



Getting Started with Mojo REST APIs

Contents

About MOJO REST API	3
API Call Structure	3
API Parameters	4
API Request	5
API Response	6
How MOJO APIs Work	7
Create a Key and Assign Service Privileges	7
Get Started	10
How to Use Mojo APIs	10
Using a Web Browser-Based REST Client	10
Add REST Client to Mozilla Firefox	10
Run Mojo APIs Using REST Client	11
Log In to Mojo Launchpad Using Mojo REST API	12
Fetch Services	13
Log In to Mojo Wireless Manager Using Mojo REST API	14
Get Mojo Wireless Manager Server Version	15
Get Mojo Wireless Manager API Version	15
Log In to Mojo Guest Manager Using Mojo REST API	16
Get Mojo Guest Manager Server Version	16
Few Sample Mojo API Calls	18
Get Observing Managed Devices For a Client	18
Get the Location Tree	18
Add a Location	19
Modify a Location	19
Delete a Location	20
Sample Codes Leveraging Mojo REST APIs	21

About MOJO REST API

Mojo's Cloud services comprising of Mojo Launchpad, Mojo Wireless Manager, and Mojo Guest Manager expose their functionality through REST APIs. Like most REST API implementations, the Mojo REST APIs perform CRUD operations through the HTTP methods PUT, GET, POST, and DELETE on the various resources. The resources or objects utilized by these services are represented in the JSON (JavaScript Object Notation) format.

This topic covers the following:

- [API Call Structure](#)
- [API Parameters](#)
- [API Request](#)
- [API Response](#)

The complete Mojo REST API reference is available at

<http://prod.mojonetworks.com/WebAPI/source/index.html>

API Call Structure

A REST API call has the following syntax:

<HTTP_request_method> <Base_URL>/<API_Endpoint>

HTTP_request_method	Can be one of the following HTTP request types: GET, PUT, POST, or DELETE.
Base_URL	<p>It is a base path for a service on which you call a Mojo API. The base URL differs based on the Mojo Cloud service on which the call is being made. For example:</p> <p>Base URL for Mojo Wireless Manager is: <code>https://<MWM_Server>/new/webservice/{version}</code></p> <p>Base URL for Mojo Launch Pad is: <code>https://dashboard.mojonetworks.com/rest/api/{version}</code></p> <p>Base URL for Mojo Guest Manager is: <code>https://<MGM_Server>/api/{version}</code></p> <p>Here,</p> <ul style="list-style-type: none">• MWM_Server — the IP address or the Fully Qualified Domain Name (FQDN) of the Mojo Wireless Manager Server.

	<ul style="list-style-type: none"> • MLP_Server — the IP address or the FQDN of the Mojo Launchpad Server. • MGM_Server— the IP address or the FQDN of the Mojo Guest Server. • Version — The API version you want to use. As a best practice, always use the latest API version. <p>Sample Base URL using IP address:</p> <p><code>https://123.123.123.123/new/webservice/v2</code></p> <p>Sample Base URL using FQDN:</p> <p><code>https://dashboard.mojonetworks.com/rest/api/v2</code></p>
API_Endpoint	An API endpoint individually identifies a resource in a service suite. For example: <code>/system/version</code> is the endpoint of the API to fetch the Mojo Wireless Manager version.

API Parameters

An API endpoint might take parameters to perform the API action.

The API request parameters are of the following types:

Path Parameters/ URL Parameters

The path parameters are part of the URL and are noted in curly brackets { }. These parameters must be replaced with the appropriate values in the API calls.

For example:

API Syntax:

```
GET https://<MWM_Host_Name>/new/webservice/v3/devices/aps/
{startIndex}/{pageSize}
```

Sample Call:

```
GET https://<MWM_Host_Name>/new/webservice
/v3/devices/aps/0/25
```

Query Parameters

The Query parameters are name-value pairs that are appended to the URL. The URL is followed by a question mark (?) and then appended with the name-value pairs of the query parameters. If an API call takes multiple query parameters, the name-value pairs are separated by the ampersand sign (&). For example:

API Syntax:

```
GET https://<MWM_Host_Name>/new/webservice/v3/devices
/manageddevices?
sortBy=<value>&offset=<value>&limit=<value>&ascending=<value>
```

Sample Call:

```
GET https://<MWM_Host_Name>/new/webservice
/v3/devices/manageddevices?sortBy="name"&offset=0
&limit=6&ascending=false
```

Query parameters can be provided in two ways, encoded or decoded format. Typically, if the value for a query parameter is in the JSON format, you need to encode it. There are many applications to encode and decode query parameters such as: [URL Decoder/Encoder](#).

