : "2016/04/27 15:30:41", "deleteOnExpire" :
"true", "enabled" : "true", "networkRights" : IT, "accessTypes" :
"[Wired, Wireless]", "accessZones": "[Ground-Floor-Left-Wing, Ground-
Floor-Left-Wing]", "comments" : "guest user creation"} }
```

Request Payload

JSON Format

```
{
   "GuestUser" : {
      "provisioningGroupName" : "pg-api-user",
      "firstName" : "fName1",
      "lastName" : "lName1",
      "userName" : "guestUser1",
      "password" : "Abc@12",
      "email" : "test@extremenetworks.com",
      "cellPhone" : "2991199112",
      "phoneCarrier" : "T-Mobile",
      "guestDetails" : "guest Details-DL",
      "startDate" : "2015/06/25 16:16:41",
      "durationUnit" : "HOURS",
      "duration" : 5,
      "endDate" : "2016/04/27 15:30:41",
      "deleteOnExpire":"true",
      "enabled":"true",
      "networkRights": IT,
      "accessTypes": "[Wired, Wireless]",
      "accessZones": "[Ground-Floor-Left-Wing, Ground-Floor-Left-
Wing]",
      "comments" : "guest user creation"

   }
}
```

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@extremenetworks.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>
   <endDate>2016/04/27 15:30:41</endDate>
   <deleteOnExpire>true</deleteOnExpire>
   <enabled>true</enabled>
   <networkRights>IT</networkRights>
   <accessTypes>[Wired, Wireless]</accessTypes>
   <accessZones>[Ground-Floor-Left-Wing, Ground-Floor-Left-Wing]</
accessZones>
   <comments>guest user creation</comments>

</GuestUser>
```

Table continues…

Guest User Registration REST API

Response Header

```
Connection → close
Content-Type → application/json
Date → Thu, 25 Jun 2015 07:10:48 GMT
Location → https://192.0.2.1/GuestManager/api/guestUsers/
guestUserDetails/guestUser1
Transfer-Encoding → chunked
```

Response Payload

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

Error Response

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

Variable Definition

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

Request Guest User

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

Table continues…

Attribute	Type/Value	Description
		The guest user first name configuration depends on the following Provisioning Group settings. Parameters and values are: • If firstAndLastNameAccessible is true, then first name value is used, otherwise it is ignored. • If firstAndLastNameRequired is true, then first name is mandatory, otherwise it is optional.
lastName	String	Specifies last name of the guest user. The maximum length of the user name can be 30 characters. The allowed characters are alphabets (upper and lower case), numbers (0 to 9), and special characters (hyphen, underscore and space). The guest user last name configuration depends on the following Provisioning Group settings. Parameters and values are: • If firstAndLastNameAccessible is true, then last name value is used, otherwise it is ignored. • If firstAndLastNameRequired is true, then last name is mandatory, otherwise it is optional.
email	String	Specifies valid email address of the guest user. The guest user email address configuration depends on the following Provisioning Group settings. Parameter and value is: • If emailRequired is true, then email is mandatory, otherwise it is optional.
password	String	Specifies the password of the guest user. It is Base64 encrypted. The guest password configuration depends on the following Provisioning Group settings. Parameter and value is: • If passwordAccessible is true, then mandatory, otherwise it is optional and value ignored. **Note:** Password must follow password complexity which defined in Provisioning Group Setting.
cellphone	String (Valid Cell Phone)	Specifies the valid cellphone of the guest user. The maximum length of the user name is 12 digits. The guest cellphone configuration depends on the following Provisioning Group settings. Parameter and value is: If cellPhoneRequired is true, then cellphone is mandatory, otherwise it is optional.
phoneCarrier	String (Carrier Name)	Specifies valid carrier name, valid only if it is registered in SMS Gateways under Administrator notification setting in Guest and IoT Manager. Required if cellphone field is not empty and no default Gateway is configured in Guest and IoT Manager. If phoneCarrier is

