Deploying APIs to a Ruckus Smartzone Controller

This document is to help in the use and deployment of API call to a Ruckus Smartzone 3.4 Controller. The below link is a reference to the locations we will be using as well as a host of other API calls you will ever need. In the location, the first part is a version folder location. This is specific to versions of Smartzone. Here is an example:

Smartzone version 3.2 - /v3_0

Smartzone version 3.4 - /v4_0

Smartzone API Reference 3.4:

http://docs.ruckuswireless.com/vscg-enterprise/vsz-e-public-api-reference-guide-3-4.html

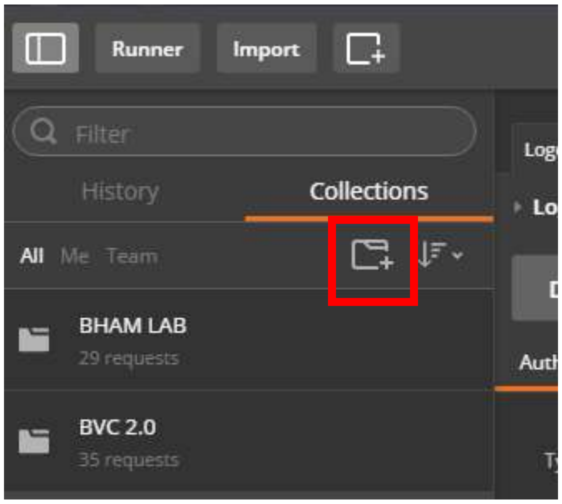
To deploy API commands to a Smartzone controller, there are several ways to accomplish this. For this demonstration, we will be using Postman by Google.

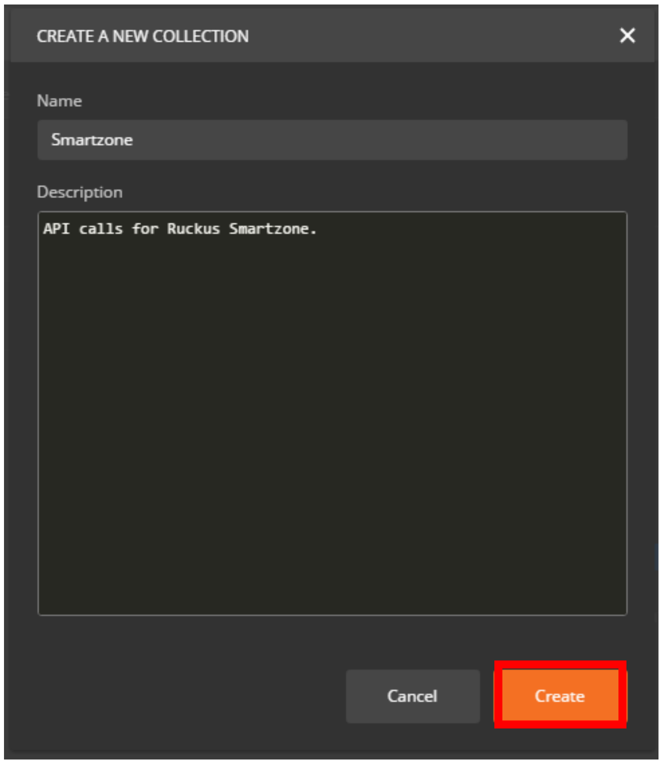
First, you will need to download and install Postman:

https://www.getpostman.com/

Once you have Postman installed, create a new Collection. This is where we will keep our API Calls.

Create a Collection:




CREATE A NEW COLLECTION

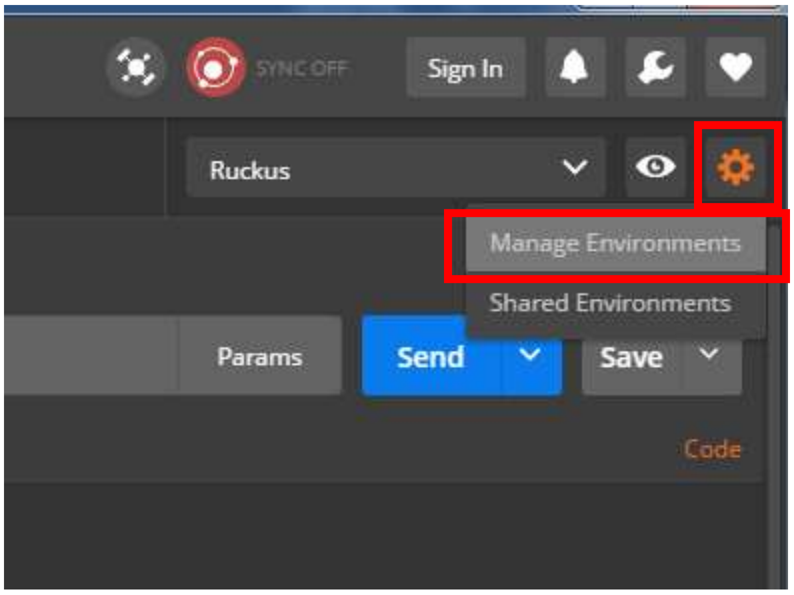
Name: Smartzone

Description:

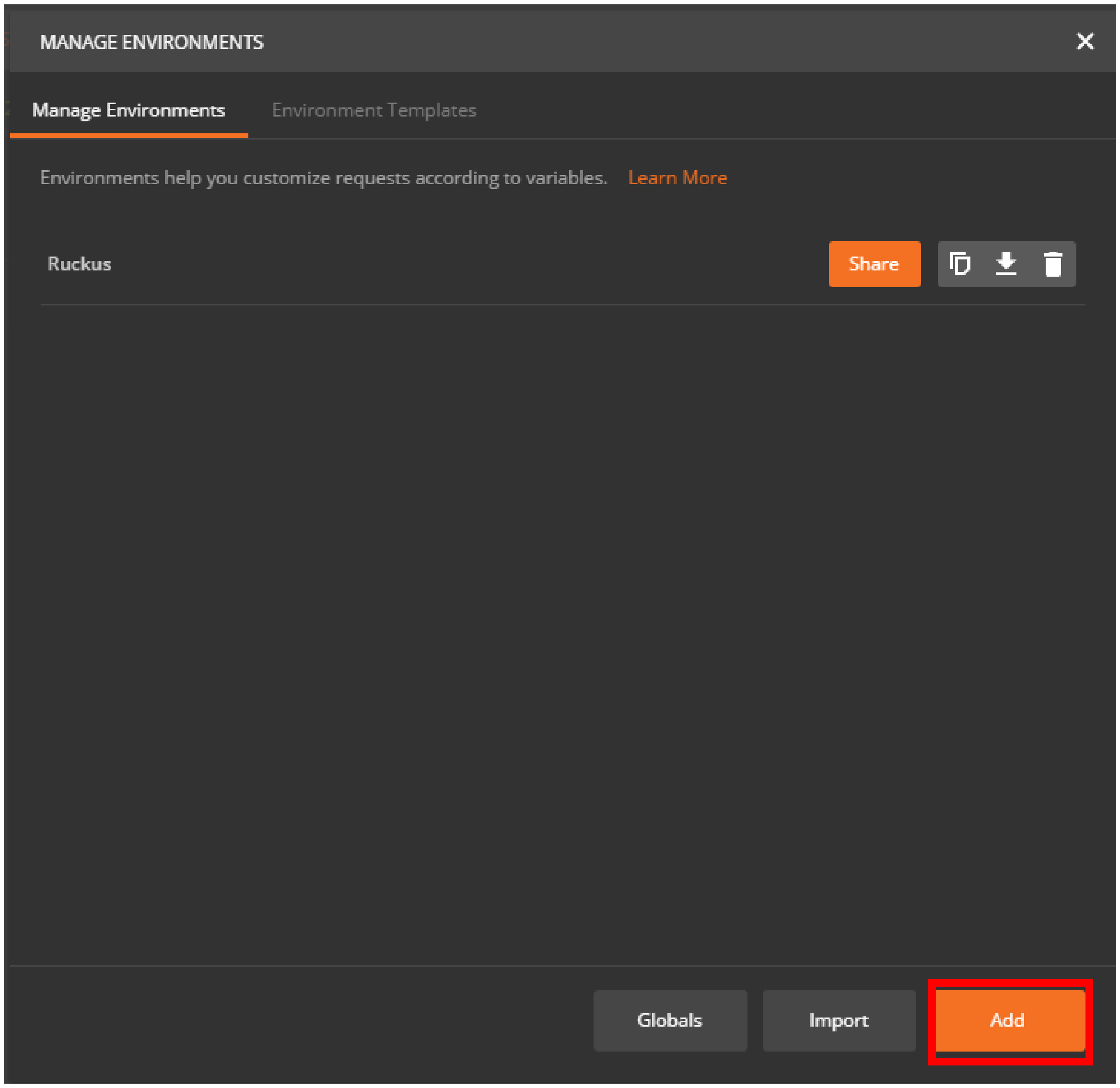
```
API calls for Ruckus Smartzone.
```

Create an Environment:

Select the Gear and Manage Environments:



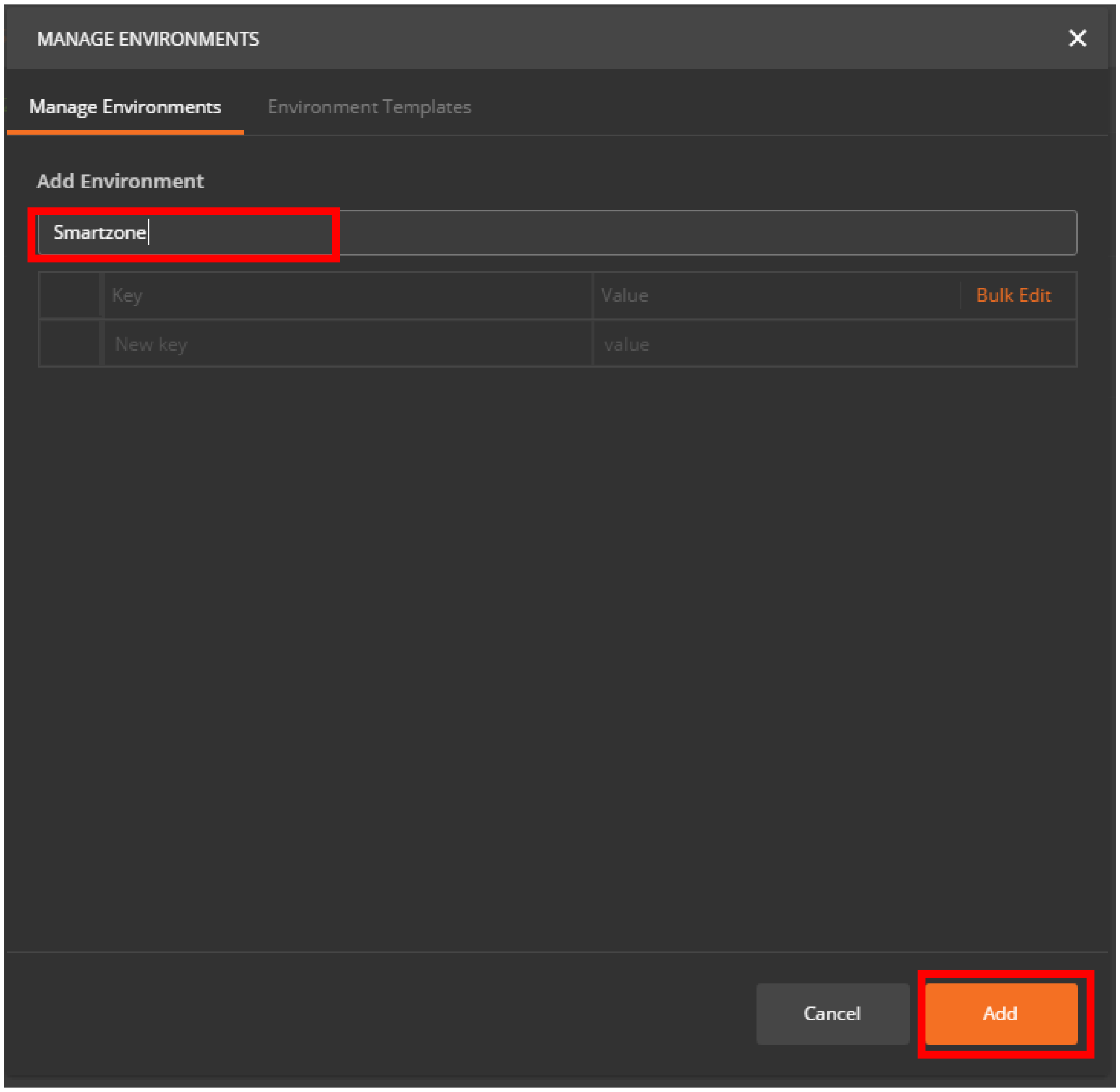
Select Add:



Environments help you customize requests according to variables. Learn More

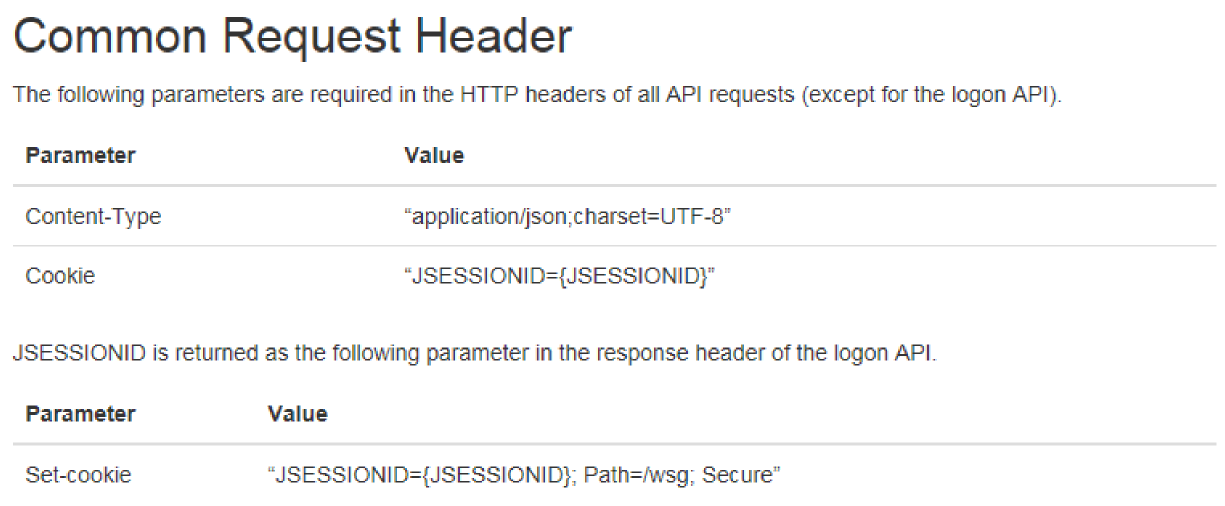
Ruckus

Give the Environment a Name and Select Add:



Inside the Environment is where we will define variables for you API calls. The Key will be the variable name and the Value will be the value for the Key.

You will need to add two header values to every API you create. Below is an explanation and the values.



Common Request Header

The following parameters are required in the HTTP headers of all API requests (except for the logon API).

Parameter	Value
Content-Type	"application/json;charset=UTF-8"
Cookie	"JSESSIONID={JSESSIONID}"

JSESSIONID is returned as the following parameter in the response header of the logon API.