Example for encoded query parameter:

The API call Get Filtered List of Clients takes Filter as an optional query parameter. Values of Filter query parameter are provided in encoded format.

```
GET https://<MWM_Host_Name>/new/webservice
/V3/devices/clients/0/25
```

Actual values of Filter query parameter:

```
filter={"property":"macaddress","value":
["B4:EF:FA:D8:F8:BB", "B4:EF:FA:D8:F8:BB"],
"operator":"="}
```

Values of Filter query parameter in an encoded format

```
GET https://<MWM_Host_Name>/new/webservice
/V3/devices/clients/0/25?filter=%7B%0A%09%22property
%22%3A%20%22macaddress%22%2C%0A%09%22value%22%3A%20
%5B%22B4%3AEF%3AFA%3AD8%3AF8%3ABB%22%2C%20%22B4%3AEF
%3AFA%3AD8%3AF8%3ABB%22%5D%2C%0A%09%22operator%22%3A
%20%22%3D%22%0A%7D
```

API Request

Some API calls take parameters in the request body. These parameter values are provided in the application/JSON format except APIs where you need to upload a file, in which case the parameter value would be in the multipart/form-data format. You must ensure that the appropriate content type is set in the header before making the API call.

The JSON objects passed as parameters in the request body are required to complete the API action. Not all attributes of the JSON object are mandatory for the API call, unless explicitly specified in the documentation. Therefore, objects with a partial list of attributes can be passed in the request body of the API call and only the attribute-value pairs passed in the request body will be used to run the API call. For non-mandatory attributes that are excluded from the request body, either the default value or an existing configured value is used.

A sample request body is as follows:

```
{
  "type":"apikeycredentials",
```

```

    "keyId": "KEY-ATN59618-1",
    "keyValue": "42ff84734541cbd98f674b02555330ef"
  }

```

The JSON provided as an input should be a valid JSON. There are various application to validate JSON, such as [JSONLint](#).

API Response

The success or failure of the API response is determined by the status code. Codes in the 200 series signify success, 404 signifies not found, whereas codes in the 500 series state server errors. For a successful API call, the API response body may or may not contain any information. If the API response contains any information, it's in the application/JSON format. A sample response is as follows:

```

[ {
  "boxId": 892,
  "macAddress": "00:11:74:D4:59:9F",
  "name": "Mojo_D4:59:9F",
  "signalStrength": -33,
  "signalVariance": 0,
  "channel2_4Ghz": 0,
  "channel5Ghz": 0,
  "numClients2_4Ghz": 0,
  "numClients5Ghz": 0,
  "locationId": {
    "type": "locallocationid",
    "id": 2
  }
} ]

```

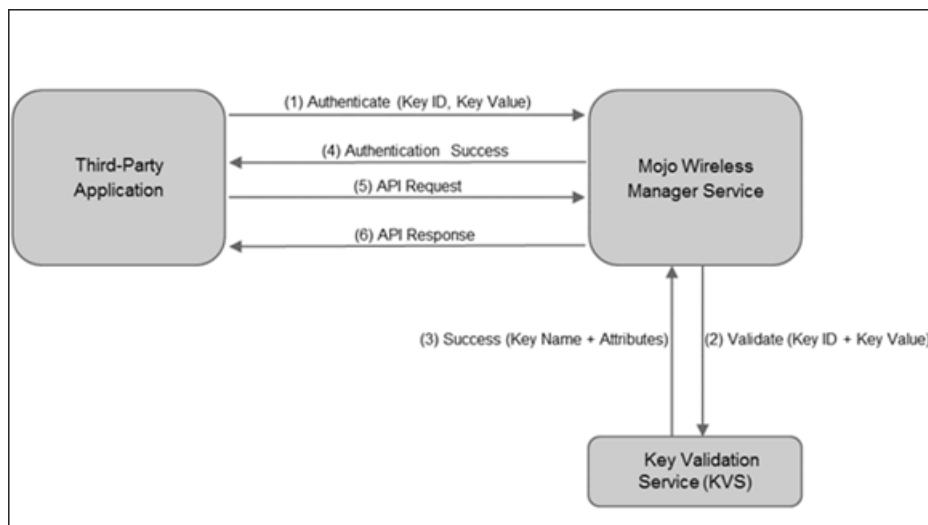
Mojo APIs can return compressed responses if the request header contains Accept-Encoding as `gzip`, `deflate`, or combination of both. The compression schemes supported are `gzip` and `deflate`. The default compression scheme used is `gzip`. The following table represents the possible values for the Accept-Encoding field and the resultant response header:

