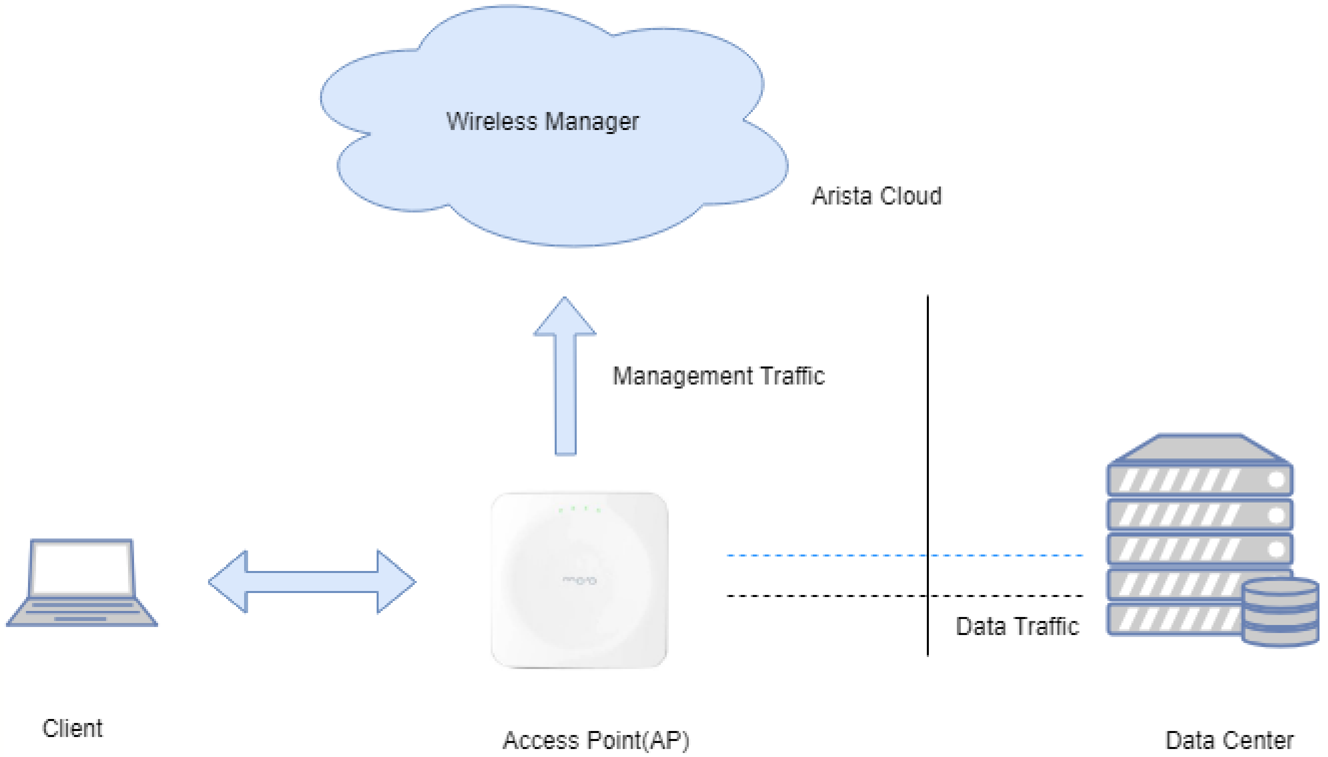
Introduction to Arista WiFi API

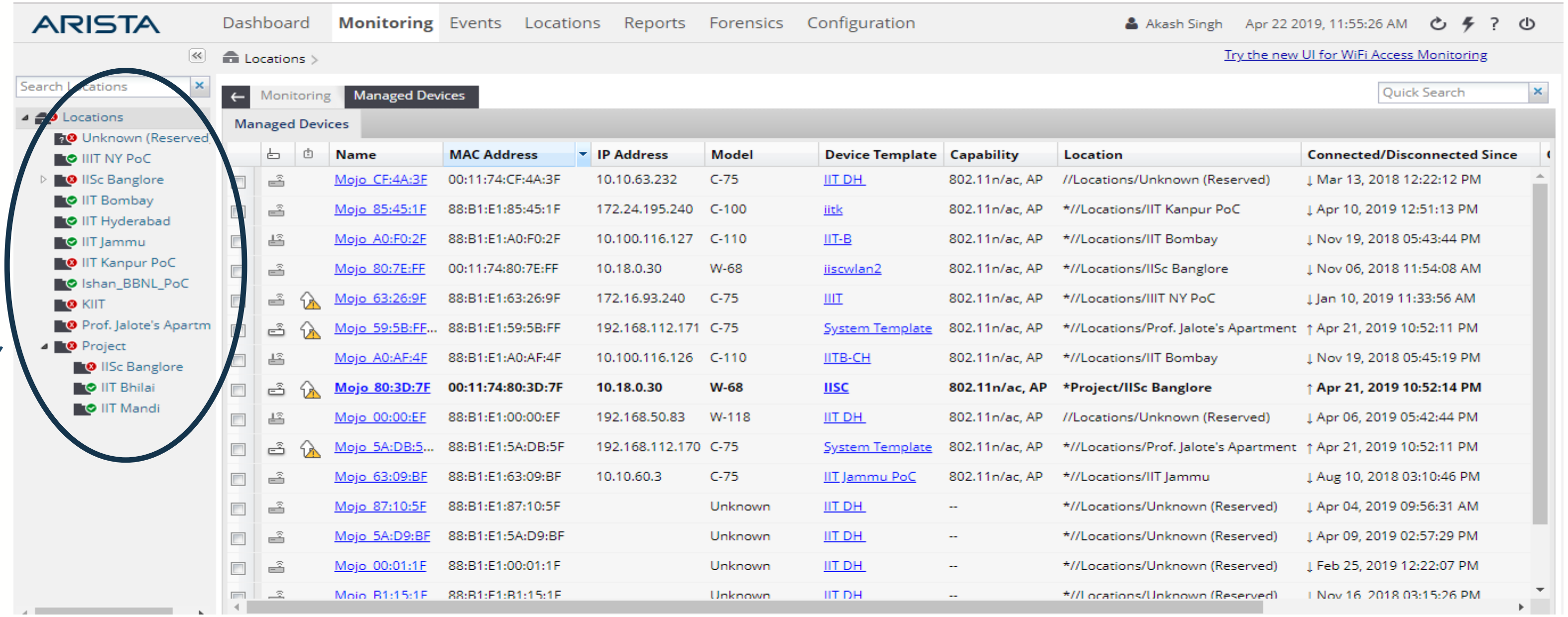
Arista Cloud Architecture



Introduction to Wireless Manager

- Wireless Manager (WM) is one of the key components of Arista's cloud-based Wi-Fi solution
  - Manages all the configurations of Access Points (AP).
  - Enables monitoring of APs with a customizable dashboard which provides hierarchical view of the network.

Introduction to Wireless Manager



Managed Devices

Name	MAC Address	IP Address	Model	Device Template	Capability	Location	Connected/Disconnected Since
Mojo_CF:4A:3F	00:11:74:CF:4A:3F	10.10.63.232	C-75	IIT DH	802.11n/ac, AP	//Locations/Unknown (Reserved)	↓ Mar 13, 2018 12:22:12 PM
Mojo_85:45:1F	88:B1:E1:85:45:1F	172.24.195.240	C-100	iitk	802.11n/ac, AP	*//Locations/IIT Kanpur PoC	↓ Apr 10, 2019 12:51:13 PM
Mojo_A0:F0:2F	88:B1:E1:A0:F0:2F	10.100.116.127	C-110	IIT-B	802.11n/ac, AP	*//Locations/IIT Bombay	↓ Nov 19, 2018 05:43:44 PM
Mojo_80:7E:FF	00:11:74:80:7E:FF	10.18.0.30	W-68	iiscwlan2	802.11n/ac, AP	*//Locations/IISc Banglore	↓ Nov 06, 2018 11:54:08 AM
Mojo_63:26:9F	88:B1:E1:63:26:9F	172.16.93.240	C-75	IIIT	802.11n/ac, AP	*//Locations/IIIT NY PoC	↓ Jan 10, 2019 11:33:56 AM