Parameter	Value
Set-cookie	"JSESSIONID={JSESSIONID}; Path=/wsg; Secure"

Create a Login Session API Call:

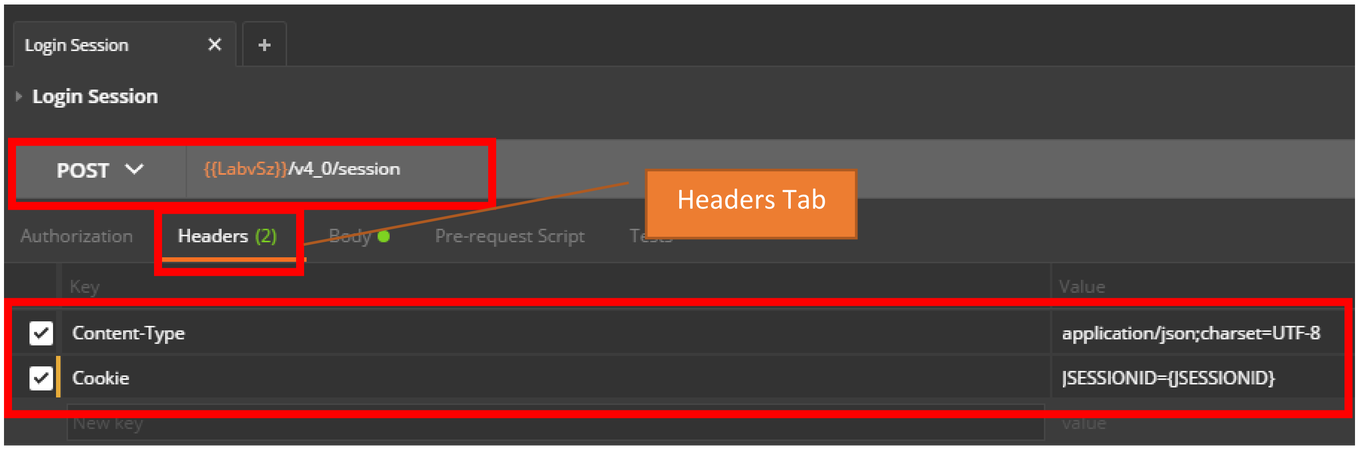
Select POST and add the destination to the controller and the location for this call. I have used a variable for the destination. Here is a break down:

{{LabvSz}} - https://{host}:7443/api/public (Where {host} is the IP Address or Hostname of your controller). If you use the {{LabvSz}}, this is a variable created in your environment.

/v4_0/session is the location.

Under the Headers Tab, add the two fields for all your API Calls:

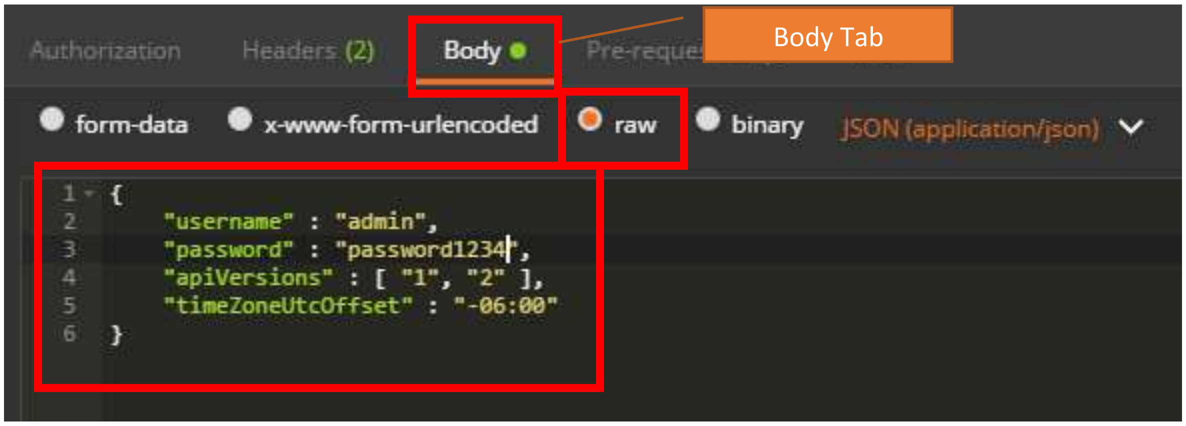
Content-Type - application/json;charset=UTF-8
Cookie - JSESSIONID={JSESSIONID}



{{LabvSz}}/v4_0/session

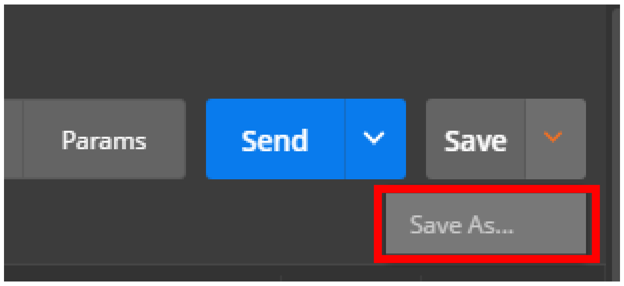
Key	Value
✓ Content-Type	application/json;charset=UTF-8
✓ Cookie	JSESSIONID={JSESSIONID}

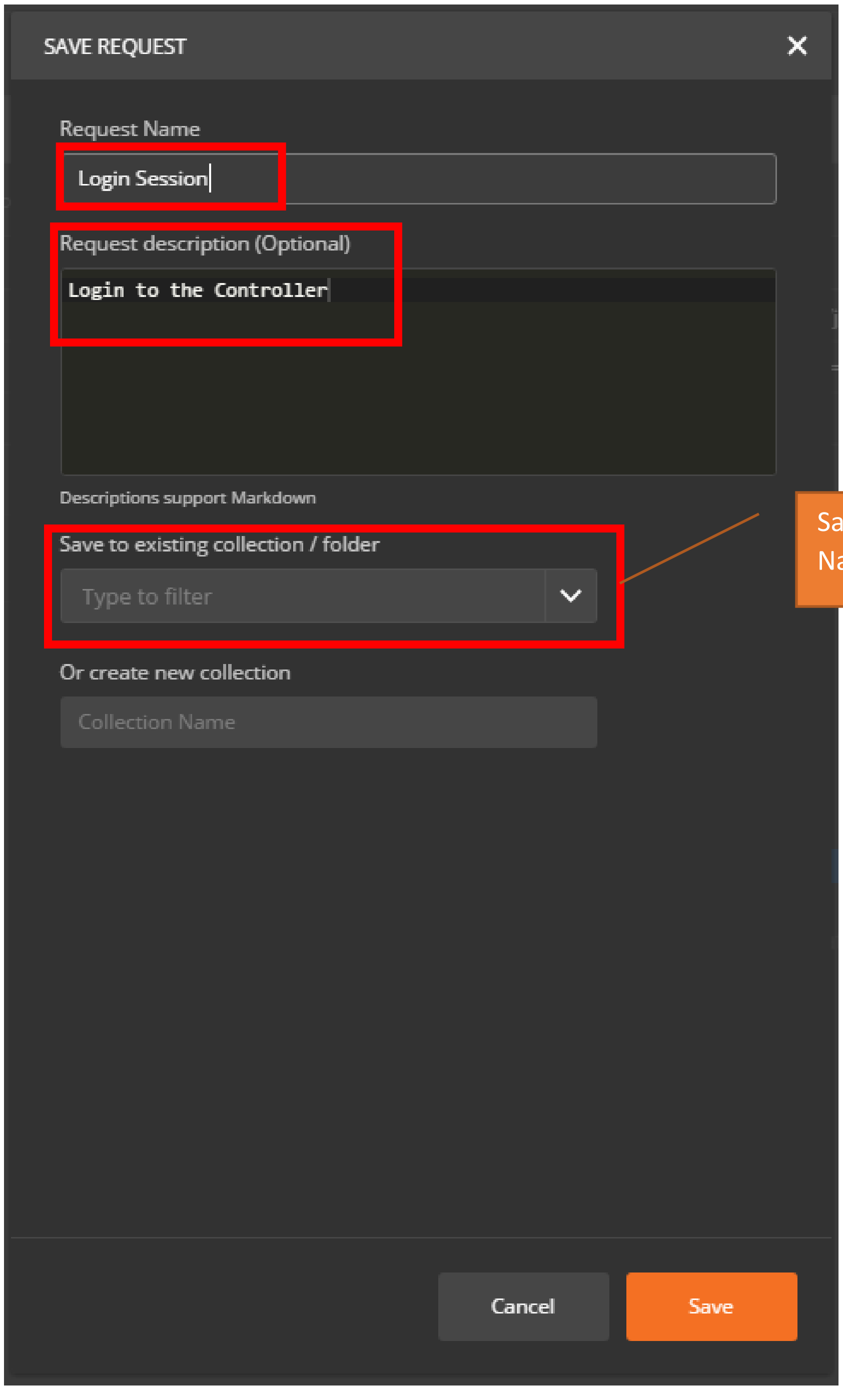
Under the Body Tab, you need to add the username and password to login to the controller. You will use the raw format and it must be formatted exactly as below. The apiVersion can be left to the default and the timeZoneUtcOffset is the time zone of the controller.



```
{
    "username" : "admin",
    "password" : "password1234",
    "apiVersions" : [ "1", "2" ],
    "timeZoneUtcOffset" : "-06:00"
}
```