Request Header	Compression	Response Header
<code>gzip, deflate</code>	Yes	<code>gzip</code>
<code>gzip</code>	Yes	<code>gzip</code>
<code>deflate</code>	Yes	<code>deflate</code>

How MOJO APIs Work

Mojo Cloud provides support for a Key Validation Service (KVS). With KVS support, Mojo Cloud users can generate key-value pairs in Mojo Launchpad and assign appropriate service privileges on this key-value pair. This key-value pair can then be used to access the Mojo Launchpad, Mojo Wireless Manager, and Mojo Guest Manager services through REST APIs and perform various actions. The key-value pair cannot be used to log into a service from UI and is exclusively for accessing the services through APIs.

KVS support is applicable only for Mojo Cloud users. The following image illustrates the flow of request and response of API calls when using a key-value pair for authentication.

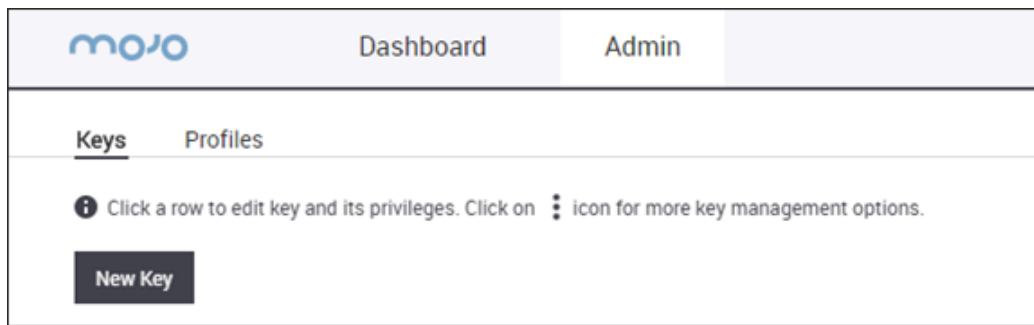


Create a Key and Assign Service Privileges

Before you begin using the Mojo REST APIs, you must create a key and assign privileges on various services to the key from Mojo Launchpad.

To create a key and assign service privileges, perform the following steps:

1. Log in to Mojo Launchpad.
2. Click **Admin** and then click **Keys**.
3. Click **New Key**.



4. Type the key name and press **Enter**.

The screenshot shows the 'New Key' form. The 'Name' field is highlighted with a red border and contains the text 'SampleKey'. A mouse cursor is shown clicking the 'Enter' key on a keyboard icon.

5. Click the three vertical dots icon from the last column for the key and select **Service Privileges**.

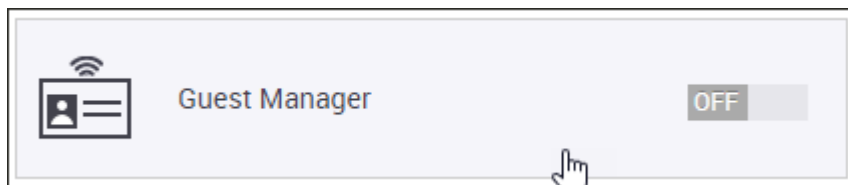
Name	ID	Value	Profile	
SampleKey	KEY-ATN559040-332	*****	Custom	⋮
SampleAPIKey	KEY-ATN559040-319	*****	Custom	⋮
MWMDocumentation	KEY-ATN559040-313	*****	Custom	⋮