Mojo_59:5B:FF...	88:B1:E1:59:5B:FF	192.168.112.171	C-75	System Template	802.11n/ac, AP	*//Locations/Prof. Jalote's Apartment	↑ Apr 21, 2019 10:52:11 PM
Mojo_A0:AF:4F	88:B1:E1:A0:AF:4F	10.100.116.126	C-110	IITB-CH	802.11n/ac, AP	*//Locations/IIT Bombay	↓ Nov 19, 2018 05:45:19 PM
Mojo_80:3D:7F	**00:11:74:80:3D:7F**	**10.18.0.30**	**W-68**	**IISC**	**802.11n/ac, AP**	***Project/IISc Banglore**	**↑ Apr 21, 2019 10:52:14 PM**
Mojo_00:00:EF	88:B1:E1:00:00:EF	192.168.50.83	W-118	IIT DH	802.11n/ac, AP	//Locations/Unknown (Reserved)	↓ Apr 06, 2019 05:42:44 PM
Mojo_5A:DB:5...	88:B1:E1:5A:DB:5F	192.168.112.170	C-75	System Template	802.11n/ac, AP	*//Locations/Prof. Jalote's Apartment	↑ Apr 21, 2019 10:52:11 PM
Mojo_63:09:BF	88:B1:E1:63:09:BF	10.10.60.3	C-75	IIT Jammu PoC	802.11n/ac, AP	*//Locations/IIT Jammu	↓ Aug 10, 2018 03:10:46 PM
Mojo_87:10:5F	88:B1:E1:87:10:5F		Unknown	IIT DH	--	*//Locations/Unknown (Reserved)	↓ Apr 04, 2019 09:56:31 AM
Mojo_5A:D9:BF	88:B1:E1:5A:D9:BF		Unknown	IIT DH	--	*//Locations/Unknown (Reserved)	↓ Apr 09, 2019 02:57:29 PM
Mojo_00:01:1F	88:B1:E1:00:01:1F		Unknown	IIT DH	--	*//Locations/Unknown (Reserved)	↓ Feb 25, 2019 12:22:07 PM
Mojo_B1:15:1F	88:B1:E1:B1:15:1F		Unknown	IIT DH	--	*//Locations/Unknown (Reserved)	↓ Nov 16, 2018 03:15:26 PM