Table continues…

Attribute	Type/Value	Description
		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 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.
endDate	String	Specifies the expiry time of Guest User. The format is yyyy/MM/dd HH:mm:ss (24 hours format). For Example, 2016/11/10 15:30:41. endDate is validated against the maximum duration set in the provisioning group. • If the endDate is not sent in the request, then the duration and durationUnit is considered for calculating the expiry time of the device. • If both the endDate and duration is not sent in the request, then the maximum duration set in the provisioning group is considered for calculating the expiry time of the device. • If both the endDate and duration are sent in the request, then the endDate is considered for calculating the expiry time of the device.
deleteOnExpire	Boolean	Specifies whether the deleteOnExpire must be sent in the request and response. Parameters and values are: • if deleteOnExpire is true, then deleteOnExpire is sent. • if deleteOnExpire is false, then deleteOnExpire is not sent.
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.

Table continues…

Attribute	Type/Value	Description
networkRights	String	Specifies the network rights selected from the provisioning group. You can select only one networkRights from the available list of networkRights in the provisioning group. **Note:** networkRights is mandatory if networkAccessRights are accessible to Provisioners as per the Provisioning Group configuration.
accessTypes	String	Specifies the access type selected from the provisioning group. You can select multiple accessTypes which is provided in the format separated by a comma (,) and enclosed in square brackets ([ ]). For Example, “[Wired, Wireless]”.
accessZones	String	Specifies the access zone selected from the provisioning group. You can select multiple accessZones which is provided in the format separated by a comma (,) and enclosed in square brackets ([ ]). For Example, “[Ground-Floor-Left-Wing, Ground-Floor-Right-Wing]”.
comments	String	Specifies the comment which is sent as a request.

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 and IoT Manager to send EMAIL/SMS notification of login credential to the Guest User. The options EMAIL/SMS is enabled by checking the checkbox 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}

Table continues…

Re-send Credentials through EMAIL/SMS to Guest User by 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: 192.0.2.1`
	`api-version: v2.0`
	`Authorization: Basic dGVzdDp0ZXN0`
	`Accept:application/json`
	`Cache-Control: no-cache`

Error Cases

Error Case	Response Code	Error Response
Guest User does not exist	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.
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.

If the Account Validation is set to Permanent in the Provisioning group for the logged in provisioner, Guest User created will be Permanent account and will not be expired. The "Delete on Expire" process will be set to false.

If the user is permanent and you provide the `end date, duration, duration units` in request pay load, then these attributes are ignored while creating / modifying the Guest User account.

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`
New Parameters	endDate, deleteOnExpire, enabled, networkRights, accessTypes, accessZones, and comments

Example	**Request**
	`PUT /GuestManager/api/guestUsers/guestUser1 HTTP/1.1`
	`Host: 192.0.2.1`
	`api-version: v2.0`
	`Authorization: Basic dGVzdDp0ZXN0`
	`Accept:application/json`
	`Content-type: application/json`
	Request Payload
	JSON
	`{`
	`"GuestUser" : {`
	`"firstName" : "fName1",`
	`"lastName" : "lName1",`
	`"password" : "Abc@12",`
	`"email" : "test@extremenetworks.com",`
	`"cellPhone" : "2991199112",`
	`"phoneCarrier" : "T-Mobile",`
	`"guestDetails" : "guest Details-DL",`

Table continues…

Update a Guest User

```
      "startDate" : "2015/06/25 16:16:41",
      "durationUnit" : "HOURS",
      "duration" : 5,
      "endDate" : "2016/04/27 15:30:41",
      "deleteOnExpire":"true",
      "enabled":"true",
      "networkRights": IT,
      "accessTypes": "[Wired, Wireless]",
      "accessZones": "[Ground-Floor-Left-Wing, Ground-Floor-Left-
Wing]",
      "comments" : "guest user creation"
   }
}
```