Service Privileges
 Reset Value
 Lock / Unlock
 Delete

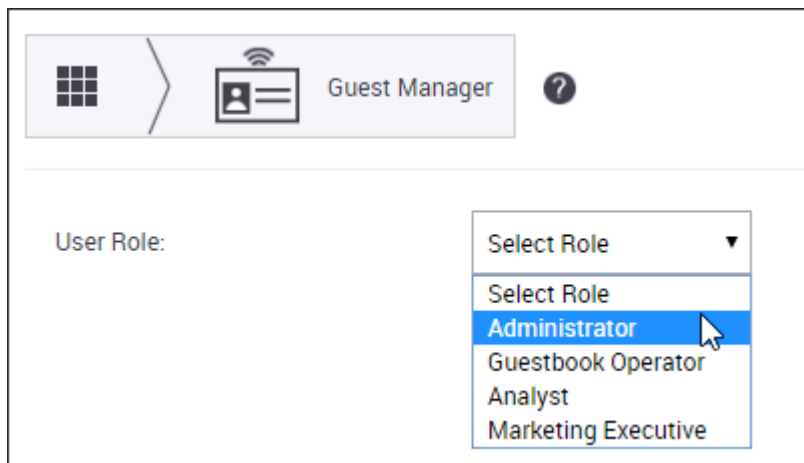
6. Select the Custom profile to assign customized privileges on each service to the key.

The screenshot shows a dropdown menu with 'Custom' selected. Below the dropdown, the text 'Services - Customized permissions on present services.' is displayed.

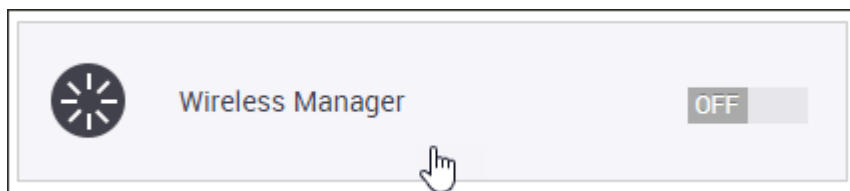
7. Click the Mojo Guest Manager tile to assign privileges on that service.



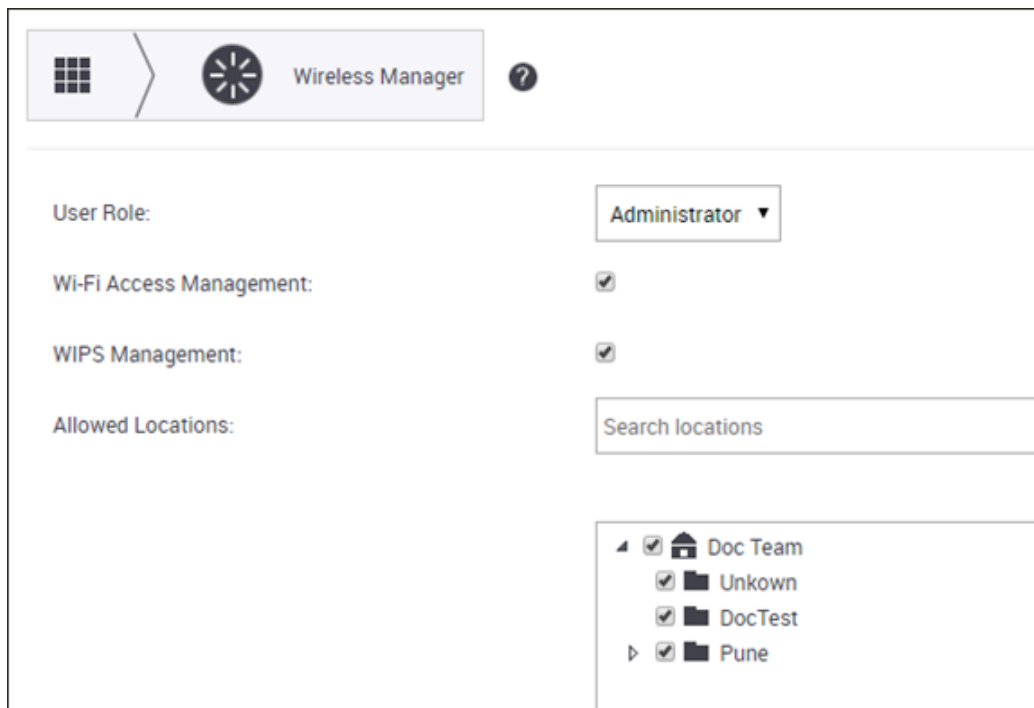
- a) Select an appropriate **User Role**.
- b) Click **Save**.