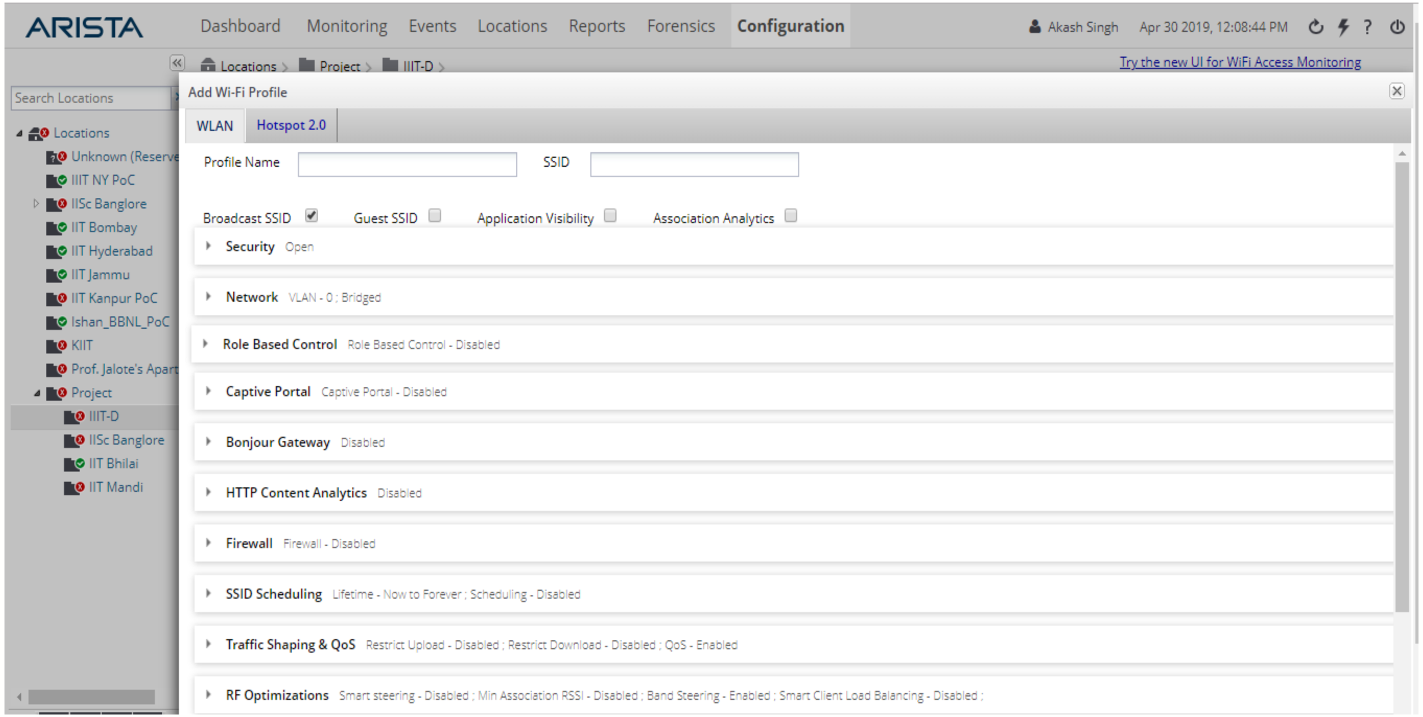
Location Hierarchy

Wireless Configuration

- All configurations on WM are based on location hierarchy.
  - Child node can inherit the configuration from Parent node.
  - Inheritance can be broken at any node/location to define configuration for that location
- All configurations on WM are object based.
- Two objects need to be defined to configure a device.
  - SSID Profile: SSID-specific parameters
  - Device Template: Radio configuration which are SSID-agnostic

Configuration: SSID Profile

- SSID Profile includes parameters which are specific to each SSID, for e.g authentication mechanism