XML

```
<?xml version="1.0" encoding="UTF-8"?>
<GuestUser>
   <firstName>fName1</firstName>
   <lastName>lName1</lastName>
   <email>test@extremenetworks.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>
   <endDate>2016/04/27 15:30:41</endDate>
   <deleteOnExpire>true</deleteOnExpire>
   <enabled>true</enabled>
   <networkRights>IT</networkRights>
   <accessTypes>[Wired, Wireless]</accessTypes>
   <accessZones>[Ground-Floor-Left-Wing, Ground-Floor-Left-Wing]</
accessZones>
   <comments>guest user creation</comments>
</GuestUser>
```

Response Payload

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

For more information on the variable definitions, see Variable Definition on page 78.

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 exist	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 access the Guest User: <username>. 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: 192.0.2.1`
	`api-version: v2.0`
	`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 exist	404 not found	
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: 192.0.2.1`
	`api-version: v2.0`
	`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: 192.0.2.1`
	`api-version: v2.0`
	`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.

Note:

If the user account is permanent, then the end date in the response pay load is displayed as : “-“

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`

Table continues…

Fetching Guest User details by username for a Provisioner	

	`api-version:{VERSION}`
	
	`Accept: application/json or application/xml`

Response Code	`200 OK`
Response Payload	`Guest User Details`
New Parameters	networkRights, accessTypes, accessZones, comments, enabled, deleteOnExpire

Example	**Request**
	
	`GET /GuestManager/api/guestUsers/guestUserDetails/guestUser1 HTTP/1.1`
	`Host: 192.0.2.1`
	`api-version: v2.0`
	`Authorization: Basic dGVzdDp0ZXN0`
	`Accept:application/json`
	`Cache-Control: no-cache`
	
	Response Payload
	
	JSON Format
	
	`{`
	`"GuestUser": {`
	`"userName": "guestUser1",`
	`"firstName": "guest",`
	`"lastName": "User1",`
	`"email": "test@extremenetworks.com",`
	`"smsAddress": "2991199112@ tmomail.net",`
	`"startDate": "2015/06/25 04:16:41 PM IST",`
	`"endDate": "2015/06/25 09:16:41 PM IST",`
	`"provisioningGroup": "pg-api-user",`
	`"provisioner": "Internal/pall",`
	`"guestDetails": "guest Details-DL",`
	`"networkRights": IT,`
	`"accessTypes": "[Wired, Wireless]",`
	`"accessZones": "[Ground-Floor-Left-Wing, Ground-Floor-Right-Wing]",`
	`"comments": "test device create",`
	`"enabled": true,`
	`"deleteOnExpire": true`
	
	`}`
	`}`
	
	End Date
	
	`{`
	`"GuestUser": {`
	`"userName": "guestUser1",`
	`"firstName": "guest",`
	`"lastName": "User1",`
	`"email": "test@extremenetworks.com",`
	`"smsAddress": "2991199112@ tmomail.net",`
	`"startDate": "2015/06/25 04:16:41 PM IST",`
	`"endDate": "-",`
	`"provisioningGroup": "pg-api-user",`
	`"provisioner": "Internal/pall",`
	`"guestDetails": "guest Details-DL",`
	`"networkRights": IT,`

Table continues…

Fetching Guest User details by username for a Provisioner

```
        "accessTypes": "[Wired, Wireless]",
        "accessZones": "[Ground-Floor-Left-Wing, Ground-Floor-Right-
Wing]",
        "comments": "test device create",
        "enabled": true,
        "deleteOnExpire": true


    }
}
```