When you are done, select the down arrow next to Save to do Save As.




SAVE REQUEST

Request Name

Login Session

Request description (Optional)

```
Login to the Controller
```

Descriptions support Markdown

Save to existing collection / folder

Type to filter

Or create new collection

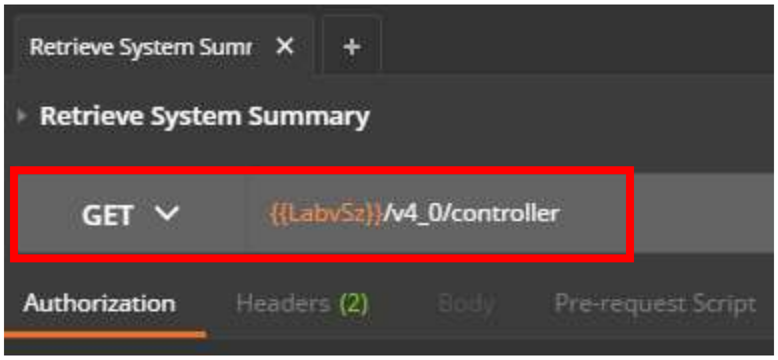
Collection Name

Save the API to the Collection Name you created earlier.

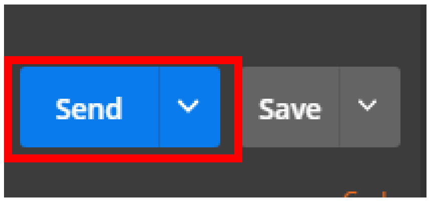
<u>Retrieve the System Summary API:</u>

Now we are going to create an API to Retrieve the System Summary.

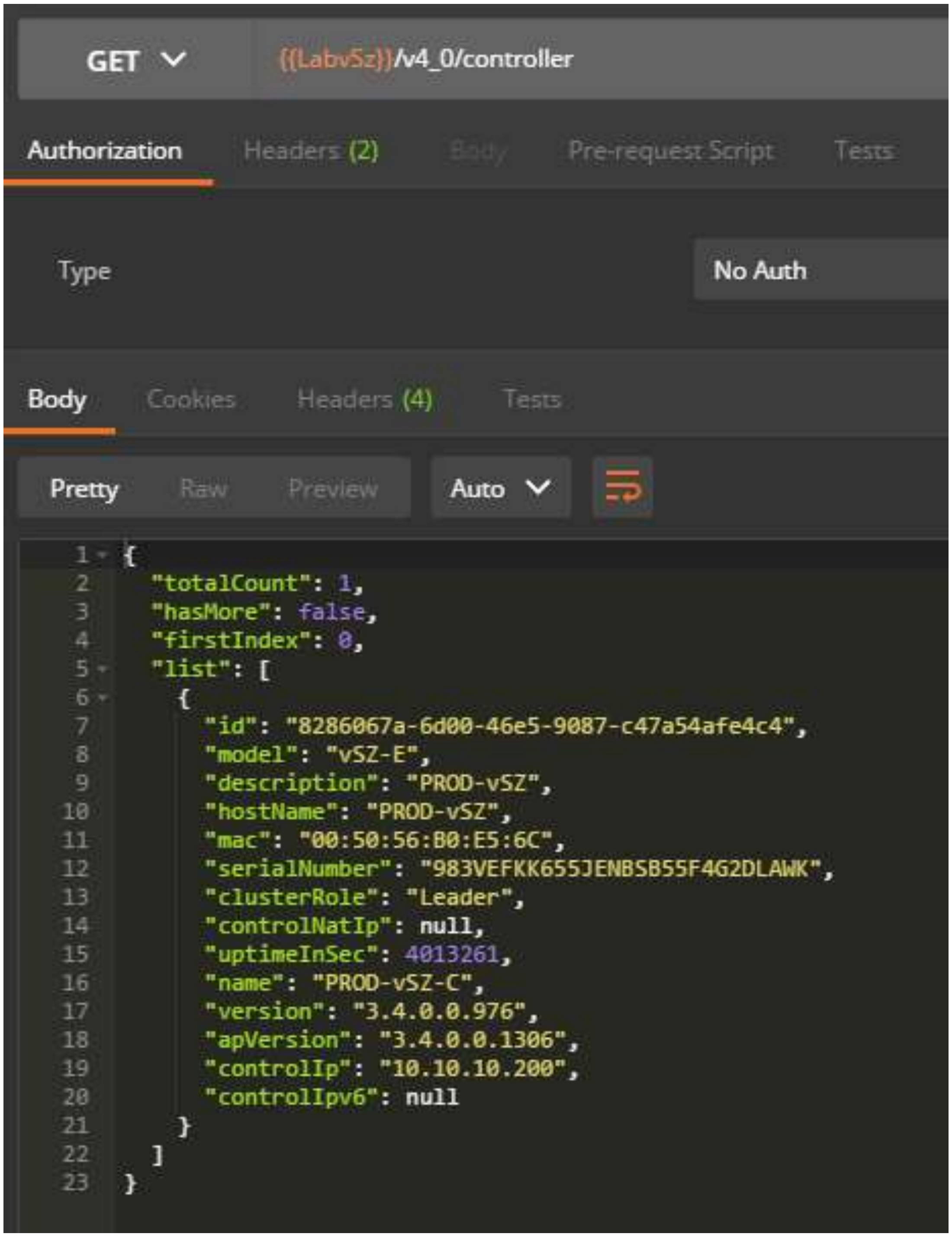
For this call Use GET and add the destination URL plus the location (/v4_0/controller). Remember to add the header information as above and Save AS when you are done.



To execute this call, select the send button.