Add Wi-Fi Profile

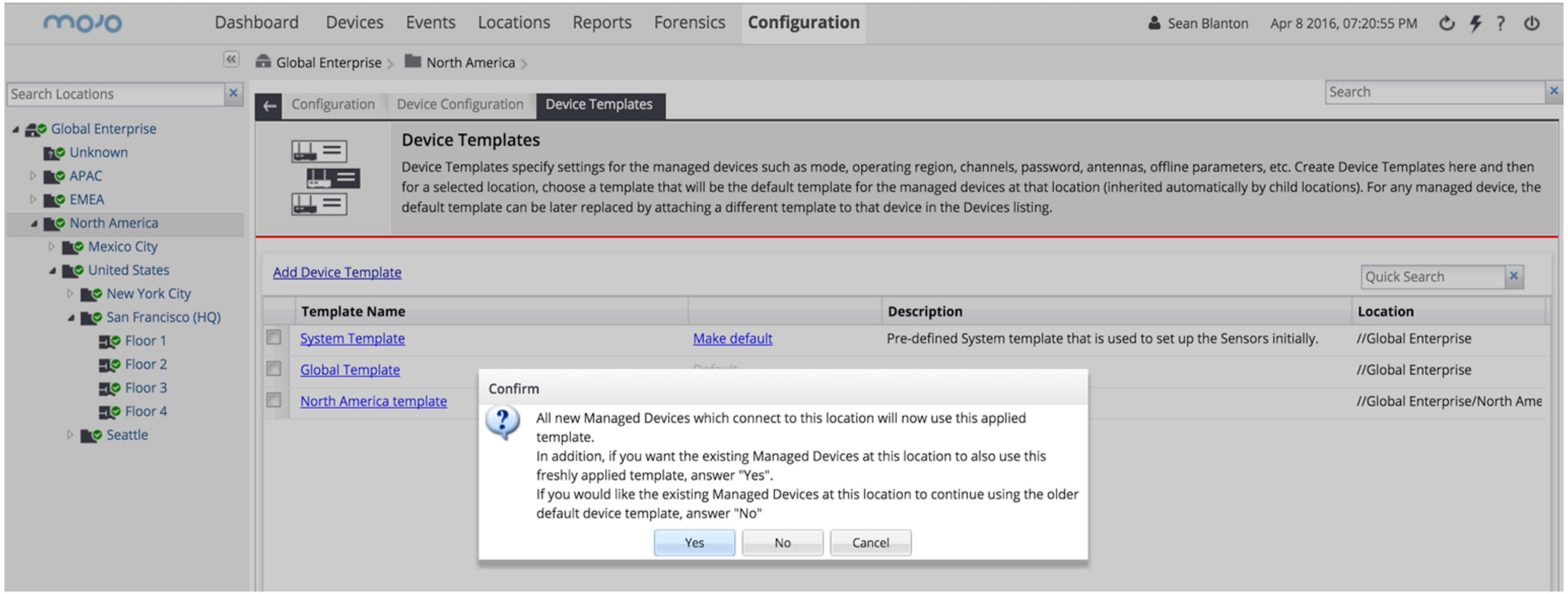
Profile Name
SSID

Broadcast SSID ✓ Guest SSID ☐ Application Visibility ☐ Association Analytics ☐

- Security Open
- Network VLAN - 0 ; Bridged
- Role Based Control Role Based Control - Disabled
- Captive Portal Captive Portal - Disabled
- Bonjour Gateway Disabled
- HTTP Content Analytics Disabled
- Firewall Firewall - Disabled
- SSID Scheduling Lifetime - Now to Forever ; Scheduling - Disabled
- Traffic Shaping & QoS Restrict Upload - Disabled ; Restrict Download - Disabled ; QoS - Enabled
- RF Optimizations Smart steering - Disabled ; Min Association RSSI - Disabled ; Band Steering - Enabled ; Smart Client Load Balancing - Disabled ;

Configuration: Device Template

- Device Template contains all the parameters that are SSID-agnostic
- Any no of device templates can be created but each location can have only active (default) device template.



Device Templates

Device Templates specify settings for the managed devices such as mode, operating region, channels, password, antennas, offline parameters, etc. Create Device Templates here and then for a selected location, choose a template that will be the default template for the managed devices at that location (inherited automatically by child locations). For any managed device, the default template can be later replaced by attaching a different template to that device in the Devices listing.

Template Name		Description	Location
System Template	Make default	Pre-defined System template that is used to set up the Sensors initially.	//Global Enterprise
Global Template	Default		//Global Enterprise
North America template			//Global Enterprise/North Ame

Confirm

All new Managed Devices which connect to this location will now use this applied template.
In addition, if you want the existing Managed Devices at this location to also use this freshly applied template, answer "Yes".
If you would like the existing Managed Devices at this location to continue using the older default device template, answer "No"

Yes No Cancel

Management of APs

- Command Line Interface (CLI)
  - SSH is required to fetch the information
  - CLI access is device (AP) specific.
  - Only limited set of commands.
  - User needs special privileges for executing advanced commands.
- Graphical User Interface (GUI)
  - Provides visual access to network configurations and monitoring dashboard.
  - User needs to navigate through the UI which sometimes makes it slow and time consuming.
  - User cannot interact with GUI programmatically.

Management of APs

- Application Programming Interface (API)
  - Enables the use of GET/SET methods to fetch information from WM and send instructions to change AP configuration/state.
  - API Response is in the JSON format, making it easier to fetch/modify the required information from WM.
  - APIs can be easily integrated with other application using any programming language that supports HTTP-based calls.
  - More than one API can be called a at time making the whole process very fast and reliable.

Arista WiFi APIs

- Web-based API that allow developers to programmatically interact with Wireless Manager(MW).
- Arista WIFI API call flow is divided into two parts:
  - Accessing Service (Authentication Required): Login into Wireless Manger
  - Fetching information of Wireless Manger Using HTTP methods: GET,POST,PUT DELETE (No additional authentication required).

What Action Does the API Perform?

Management

- User Management
- Location and Layouts
- Device Management
- Event Management
- Reports Management
- Local Policies

Troubleshooti

- Troubleshooting

Analytics

- Association and Visibility Analytics
- Application Visibility
- User Action Log