8. Click the Mojo Wireless Manager tile to assign privileges on that service.



- a) Select an appropriate **User Role**.
- b) Select the management actions that the key can perform.
- c) Select the locations on which the key has access to.
- d) Click **Save**.



Get Started

Assuming that you have already created a key-value pair from Mojo Launchpad and assigned it appropriate privileges on the various services, you are almost ready to start making API calls. You can run an API call either by using a Web browser-based REST client, such as the Mozilla Firefox REST client, or by writing a code in any programming language to leverage the Mojo REST APIs.

How to Use Mojo APIs

Before we dive into running a few API calls, let us understand what does it take to use these APIs.

The points to remember are:

- You need to create a key.
- You must assign appropriate privileges on the services to the key.
- You must define the content header before running an API call.
- You must log into each service using a key id-value pair before you run any API call on that service.

Understanding the Mojo Cloud services and what role each service plays would help you in leveraging the Mojo REST APIs.

Using a Web Browser-Based REST Client

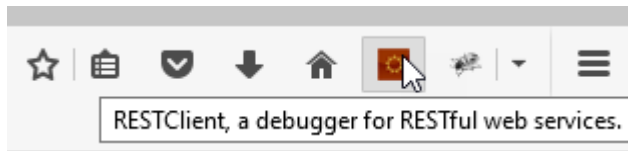
If you are just looking to try a few Mojo REST APIs to see how they work and the output that they generate, you can use any Web browser-based REST client. Here we will cover the following:

- [Add REST Client to Mozilla Firefox](#)
- [Run Mojo APIs Using REST Client](#)
- [Log Into Mojo Launchpad Using Mojo REST API](#)

Add REST Client to Mozilla Firefox

To add REST Client to Mozilla Firefox, perform the following steps:

1. In Mozilla Firefox, open the <https://addons.mozilla.org/en-US/firefox/addon/restclient/> URL.
2. Click the **Add to Firefox** button.
3. Click **Install**.
4. Click **Restart**.
5. After the browser restarts, click the REST Client icon on the top-right corner of the browser.

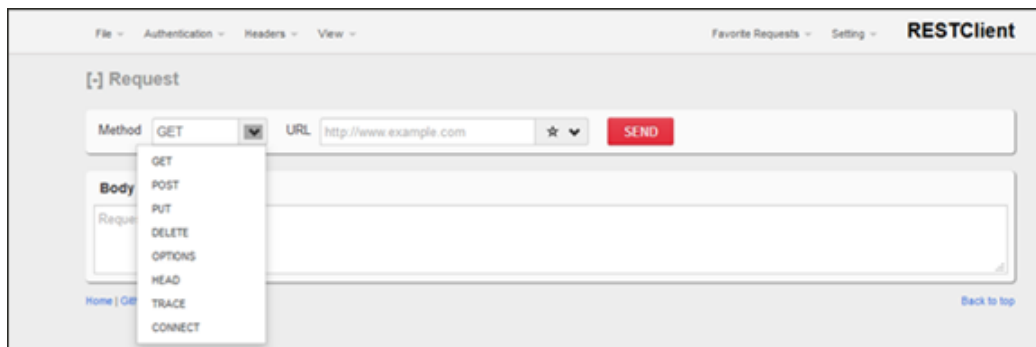


Run Mojo APIs Using REST Client

To run an API call you need REST API client. There are various browser-based REST API clients. On Chrome you can use Postman or Advanced REST Client. On Mozilla Firefox you can use the REST Client. In this document, we will cover the example using the REST Client addon for Mozilla Firefox . It is a debugger for RESTful Web services.