Below is a Sample Output:



```
{{LabvSz}}/v4_0/controller
```

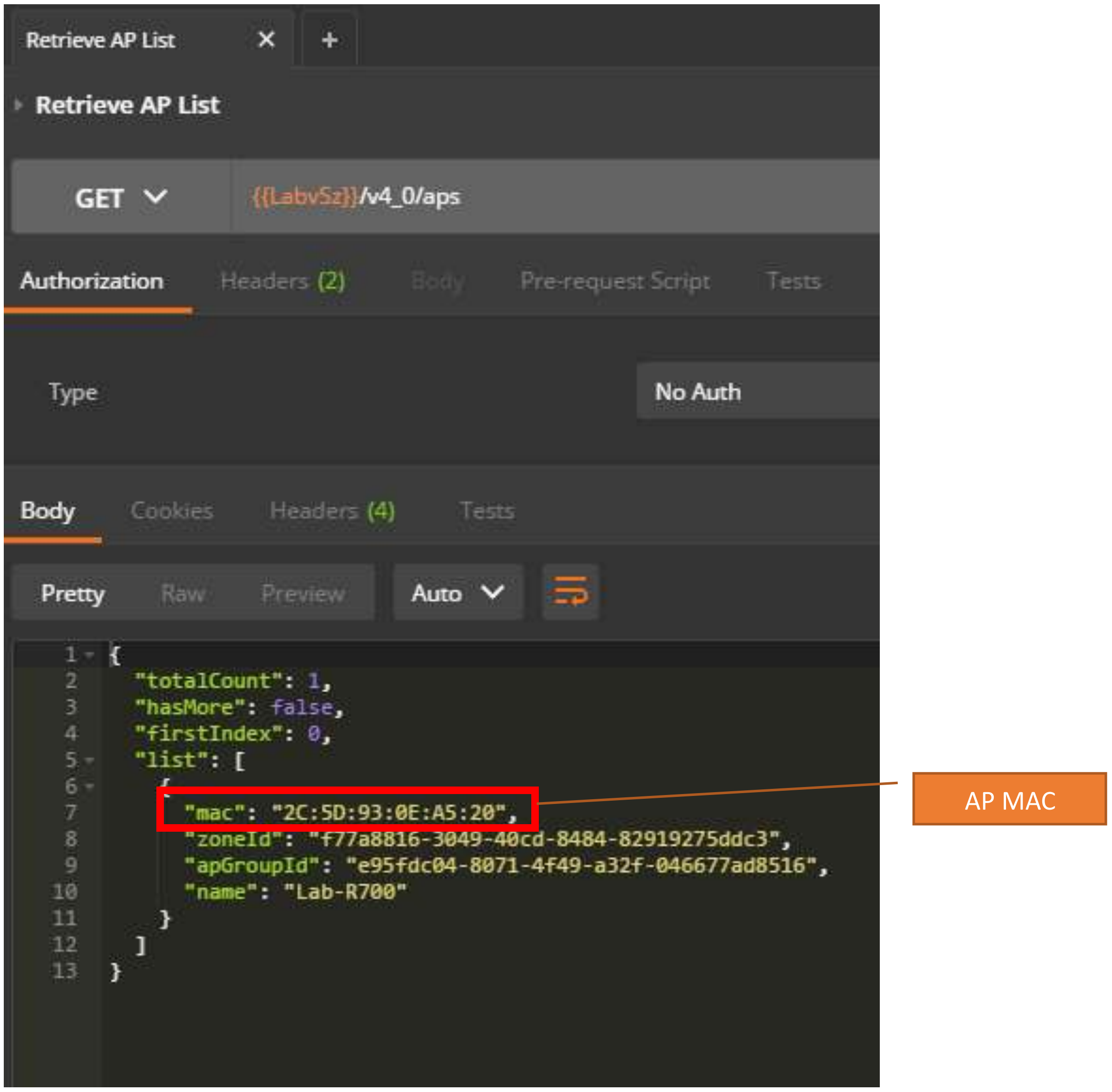
Type: No Auth

```
{
  "totalCount": 1,
  "hasMore": false,
  "firstIndex": 0,
  "list": [
    {
      "id": "8286067a-6d00-46e5-9087-c47a54afe4c4",
      "model": "vSZ-E",
      "description": "PROD-vSZ",
      "hostName": "PROD-vSZ",
      "mac": "00:50:56:B0:E5:6C",
      "serialNumber": "983VEFKK655JENBSB55F4G2DLAWK",
      "clusterRole": "Leader",
      "controlNatIp": null,
      "uptimeInSec": 4013261,
      "name": "PROD-vSZ-C",
      "version": "3.4.0.0.976",
      "apVersion": "3.4.0.0.1306",
      "controlIp": "10.10.10.200",
      "controlIpv6": null
    }
  ]
}
```

<u>Retrieve AP List API:</u>

Now we are going to create an API to retrieve a list of APs from the controller.

For this call Use GET and add the destination URL plus the location (/v4_0/aps). Remember to add the header information as above and Save AS when you are done. Below is the command and sample output.



GET {{LabvSz}}/v4_0/aps

Type: No Auth

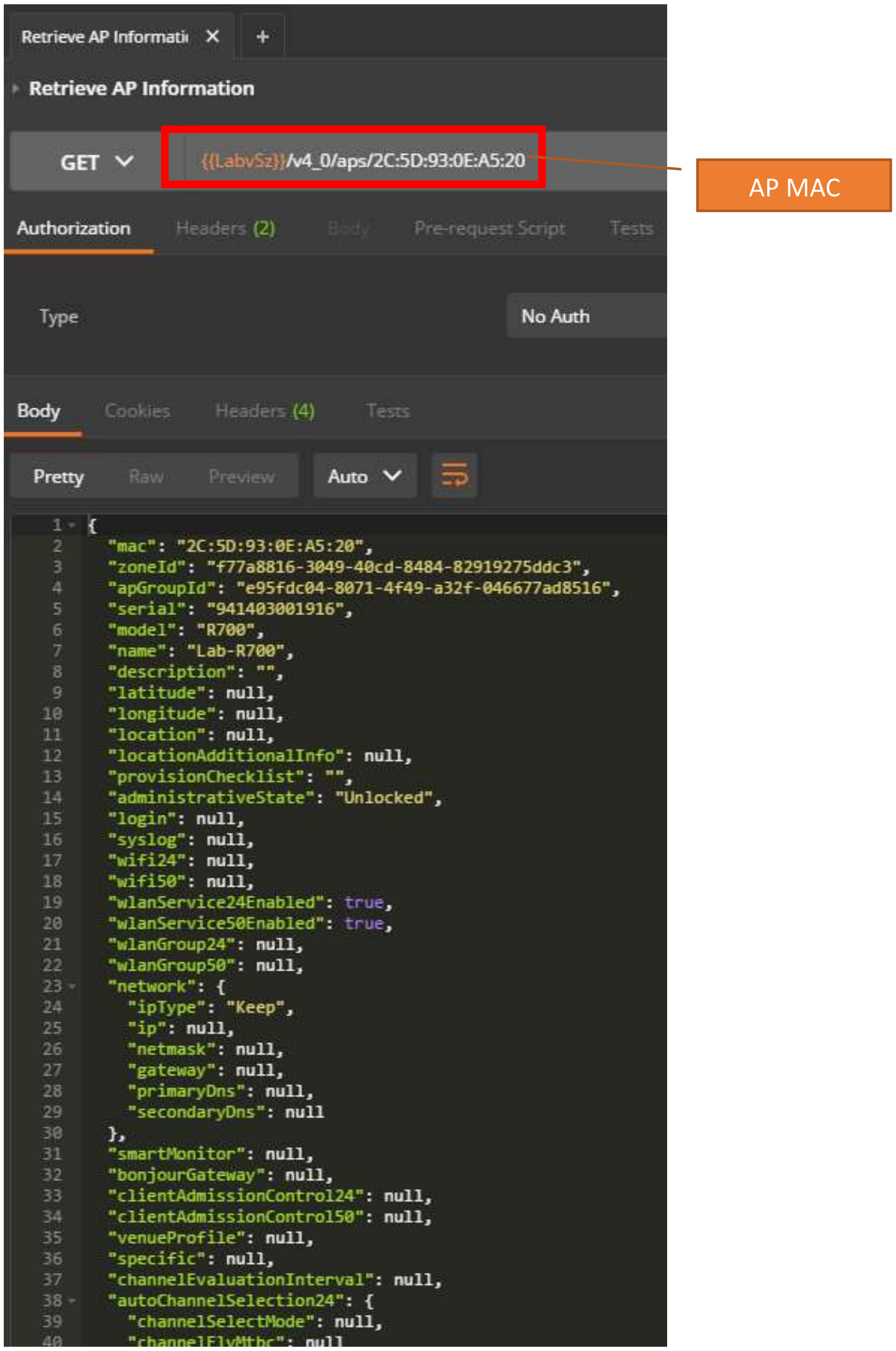
```
{
  "totalCount": 1,
  "hasMore": false,
  "firstIndex": 0,
  "list": [
    {
      "mac": "2C:5D:93:0E:A5:20",
      "zoneId": "f77a8816-3049-40cd-8484-82919275ddc3",
      "apGroupId": "e95fdc04-8071-4f49-a32f-046677ad8516",
      "name": "Lab-R700"
    }
  ]
}
```

In this example, we received a list of APs from the controller. I only have one AP. From here we can get the MAC Address of the AP to use in the next API.