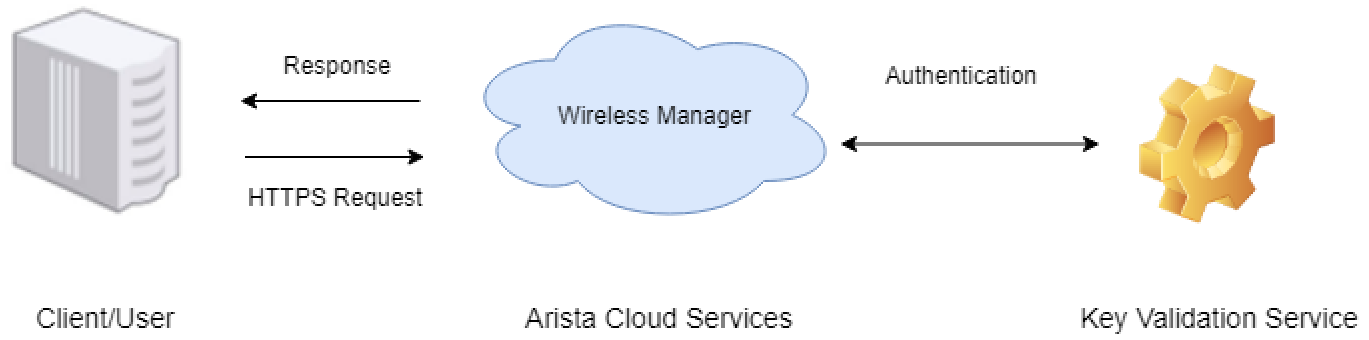
Arista API Syntax

API Syntax **:** <HTTP_request_method> <Base_URL>/<API_signature>

- HTTP request types: GET, PUT, POST, or DELETE.
- Base Url: https://<Mojo_Server_IP>/new/webservice/{version}
- Sample Call: $\underbrace{\textbf{POST}}_{\text{Request Method}}$ $\underbrace{\text{https://awm17001.srv.wifi.arista.com/new/webservice/v2}}_{\text{Base URL}}$

Login into Wireless Manger

- Wireless Manager provides support for a Key Validation Service (KVS).
- Launchpad generates key-value pairs and assigns appropriate service privileges on this key-value pair.
- Key-value pair is used to authenticate Wireless Manager through REST APIs.



API Call For Accessing Wireless Manager

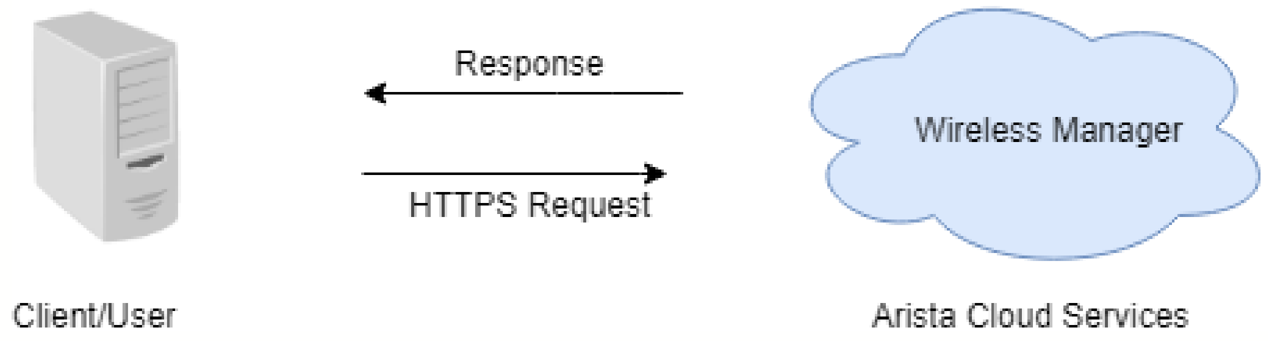
- API is used to log in to the Arista Server in the Arista Cloud.
- **URL**: https://awm17001.srv.wifi.arista.com/new/webservice/login/$\underbrace{\text{modScanWifi}}_{\text{Client Identifier}}$/$\underbrace{\text{86400}}_{\text{Timeout Period}}$
- **Method:** POST
- **Status Code** : 200
- **Request Body:**

```
{
            "type":"apikeycredentials",
            "keyId":"KEY-ATN***********",
            “keyValue":“*************************************“
      }
```

- **Response Body:**
  - (Check Response on REST Client)

Fetching Information Using API Call

- No additional authentication is required for API call.



Making an API Call (1/2)

- **API Request:**
  - **Path Parameters**
    - API Syntax: POST <Base_URL>/login/{clientidentifier}/{timeout}
    - Sample URL: https://awm17001.srv.wifi.arista.com/new/webservice/login/$\underbrace{\text{modScanWifi/}}_{\text{Client Identifier}}\underbrace{\text{3600}}_{\text{Timeout Period}}$
  - **URL Parameters**: are name-value pairs that are appended to the request URL
    - API Syntax:
      GET<Base_URL>/ual/{starttime}/{endtime}/{filtertype}/{encoding}?sortfilter=<value>&location=<value>
    - Sample URL: GET
      https://awm17001.srv.wifi.arista.com/new/webservice/ual/1000000/2000000/1/UTF-8?sortfilter=DATE_TIME&location=$\underbrace{\text{\{"type":"locallocationid","id":1\}}}_{\text{Name Value Pair}}$
  - **Body Parameters:** are provided in the application/JSON format

```
{
        "type":"apikeycredentials",
        "keyId":"KEY-ATN***********",
        “keyValue":“**********************************“
}
```