XML Format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<GuestUser>
    <userName>guestUser1</userName>
    <firstName>guest</firstName>
    <lastName>User1<lastName>
    <email>test@extremenetworks.com</email>
    <smsAddress>2991199112@ tmomail.net</smsAddress>
    <startDate>2015/06/25 04:16:41 PM IST</startDate>
    <endDate>2015/06/25 09:16:41 PM IST</endDate>
    <provisioningGroup>pg-api-user</provisioningGroup>
    <provisioner>Internal/pall</provisioner>
    <guestDetails>guest Details-DL</guestDetails>
    <networkRights>IT</networkRights>
    <accessTypes>[Wired, Wireless]</accessTypes>
    <accessZones>[Ground-Floor-Left-Wing, Ground-Floor-Right-Wing]</
accessZones>
    <comments>test device create</comment>
    <enabled>true</enabled>
    <deleteOnExpire>true</deleteOnExpire>
</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.
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
startDate	String	Specifies the activation date and 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
endDate	String	Specifies the expiry date and time of guest user account The format is yyyy/MM/dd hh:mm:ss a z

Table continues…

Attribute	Type/Value	Description
		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.
networkRights	String	Specifies the network rights selected from the provisioning group. You can select only one networkRights from the available list of networkRights in the provisioning group.
accessTypes	String	Specifies the access type selected from the provisioning group. You can select multiple accessTypes which is provided in the format separated by a comma (,) and enclosed in square brackets ([ ]). For Example, “[Wired, Wireless]”.
accessZones	String	Specifies the access zone selected from the provisioning group. You can select multiple accessZone which is provided in the format separated by a comma (,) and enclosed in square brackets ([ ]). For Example, “[Ground-Floor-Left-Wing, Ground-Floor-Right-Wing]”.
Comments	String	Specifies the comment which is sent as a request.
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.
deleteOnExpire	Boolean	Specifies whether the deleteOnExpire must be sent in the request and response. Parameters and values are: • if deleteOnExpire is true, then deleteOnExpire is sent. • if deleteOnExpire is false, then deleteOnExpire is not sent.

Fetching Guest Users Iteratively for a Provisioner

Use this procedures in sequence to fetch guest users iteratively for a Provisioner.

1. Get Cursor Id. For more information, see GET Cursor Id on page 80.
2. Get next N Guest Users. For more information, see GET next N Guest Users on page 80.
3. Get first N Guest Users. For more information, see GET first N Guest Users on page 82.
4. Get last N Guest Users. For more information, see GET last N Guest Users on page 83.
5. Get count of total available Guest User records. For more information, see GET count of Total Available Guest User Records on page 84.

6. Close Cursor Id. For more information, see Close Cursor Id on page 85.

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: 192.0.2.1`
	`api-version: v2.0`
	`Authorization: Basic dGVzdDp0ZXN0`
	`Accept:application/json`
	`Cache-Control: no-cache`
	Response Payload
	`{`
	`"PagingInfo": {`
	`"cursorId": "13666304570298546472",`
	`"totalRecord": 10`
	`}`
	`}`

GET next N Guest Users

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

GET next N Guest Users	
URI	/api/guestUsers/next/{N}/{cursorId}
Method	GET

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

Response Code	`200 OK`

Table continues…

GET next N Guest Users	
Response Payload	`Guest User List`
New Parameters	networkRights, accessTypes, accessZones, comments, enabled, deleteOnExpire

Example	**Request**
	`GET /GuestManager/api/guestUsers/next/2/13666304570298546472 HTTP/1.1`
	`Host: 192.0.2.1`
	`api-version: v2.0`
	`Authorization: Basic dGVzdDp0ZXN0`
	`Accept:application/json`
	`Cache-Control: no-cache`
	Response Payload

```
{
    "GuestUserList": {
        "GuestUser": [
            {
                "userName": "GuestUser1",
                "firstName": "guest",
                "lastName": "User1",
                "email": "testl2@extremenetworks.com",
                "smsAddress": "9845342309@T-Mobile",
                "startDate": "2015/06/24 04:16:41 PM IST",
                "endDate": "2015/06/24 08:16:41 PM IST",
                "provisioningGroup": "p-api-user-device",
                "provisioner": "Internal/pall",
                "guestDetails": "guest Details",
                "networkRights": IT,
                "accessTypes": "[Wired, Wireless]",
                "accessZones": "[Ground-Floor-Left-Wing, Ground-Floor-
Right-Wing]",
                "comment": "test device create",
                "enabled": true,
                "deleteOnExpire": true
            },
            {
                "userName": "GuestUser2",
                "firstName": "Guest",
                "lastName": "User2",
                "email": "test884@extremenetworks.com",
                "smsAddress": "9622000000@tmomail.net",
                "startDate": "2015/06/24 04:16:41 PM IST",
                "endDate": "2015/06/25 12:16:41 AM IST",
                "provisioningGroup": "pg-fl-no",
                "provisioner": "Internal/pall",
                "guestDetails": "guest Details",
                "networkRights": IT,
                "accessTypes": "[Wired, Wireless]",
                "accessZones": "[Ground-Floor-Left-Wing, Ground-Floor-
Right-Wing]",
                "comment": "test device create",
                "enabled": true,
                "deleteOnExpire": true
            }
        ]
    }
}
```