To run API calls from the REST Client, perform the following steps:

1. Open the REST Client.
2. Select the appropriate method from the drop-down list.



3. Enter the appropriate URL.
4. Enter a valid JSON as a request body, if required.
5. Enter Request Header. Whenever a request body is required by a call, specifying the header is mandatory. The header is used to indicate the media type of the resource.
 - a) To specify the header navigate to **Headers > Custom Header** from the menu bar.
 - b) Enter appropriate values for Request Header.

Name - Content-Type

Value - application/json

You can save the header for later use by checking **Save to favorite**.

- c) Click **Okay**.



Request Header

Name
Content-Type

Value
application/json

☐ Save to favorite

Okay Cancel

6. Verify the Request Method, URL, and Body and then click **SEND**.

After the call is executed you can see the response in Response section. Here a status code of 200 OK in response signifies successful execution of API. To view the content of the response body, click **Response Body (Preview)**.



[-] Response

Response Headers Response Body (Raw) Response Body (Highlight) Response Body (Preview)

1. Status Code	: 200 OK
2. Content-Encoding	: gzip
3. Content-Type	: application/json
4. Date	: Mon, 22 May 2017 12:23:20 GMT
5. Deprecated	: Yes
6. Server	: wifi-security-server
7. Set-Cookie	: JSESSIONID=90D7708E466031C16DF41D17F90FB4FE; Path=/news; Secure
8. Transfer-Encoding	: chunked
9. Web-API-Documentation-Link	: http://prod.airtightnetworks.com/WebAPI/source/index.html

Log In to Mojo Launchpad Using Mojo REST API


You must first log in to the server with a valid key ID-value pair before you can execute any other API call. To log in to the Mojo Launchpad using a valid key, perform the following steps:

1. Open REST Client.
2. Select the HTTP Method as **GET**.
3. Set URL as: `https://dashboard.mojonetworks.com/rest/api/v2/kvs/login?key_id=KEY-ATN014&key_value=73dccbfb4778b97fd4bf1ca1f7f99a3f6&session_timeout=60`

This API call does not require any request body parameters and, therefore, you need not specify any headers.

4. Click **SEND**.

If the API call is successful, the HTTP response status is 200.

 **Important:** Whether you are trying the API call using the REST Client on Mozilla Firefox or writing a code to leverage the Mojo REST APIs, you must ensure that you explicitly log in to each Mojo service before making an API call on the service.

Fetch Services

Mojo Launchpad can provide you with the information about the services that you have access on. Once you log into Mojo Launchpad, you can fetch the MWM and MGM services and their corresponding URLs to log into those services and make the necessary API calls.

To fetch the list of services that you have access on and their URLs, perform the following steps:

1. Open REST Client.
2. Select HTTP method as **GET**.
3. Set URL as: `https://dashboard.mojonetworks.com/rest/api/v2/services?type=amc`

Here, `type` is a query parameter with possible values as:

- `gms` - Mojo Guest Manager.
- `amc` - Mojo Wireless Manager.

This API call does not require any request body parameters and, therefore, you need not specify any headers.

4. Click **SEND**.

If the API call is successful, the HTTP response status is 200. The response body contains all the services of user in application/json format.

A sample response with the Mojo Guest Manager service URL highlighted is shown below:

```
{
  "success": true,
  "message": "Record(s) Found",
  "data":
  {
    "totalCount": 2,
    "customerServices":
    [
      {
        "id": 2693582,
        "customer_service_name": "Guest Manager",
        "show_tile": true,
        "service":
        {
          "service_id": 78,
          "service_url": "https://mgm-at21001.mojonetworks.com",
          "service_type_id": 3
        }
      }
    ],
  },
}
```

A sample response with the Mojo Wireless Manager service URL highlighted is shown below:

```
{
  "id": 2722983,
  "customer_service_name": "Wireless Manager",
  "show_tile": true,
  "service": {
    "service_id": 227,
    "service_url": "https://amc-at11009.mojonetworks.com",
    "service_type_id": 2
  }
}
```

Log In to Mojo Wireless Manager Using Mojo REST API

To fetch services for Mojo Wireless Manager you need to explicitly login using Mojo Wireless Manager login API call.