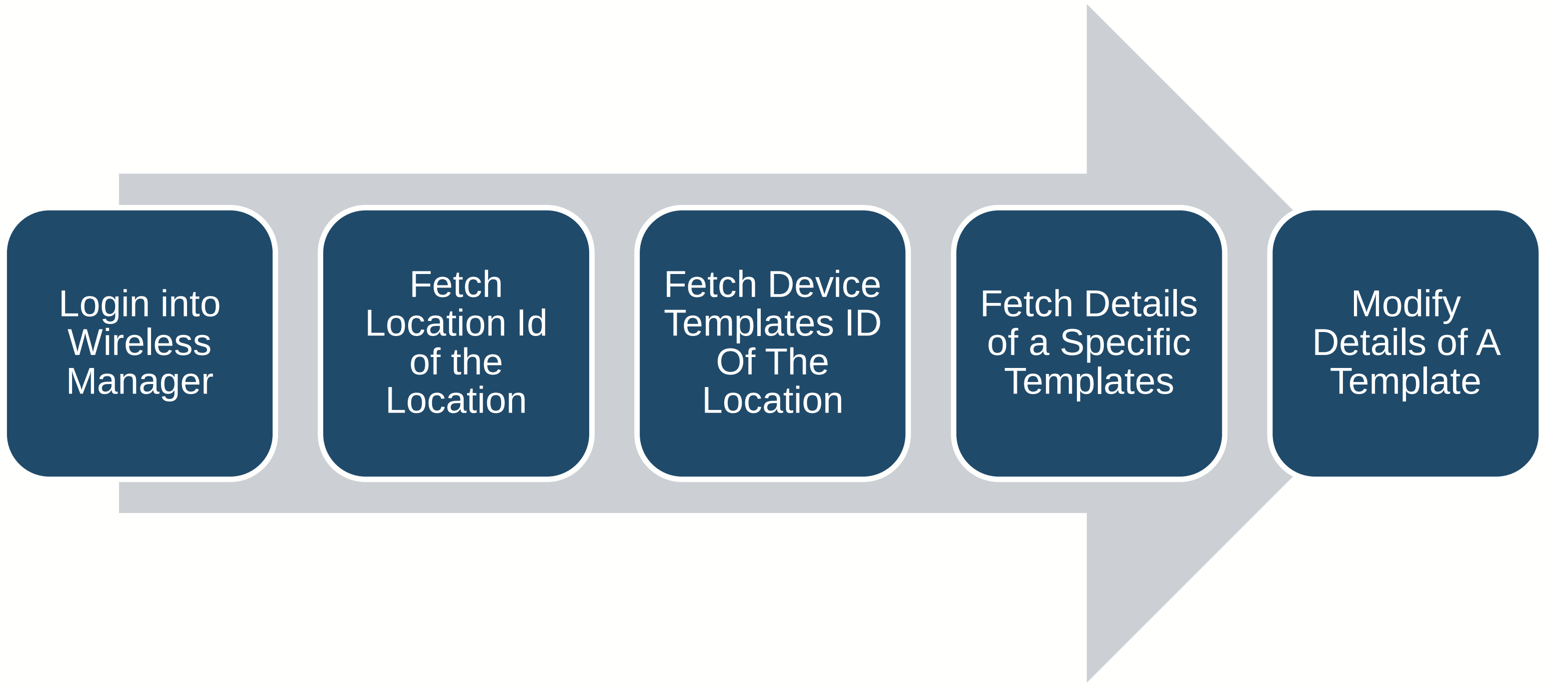
Making an API Call (2/2)

- **API Response**:
  - The success/failure of the API response is determined by the status code.
  - API response body may or may not contain any information:
    - » Sample Response: provided in the application/JSON format

```
{
      "version":"6.8",
        "build":"6.8.10",
 }
```

Examples 1

- Modifying Transmission Power in Device Template


Login into Wireless Manager
Fetch Location Id of the Location
Fetch Device Templates ID Of The Location
Fetch Details of a Specific Templates
Modify Details of A Template

Example 1/5

- Login into Wireless Manager
  - API is used to log in to the Arista Server in the Arista Cloud.
  - **URL:** https://awm17001.srv.wifi.arista.com/new/webservice/login/modScanWifi/86400
  - **Method:** POST
  - **Status Code** : 200
  - **Request Body:**

```
{
        "type":"apikeycredentials",
        "keyId":"KEY-ATN***********",
        “keyValue":“***********************************“
 }
```

  - **Response Body:**

    (Check Response on REST Client)

Examples (2/5)

- Get Location Id
  - API call is used to fetch location Id of all the locations in location tree.
  - **URL**: https://awm17001.srv.wifi.arista.com/new/webservice/v2/locations/tree
  - **Method:** Get
  - **Status Code**: 200
  - **Request Body**- NIL
  - **Response Body:**
    - (Check Response on REST Client)

Note:

- Please note down the LocationID for location specific response.
- Location Id is a system generated Unique Identifier of each location in the location hierarchy.
- Location in the Location Hierarchy is of three type: Root Location, Parent Location, Child Location
- Every Location has unique LocationId

Examples (3/5)

- Get Template ID of Device Templates at Specific Location
  - API is used to fetch list of Templates configured on specific location.
  - **URL**: https://https://awm17001.srv.wifi.arista.com//new/webservice/v3/templates/26/DEVICE_TEMPLATE?locationid=30

    Template Id
  - Method: GET
  - Request Body: N/A
  - Response Code: 200
  - Response Body:
    - (Check Response on REST Client)

Note:

- Please note down the TemplateID for template specific response
- TemplateID is a system generated Unique Identifier for the templates
- Templates are used to push the configuration on APs

ARISTA

Examples (4/5)