GET first N Guest Users

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

GET first N Guest Users	
URI	/api/ guestUsers/first/{N}/{cursorId}
Method	GET

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

Response Code	`200 OK`
Response Payload	`Guest User List`
New Parameters	networkRights, accessTypes, accessZones, comments, enabled, deleteOnExpire

Example	**Request**
	`GET /GuestManager/api/guestUsers/first/2/13666304570298546472 HTTP/1.1`
	`Host: 192.0.2.1`
	`api-version: v2.0`
	`Authorization: Basic dGVzdDp0ZXN0`
	`Accept:application/json`
	`Cache-Control: no-cache`
	Response Payload
	`{ "GuestUserList": { "GuestUser": [{ "userName": "GuestUser1", "firstName": "Guest", "lastName": "User1", "email": "testl2@extremenetworks.com", "smsAddress": "9845342309@T-Mobile", "startDate": "2015/06/24 04:16:41 PM IST", "endDate": "2015/06/24 08:16:41 PM IST", "provisioningGroup": "p-api-user-device", "provisioner": "Internal/pall", "guestDetails": "guest Details", "networkRights": IT, "accessTypes": "[Wired, Wireless]", "accessZones": "[Ground-Floor-Left-Wing, Ground-Floor-Right-Wing]", "comment": "test device create", "enabled": true, "deleteOnExpire": true }, { "userName": "GuestUser2", "firstName": "Guest", "lastName": "User2", "email": "test884@extremenetworks.com", "smsAddress": "9622000000@tmomail.net",`

Table continues…

GET first N Guest Users	

	<pre>"startDate": "2015/06/24 04:16:41 PM IST",
	"endDate": "2015/06/25 12:16:41 AM IST",
	"provisioningGroup": "pg-fl-no",
	"provisioner": "Internal/pall",
	"guestDetails": "guest Details",
	"networkRights": IT,
	"accessTypes": "[Wired, Wireless]",
	"accessZones": "[Ground-Floor-Left-Wing, Ground-Floor-Right-Wing]",
	"comment": "test device create",
	"enabled": true,
	"deleteOnExpire": true
	}
	]
	}
	}

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`
New Parameters	networkRights, accessTypes, accessZones, comments, enabled, deleteOnExpire

Example	**Request**
	<pre>GET /GuestManager/api/guestUsers/last/2/13666304570298546472 HTTP/1.1
	Host: 192.0.2.1
	api-version: v2.0
	Authorization: Basic dGVzdDp0ZXN0
	Accept:application/json
	Cache-Control: no-cache</pre>**Response Payload**
	<pre>{
	"GuestUserList": {
	"GuestUser": [
	{
	"userName": "GuestUser10",
	"firstName": "Guest",
	"lastName": "User10",
	"email": "test10@extremenetworks.com",
	"smsAddress": "9123456789@tmomail.net",</pre>