A valid key id-value pair is required for successful login. The key id and value is generated from Mojo Launchpad and provided as query parameter in this call. To generate a valid key id and value pair refer API call [Log Into Mojo Launchpad Using Mojo REST API](#). A sample Key id and value are:

```
key_id=KEY-ATN59618-1
key_value=42ff84734541cbd98f674b02555330ef
```

To log in to the Mojo Wireless Manager, perform the following steps:

1. Open REST Client.
2. Select the HTTP Method as **POST**.
3. Set URL as: `https://<MWM_Host_Name>/new/webservice/login/key/modScanWifi/3600?getClusterChildrenData=true`
4. Set Header as Content-Type = application/json
5. Set Request Body Parameters as :

```
{
  "type": "apikeycredentials",
  "keyId": "KEY-ATN59618-1",
  "keyValue": "42ff84734541cbd98f674b02555330ef"
}
```

6. Click **SEND**.

If the API call is successful, the HTTP response status is 200.

Get Mojo Wireless Manager Server Version

This API is used to obtain the version of the Mojo Server. In the case of an Mojo Cloud Services user, this API call will fetch the version of the server on which the customer account is provisioned.

To fetch Mojo Wireless Manager Server Version, perform the following steps:

1. Open REST Client.
2. Select the HTTP Method as **GET**.
3. Set URL as: `https://<MWM_Host_Name>/new/webservice/V3/system/version?locationid=1855`

This API call does not require any request body parameters and, therefore, you need not specify any headers.

4. Click **SEND**.

If the API call is successful, the HTTP response status is 200. The response body contains the details of the server version.



Get Mojo Wireless Manager API Version

This API is used to obtain the API version supported by the system.

To fetch Mojo Wireless Manager API Version, perform the following steps:

1. Open REST Client.
2. Select the HTTP Method as **GET**.
3. Set URL as: `https://<MWM_Host_Name>/new/webservice/V3/system/apiversion`

This API call does not require any request body parameters and, therefore, you need not specify any headers.

4. Click **SEND**.

If the API call is successful, the HTTP response status is 200.
The response body contains the details of the API version.



Log In to Mojo Guest Manager Using Mojo REST API

To fetch services for Mojo Guest Manager you need to explicitly login using Mojo Guest Manager login API call.

A valid key id-value pair is required for successful login. The key id and value is generated from Mojo Launchpad and provided as query parameter in this call. To generate a valid key id and value pair refer API call [Log Into Mojo Launchpad Using Mojo REST API](#). A sample Key id and value are:

```
key_id=KEY-ATN13-  
key_value=14ed8519bf636ed49f47e9b0038133aa
```

To log in to the Mojo Guest Manager, perform the following steps:

1. Open REST Client.
2. Select the HTTP Method as **GET**.
3. Using keyid and value pair, set URL as: `https://<MGM_Host_Name>/api/v1.13/site/keyLogin?key_id=KEY-ATN13-&key_value=14ed8519bf636ed49f47e9b0038133aa`
This API call does not require any request body parameters and, therefore, you need not specify any headers.
4. Click **SEND**.

If the API call is successful, the HTTP response status is 200.

Get Mojo Guest Manager Server Version

This API returns Mojo Wireless Manager's API build number, version number and API version of the deployed Mojo Guest Manager build.

To fetch Mojo Guest Manager Server Version, perform the following steps:

1. Open REST Client.
2. Select the HTTP Method as **GET**.
3. Set URL as: `https://<MGM_Host_Name>/api/site/version`
This API call does not require any request body parameters and, therefore, you need not specify any headers.
4. Click **SEND**.

If the API call is successful, the HTTP response status is 200. The response body contains the details of the server version.



Few Sample Mojo API Calls

Assuming that you have logged into the Mojo Launchpad and Mojo Wireless Manager services and have an active session running, you can perform various API call on these services.

To explore that various types of calls that can be made on these services, we cover the following examples:

- [Get Observing Managed Devices for a Client](#)
- [Get the Location Tree](#)
- [Add a Location](#)
- [Modify a Location](#)
- [Delete a Location](#)

Note: For the sake of simplicity and ease of running the API calls, we will continue using the REST Client add-on for Mozilla Firefox to cover the sample API calls.