- **Get Details of Device Templates at Specific Location**
  - API is used to fetch list of Templates configured on specific location.
  - **URL**: https://awm17001.srv.wifi.arista.com//new/webservice/v3/templates/21/DEVICE_TEMPLATE?locationid=24
  - Method: GET
  - Request Body: N/A
  - Response Code: 200
  - Response Body:
    - (Check Response on REST Client)

ARISTA

Examples (5/5)

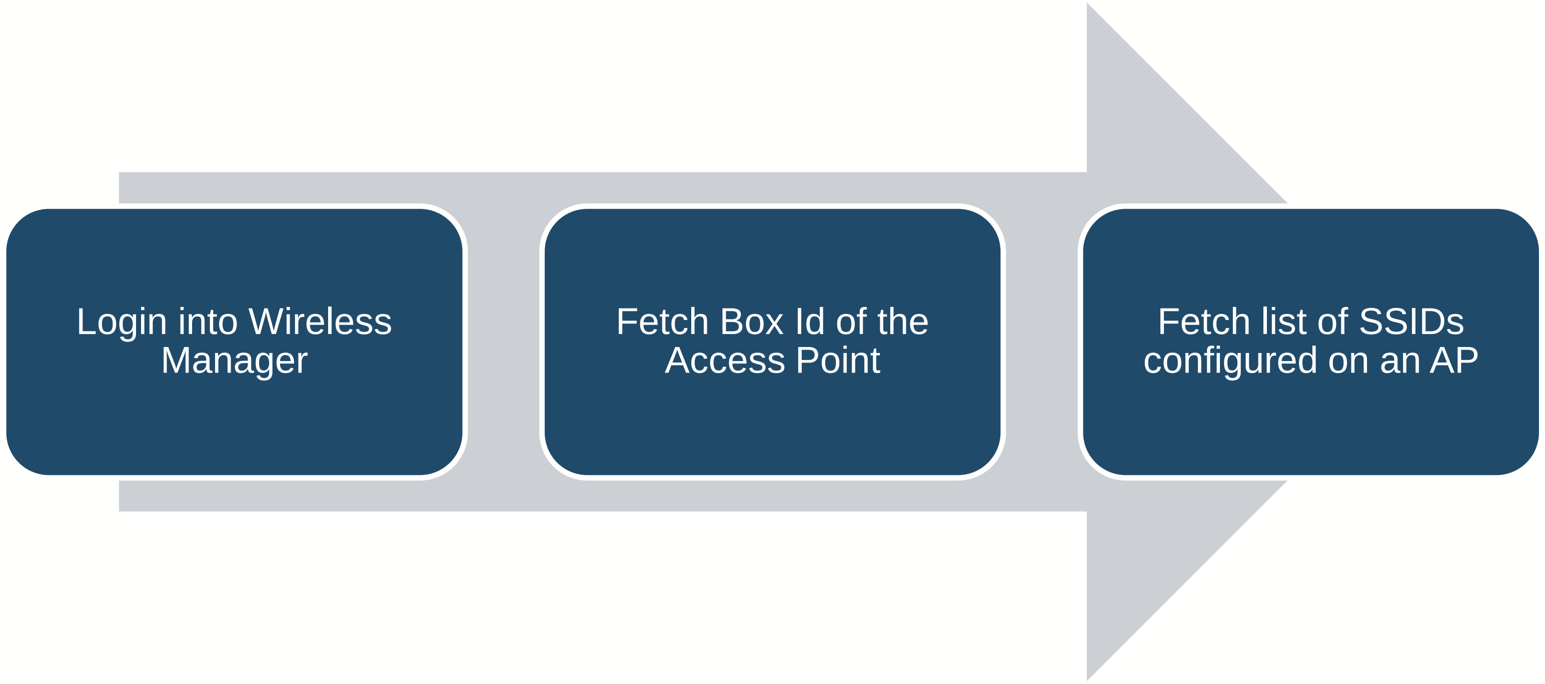
Modifying Transmission Power of 2.4GHz Band

- Get Signal Strength of AP or client radio that is acting as a transmitter
- **URL**: https://awm17001.srv.wifi.arista.com/new/webservice/v3/templates
- **Method** : POST
- **Request Body** :
- **Response Code** : 200
- **Response Body**:
  - (Please check Response in REST Client)
  - Changed parameters can be verified by checking the required parameter in the JSON response

ARISTA

Examples 2

- Fetching SSIDs Radiating on a Specific AP


Login into Wireless Manager
Fetch Box Id of the Access Point
Fetch list of SSIDs configured on an AP

Example 2: 1/3

- Login into Wireless Manager
  - API is used to log in to the Arista Server in the Arista Cloud.
  - **URL:** https://awm17001.srv.wifi.arista.com/new/webservice/login/modScanWifi/86400
  - **Method:** POST
  - **Status Code** : 200
  - **Request Body:**

```
{
        "type":"apikeycredentials",
        "keyId":"KEY-ATN***********",
        “keyValue":“************************************“
 }
```

  - **Response Body:**

    (Check Response on REST Client)

Example 2: 2/3

- Get Managed Devices
  - This API is used to retrieve a paged list against the MAC address of the device.
  - **URL**: https://awm17001.srv.wifi.arista.com/new/webservice/v3/devices/manageddevices?sortby="name"&filter={"property":"macaddress","value": ["00:11:74:80:3D:7F"],"operator": "contains"}
  - **Method:** GET
  - **Status Code** : 200
  - **Request Body:** N/A
  - **Response Body:**