<u>Retrieve AP Information API:</u>

Now we are going to create an API to retrieve information about a specific AP from the controller.

For this call Use GET and add the destination URL plus the location (/v4_0/aps/{apMac}). For this example, we need the AP MAC Address from the last example to replace {apMac}. Remember to add the header information as above and Save AS when you are done. Below is the command and sample output.



```
{{LabvSz}}/v4_0/aps/2C:5D:93:0E:A5:20
```

Type: No Auth

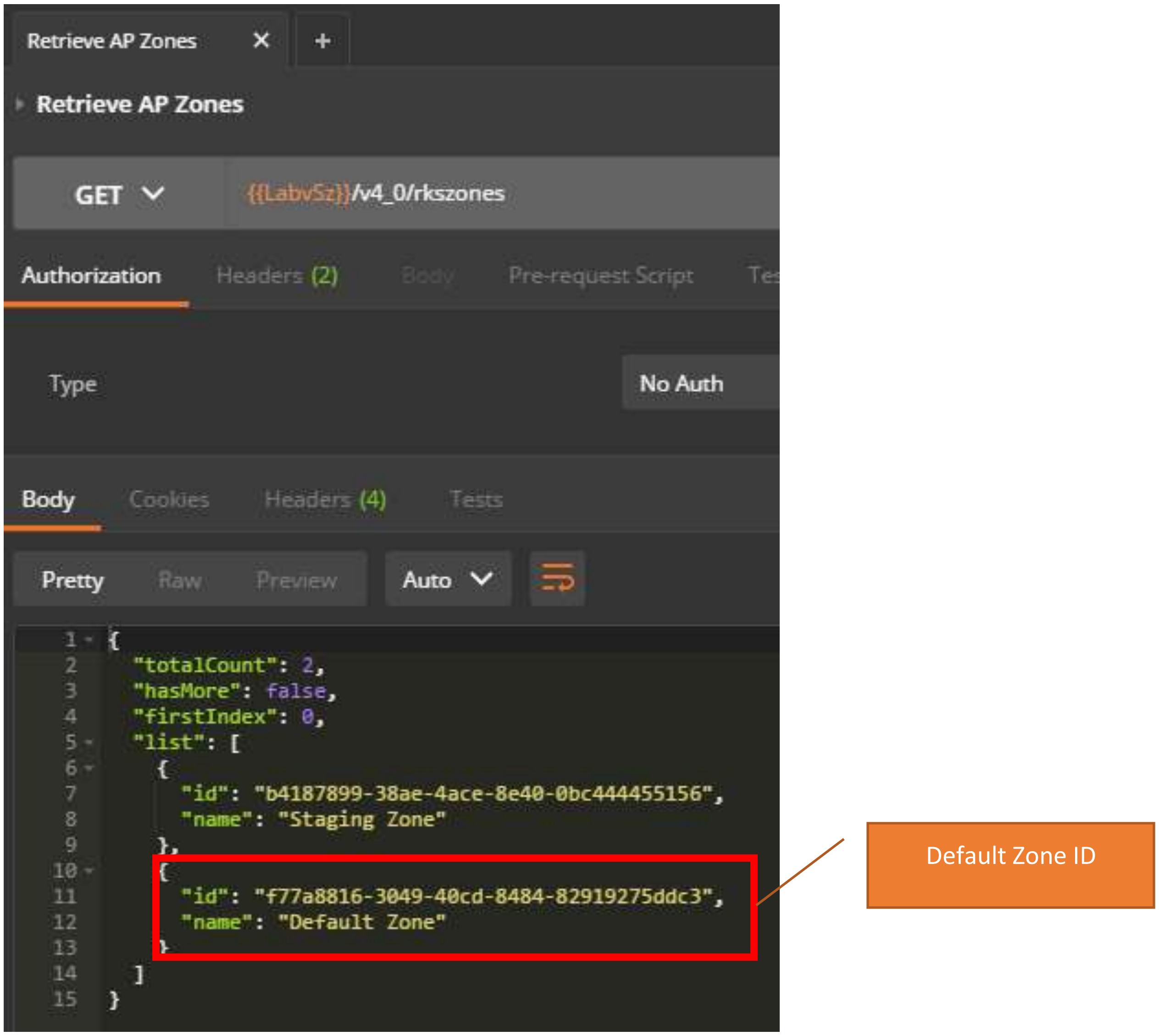
Body: Auto

```
{
  "mac": "2C:5D:93:0E:A5:20",
  "zoneId": "f77a8816-3049-40cd-8484-82919275ddc3",
  "apGroupId": "e95fdc04-8071-4f49-a32f-046677ad8516",
  "serial": "941403001916",
  "model": "R700",
  "name": "Lab-R700",
  "description": "",
  "latitude": null,
  "longitude": null,
  "location": null,
  "locationAdditionalInfo": null,
  "provisionChecklist": "",
  "administrativeState": "Unlocked",
  "login": null,
  "syslog": null,
  "wifi24": null,
  "wifi50": null,
  "wlanService24Enabled": true,
  "wlanService50Enabled": true,
  "wlanGroup24": null,
  "wlanGroup50": null,
  "network": {
    "ipType": "Keep",
    "ip": null,
    "netmask": null,
    "gateway": null,
    "primaryDns": null,
    "secondaryDns": null
  },
  "smartMonitor": null,
  "bonjourGateway": null,
  "clientAdmissionControl24": null,
  "clientAdmissionControl50": null,
  "venueProfile": null,
  "specific": null,
  "channelEvaluationInterval": null,
  "autoChannelSelection24": {
    "channelSelectMode": null,
    "channelFlyMtbc": null
```

Retrieve AP Zones API:

Now we are going to create an API to retrieve a list of AP Zones from the controller.

For this call Use GET and add the destination URL plus the location (/v4_0/rkszones). Remember to add the header information as above and Save AS when you are done. Below is the command and sample output.