Get Observing Managed Devices For a Client

Mojo Wireless Manager provides various functionality, such as managing the clients, access points, and sensors in your deployment. We will see how to fetch the Mojo APs that are observing a client. You must know the box ID of the client to fetch this information. You can get the box ID of a client from the response JSON of the [Get Clients](#) or [Get Paged List of Clients](#) API calls.

To fetch the Mojo APs observing a client, perform the following steps:

1. Open REST Client.
2. Select HTTP method as GET.
3. Set URL as: `https://<MWM_Host_Name>/new/webservice/V3/devices/clients/{boxid}/observingmanageddevices`

This API call does not require any request body parameters and, therefore, you need not specify any headers.

4. Click **SEND**.

If the API call is successful, the HTTP response status is 200. The response body contains list of managed devices in application/json format.

Get the Location Tree

Mojo Wireless Manager provides functionality, to retrieve the location tree. The location tree can be retrieved for logged in user and contains information regarding parent location as well as its corresponding child locations. The below call explains the procedure to retrieve the location tree for logged in users.

1. Open REST Client.
2. Select HTTP method as **GET**.
3. Set URL as: `https://<MWM_Host_Name>/new/webservice/v2/locations/tree`

This API call does not require any request body parameters and, therefore, you need not specify any headers.

4. Click **SEND**.

If the API call is successful, the HTTP response status is 200. The response body contains location tree details in application/json format.

Add a Location

In addition to retrieving a location tree, a logged in user can add new locations to the existing location tree. To add a location type and id of a parent location are passed as request body parameters. You can get required information of a parent location from [Get the Location Tree](#) call.

1. Open REST Client.
2. Select HTTP method as **PUT**.
3. Set URL as: `https://<MWM_Host_Name>/new/webservice/v2/locations`
4. Set Header as Content-Type = application/json
5. Set Request Body Parameters as :

```
{
  "parentLocation":{
    "type": "locallocationid",
    "id": 0
  },
  "location":{
    "type": "folderlocation",
    "id":{
      "type": "locallocationid"
    },
    "name": "MyLocation01",
    "timezoneId": "Asia/Calcutta"
  }
}
```

6. Click **SEND**.

If the API call is successful, the HTTP response status is 200. The response body contains details of newly added location in application/json format.

Modify a Location

In Mojo Wireless Manager information of an existing location can be modified. Details of the existing location are passed as request body parameters. You can get the details of an existing location

from [Get the Location Tree](#) call. To update information of existing location perform the following steps:

1. Open REST Client.
2. Select HTTP method as **POST**.
3. Set URL as: `https://<MWM_Host_Name>/new/webservice/v2/locations`
4. Set Header as Content-Type = `application/json`
5. Set Request Body Parameters as:

```
{
  "locationId":{
    "type": "locallocationid",
    "id": 3870
  },
  "name": "Location-Rebranded",
  "timezoneId": "Asia/Calcutta",
  "applyRecursive": false
}
```

6. Click **SEND**.

If the API call is successful, the HTTP response status is 200. The response body contains details of the location updated.

Delete a Location

An existing location and its child locations, if any can be deleted using this API call. Details of a location to be deleted are passed as request body parameters. You can get the details of an existing location from [Get the Location Tree](#) call. To delete location perform the following steps:

1. Open REST Client.
2. Select HTTP method as **DELETE**.
3. Set URL as: `https://<MWM_Host_Name>/new/webservice/v2/locations`
4. Set Header as Content-Type = `application/json`
5. Set Request Body Parameters as:

```
{
  "type": "locallocationid",
  "id": 14
}
```

6. Click **SEND**.

If the API call is successful, the HTTP response status is 204.

Sample Codes Leveraging Mojo REST APIs

For the sake of better understanding of how to use the Mojo REST APIs, we have written a few sample codes and hosted them on GitHub.

You can access the GitHub account from <https://github.com/MojoNetworksInc>.

The account contains different repositories of codes in different programming languages. You can access these repositories to go through the sample codes and understand how the Mojo REST APIs are leveraged.