Table continues…

GET last N Guest Users


```
                    "startDate": "2015/06/25 04:16:41 PM GMT",
                    "endDate": "2015/06/25 09:16:41 PM GMT",
                    "provisioningGroup": "pg-api-user",
                    "provisioner": "Internal/pall",
                    "guestDetails": "guest Details-DL",
                             "networkRights": IT,
                    "accessTypes": "[Wired, Wireless]",
                    "accessZones": "[Ground-Floor-Left-Wing, Ground-Floor-
Right-Wing]",
                    "comment": "test device create",
                    "enabled": true,
                    "deleteOnExpire": true
                },
                {
                    "userName": "GuestUser9",
                    "firstName": "Guest",
                    "lastName": "User9",
                    "email": "test9@extremenetworks.com",
                    "smsAddress": "9329393922@tmomail.net",
                    "startDate": "2015/06/24 04:16:41 PM IST",
                    "endDate": "2015/06/25 12:16:41 AM IST",
                    "provisioningGroup": "pg-user-email-phone",
                    "provisioner": "Internal/pall",
                    "guestDetails": "guest Details",
                    "networkRights": IT,
                    "accessTypes": "[Wired, Wireless]",
                    "accessZones": "[Ground-Floor-Left-Wing, Ground-Floor-
Right-Wing]",
                    "comment": "test device create",
                    "enabled": true,
                    "deleteOnExpire": true
                }
            ]
        }
}
```


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`

Table continues…

GET count of total available Guest User records	

Example	**Request**
	`GET /GuestManager/api/guestUsers/count/13666304570298546472 HTTP/1.1`
	`Host: 192.0.2.1`
	`api-version: v2.0`
	`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`
Response Payload	NA

Example	**Request**
	`GET /GuestManager/api/guestUsers/close/13666304570298546472 HTTP/1.1`
	`Host: 192.0.2.1`
	`api-version: v2.0`
	`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 79.

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`
New Parameters	networkRights, accessTypes, accessZones, comments, enabled, deleteOnExpire

Example	**Request**
	`GET /GuestManager/api/guestUsers?filterCriteria=userName&op=startWith&val=User HTTP/1.1`
	`Host: 192.0.2.1`
	`api-version: v2.0`
	`Authorization: Basic dGVzdDp0ZXN0`
	`Accept:application/json`
	`Content-type: application/json`
	Response Payload
	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@extremenetworks.com", "smsAddress": "9845342309@T-Mobile", "startDate": "2015/11/09 04:16:41 PM IST", "endDate": "2015/11/90 08:16:41 PM IST", "provisioningGroup": "p-api-user-device", "provisioner": "Internal/pall", "guestDetails": "guest Details" "networkRights": IT "accessTypes": "[Wired, Wireless]" "accessZones": "[Ground-Floor-Left-Wing, Ground-Floor-Right-Wing]",`

Table continues…

Fetching Guest User with filter

```
                "comment": "test device create"
                "enabled": true,
                "deleteOnExpire":
true,                                          },
            {
                "userName": "GuestUser2",
                "firstName": "Guest",
                "lastName": "User2",
                "email": "test884@extremenetworks.com",
                "smsAddress": "9622000000@tmomail.net",
                "startDate": "2015/11/09 04:16:41 PM IST",
                "endDate": "2015/11/10 12:16:41 AM IST",
                "provisioningGroup": "pg-fl-no",
                "provisioner": "Internal/pall",
                "guestDetails": "guest Details"
                "networkRights": IT
                "accessTypes": "[Wired, Wireless]"
                "accessZones": "[Ground-Floor-Left-Wing, Ground-Floor-
Right-Wing]",
                "comment": "test device create"
                "enabled": true,
                "deleteOnExpire": true,               }
        ]
    }
}
```