Retrieve AP Zones

GET {{LabvSz}}/v4_0/rkszones

Type: No Auth

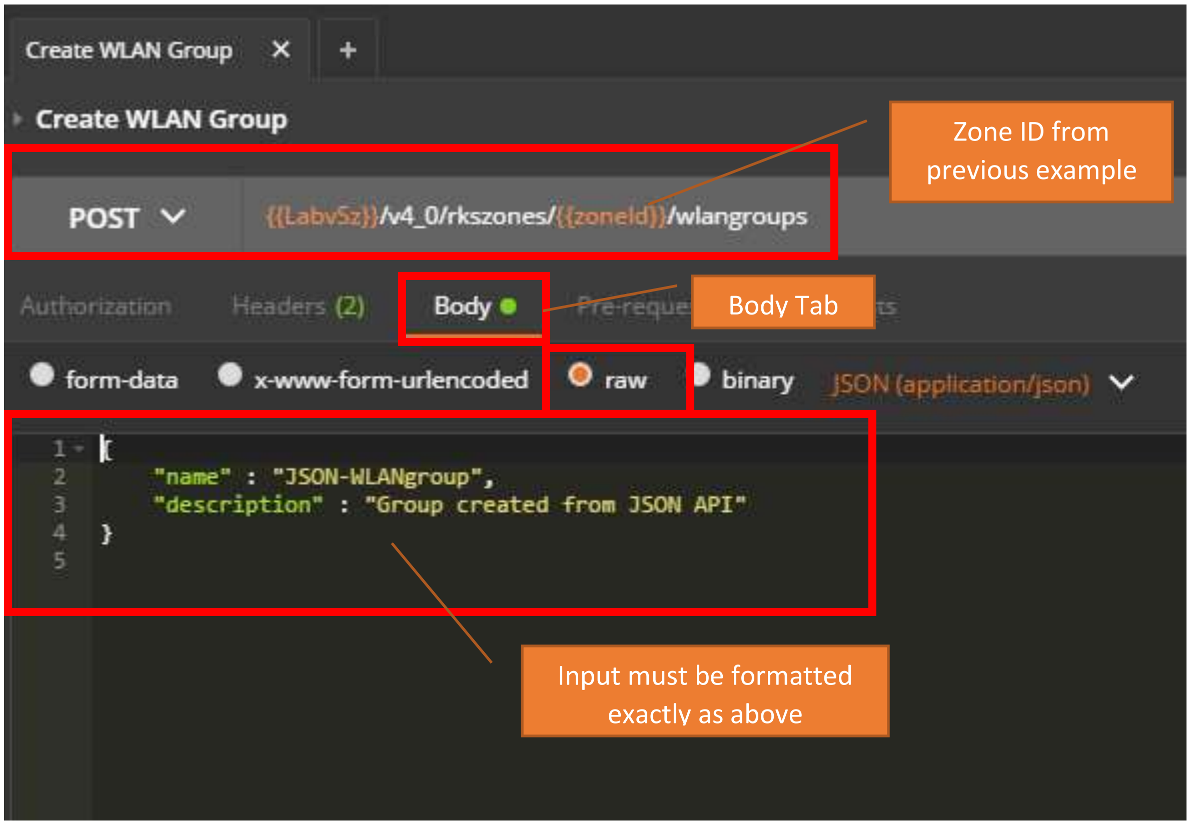
```
{
  "totalCount": 2,
  "hasMore": false,
  "firstIndex": 0,
  "list": [
    {
      "id": "b4187899-38ae-4ace-8e40-0bc444455156",
      "name": "Staging Zone"
    },
    {
      "id": "f77a8816-3049-40cd-8484-82919275ddc3",
      "name": "Default Zone"
    }
  ]
}
```

We will use the “Default Zone” ID to perform the next API.

Create WLAN Group API:

Now we are going to create an API to create a WLAN Group on the controller.

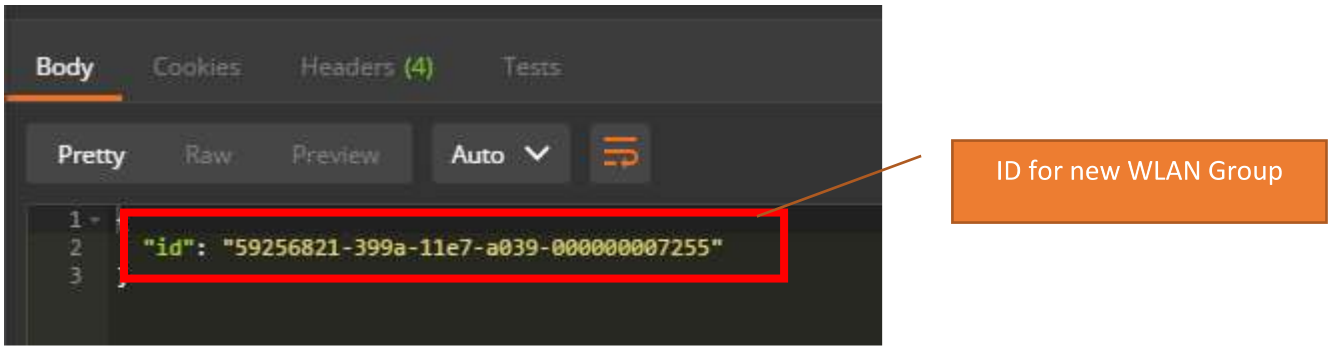
For this call Use POST and add the destination URL plus the location (/v4_0/rkszones/{zoneId}/wlangroups). Remember to add the header information as above. Here we will also have to add information to the Body Tab. The body tab has the name of the WLAN Group we want to create and the Description. This is input in the raw format. Refer to the API Guide for more details. Be sure to Save AS when you are done. Below is the command and sample output.



```
{{LabvSz}}/v4_0/rkszones/{{zoneId}}/wlangroups
```

```
{
     "name" : "JSON-WLANgroup",
     "description" : "Group created from JSON API"
}
```

When you execute this command with the Send Button, you will see the following:



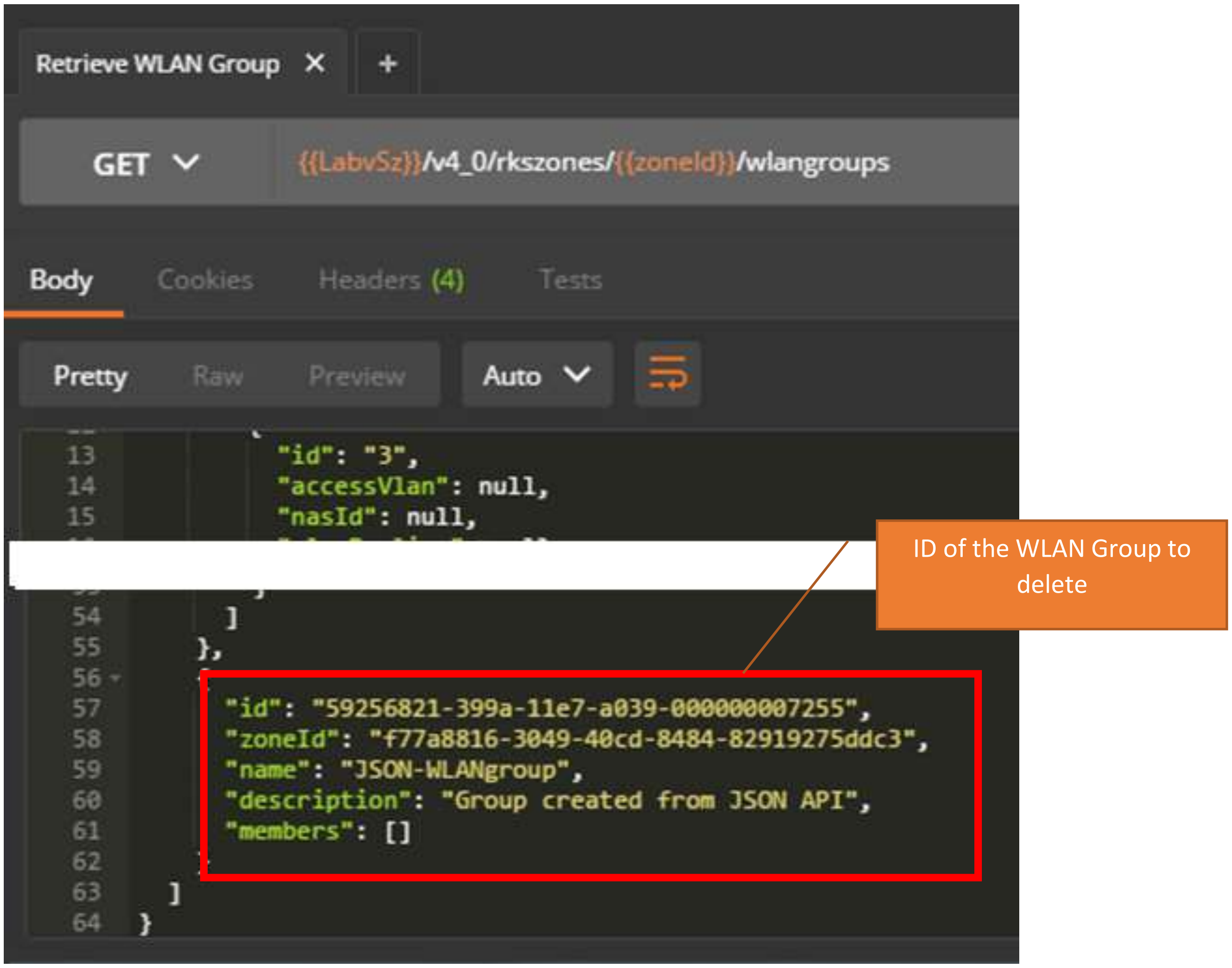
```
{
  "id": "59256821-399a-11e7-a039-000000007255"
}
```

Delete WLAN Group API:

Now we are going to create an API to delete a WLAN Group on the controller.

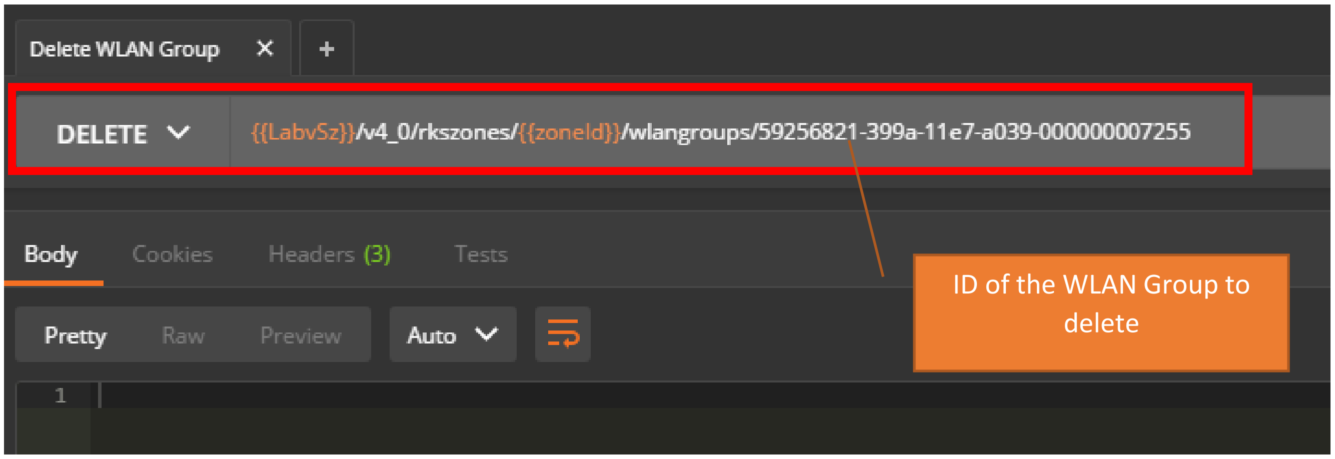
For this call Use DELETE and add the destination URL plus the location (/v4_0/rkszones/{zoneId}/wlangroups/{id}). We will use the zone ID from the previous example and the ID of the WLAN Group we want to delete. This can be found from a retrieve WLAN List API. Remember to add the header information as above and Save AS when you are done.

Below is the output from the Retrieve WLAN Group List: Total Count is 2 for my example.


Retrieve WLAN Group
GET
{{LabvSz}}/v4_0/rkszones/{{zoneId}}/wlangroups
Body
Cookies
Headers (4)
Tests
Pretty
Raw
Preview
Auto
ID of the WLAN Group to delete

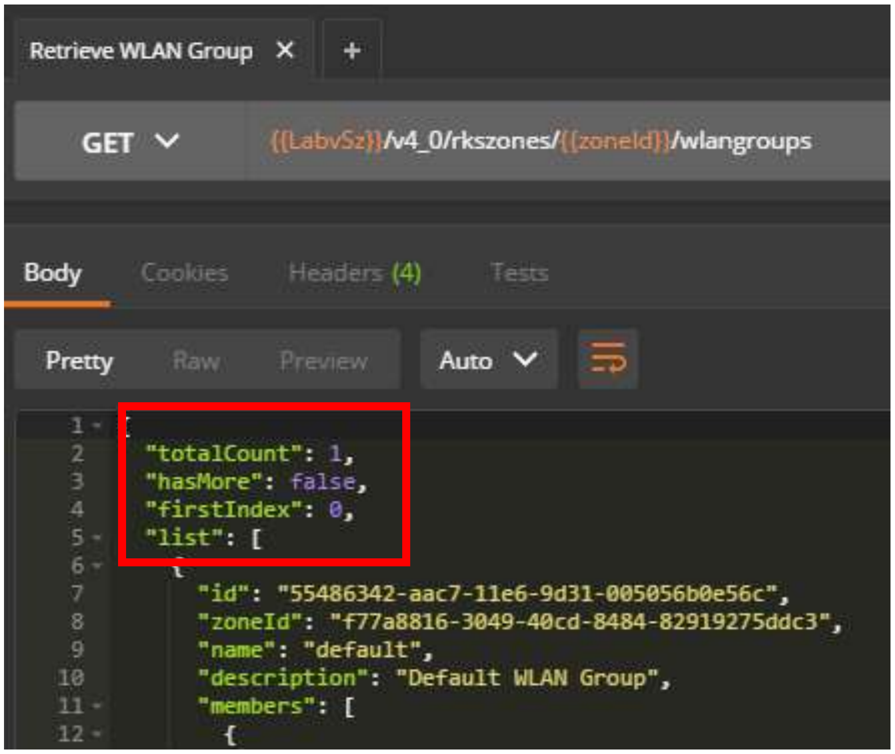
```
            "id": "3",
            "accessVlan": null,
            "nasId": null,
        ]
    },
    {
        "id": "59256821-399a-11e7-a039-000000007255",
        "zoneId": "f77a8816-3049-40cd-8484-82919275ddc3",
        "name": "JSON-WLANgroup",
        "description": "Group created from JSON API",
        "members": []
    }
  ]
}
```

Below is the command and sample output.



```
DELETE {{LabvSz}}/v4_0/rkszones/{{zoneId}}/wlangroups/59256821-399a-11e7-a039-000000007255
```

Now we run the WLAN Group Retrieve API again and the group has been deleted. Total Count is now one.

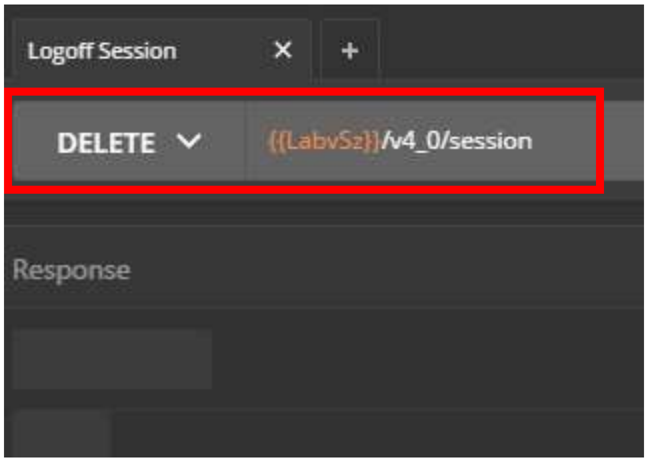


```
GET {{LabvSz}}/v4_0/rkszones/{{zoneId}}/wlangroups
```

```
{
  "totalCount": 1,
  "hasMore": false,
  "firstIndex": 0,
  "list": [
    {
      "id": "55486342-aac7-11e6-9d31-005056b0e56c",
      "zoneId": "f77a8816-3049-40cd-8484-82919275ddc3",
      "name": "default",
      "description": "Default WLAN Group",
      "members": [
        {
```

<u>Create a Logout API:</u>

When you are finished, we need to logout of the controller. For this call Use DELETE and add the destination URL plus the location (/v4_0/session). Remember to add the header information as above and Save AS when you are done.



```
{{LabvSz}}/v4_0/session
```

This is only the beginning of the APIs you can create. Please refer to the API Guide for reference.