GET first N Guest Users

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

```
{
    "GuestUserList": {
        "GuestUser": [
            {
                "userName": "GuestUser1",
                "firstName": "Guest",
                "lastName": "User1",
                "email": "test12@extremenetworks.com",
                "smsAddress": "9845342309@T-Mobile",
                "startDate": "2015/11/09 04:16:41 PM IST",
                "endDate": "2015/11/09 08:16:41 PM IST",
                "provisioningGroup": "p-api-user-device",
                "provisioner": "Internal/pall",
                "guestDetails": "guest Details"
                "networkRights": IT
                "accessTypes": "[Wired, Wireless]"
                "accessZones": "[Ground-Floor-Left-Wing, Ground-Floor-
Right-Wing]",
                "comment": "test device create"
                "enabled": true,
                "deleteOnExpire": true,               },
            {
                "userName": "GuestUser2",
                "firstName": "Guest",
                "lastName": "User2",
                "email": "test884@extremenetworks.com",
                "smsAddress": "9622000000@tmomail.net",
                "startDate": "2015/11/09 04:16:41 PM IST",
                "endDate": "2015/11/10 12:16:41 AM IST",
                "provisioningGroup": "pg-fl-no",
                "provisioner": "Internal/pall",
                "guestDetails": "guest Details"
```

Table continues…

Fetching Guest User with filter

```
                "networkRights": IT
                "accessTypes": "[Wired, Wireless]"
                "accessZones": "[Ground-Floor-Left-Wing, Ground-Floor-
Right-Wing]",
                "comment": "test device create"
                "enabled": true,
                "deleteOnExpire": true,                }
        ]
    }
}
```

GET last N Guest Users

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

```
{
    "GuestUserList": {
        "GuestUser": [
            {
                "userName": "GuestUser10",
                "firstName": "Guest",
                "lastName": "User10",
                "email": "test10@extremenetworks.com",
                "smsAddress": "9123456789@tmomail.net",
                "startDate": "2015/11/10 04:16:41 PM GMT",
                "endDate": "2015/11/10 09:16:41 PM GMT",
                "provisioningGroup": "pg-api-user",
                "provisioner": "Internal/pall",
                "guestDetails": "guest Details-DL"
                "networkRights": IT
                "accessTypes": "[Wired, Wireless]"
                "accessZones": "[Ground-Floor-Left-Wing, Ground-Floor-
Right-Wing]",
                "comment": "test device create"
                "enabled": true,
                "deleteOnExpire": true,                },
            {
                "userName": "GuestUser9",
                "firstName": "Guest",
                "lastName": "User9",
                "email": "test9@extremenetworks.com",
                "smsAddress": "9329393922@tmomail.net",
                "startDate": "2015/11/10 04:16:41 PM IST",
                "endDate": "2015/11/10 12:16:41 AM IST",
                "provisioningGroup": "pg-user-email-phone",
                "provisioner": "Internal/pall",
                "guestDetails": "guest Details",
                "networkRights": IT,
                "accessTypes": "[Wired, Wireless]",
                "accessZones": "[Ground-Floor-Left-Wing, Ground-Floor-
Right-Wing]",
                "comment": "test device create",
                "enabled": true,
                "deleteOnExpire": true                }
        ]
    }
}
```

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
startDate	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
endDate		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
provisioningGroup	Equal	String	Provisioning Group name

Table continues…

Filter Criteria	Options	Value Type	Description
			Note: Note: Provisioning group must be accessible to provisioner and have Guest User rights.

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 85. 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: 192.0.2.1`
	`api-version: v2.0`
	`Authorization: Basic dGVzdDp0ZXN0`
	`Accept:application/json`
	`Content-Type: application/json`

Table continues…

Fetching Guest Users with filter and without details

Response Payload

GET Cursor Id

```
{
  "PagingInfo": {
    "cursorId": "4490890776062129399",
    "totalRecord": 5
  }
}
```

GET next N Guest Users

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 exist 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 exist. • FOUND_BUT_EXPIRED - if user exists but expired.

Example	**Request**
	`GET /GuestManager/api/guestUsers/userStatusQuery/user1 HTTP/1.1`
	`Host: 192.0.2.1`
	`api-version: v2.0`
	`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 exist 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.
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: 192.0.2.1`
	`api-version: v2.0`
	`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"`
	`},`
	`]`
	`}`
	`}`