    (Check Response on REST Client)

Note:

- Please note down the boxId for the device for subsequent API calls.
- BoxId is a unique system generated identifier of the AP

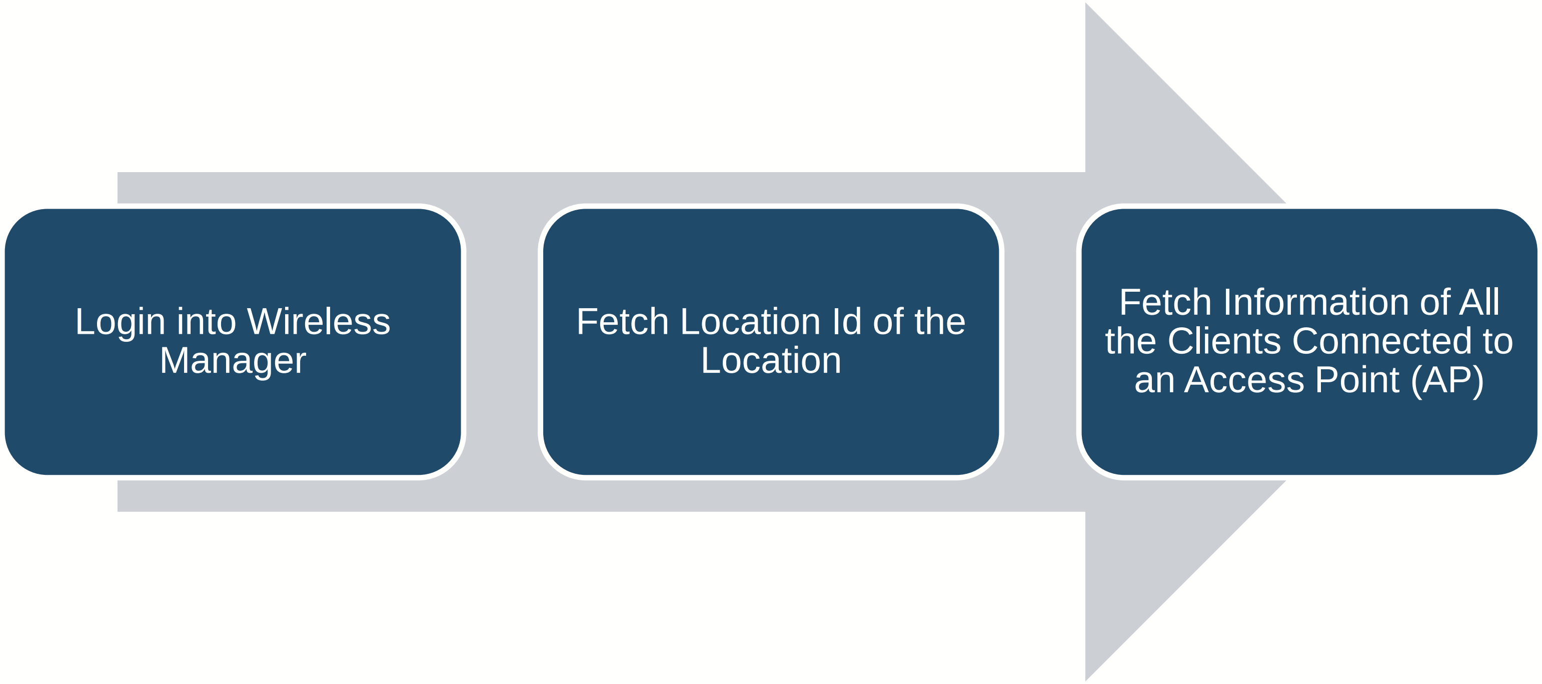
Example 2: 3/3

- Get Managed Devices
  - This API is used to retrieve a list of SSIDs configured on the Access Point .
  - **URL**: https://awm17001.srv.wifi.arista.com/new/webservice/V4/devices/ssids?manageddeviceboxid=18
  - **Method:** GET
  - **Status Code** : 200
  - **Request Body:** N/A
  - **Response Body:**

```
[
  "test_2.4",
  "iisc_test_5",
  "IISC_test_2.4“
]
```

Examples 3

- Fetching List of Clients Connected on a Specific Access Point


Login into Wireless Manager
Fetch Location Id of the Location
Fetch Information of All the Clients Connected to an Access Point (AP)

Example 3: 1/3

- Login into Wireless Manager
  - API is used to log in to the Arista Server in the Arista Cloud.
  - **URL:** https://awm17001.srv.wifi.arista.com/new/webservice/login/modScanWifi/86400
  - **Method:** POST
  - **Status Code** : 200
  - **Request Body:**

```
{
        "type":"apikeycredentials",
        "keyId":"KEY-ATN***********",
        “keyValue":“***********************************“
 }
```

  - **Response Body:**

    (Check Response on REST Client)

Example 3: (2/3)

- Get Location Id
  - API call is used to fetch location specific details of all the locations in location hierarchy.
  - **URL**: https://awm17001.srv.wifi.arista.com/new/webservice/v2/locations/tree
  - **Method:** Get
  - **Status Code**: 200
  - **Request Body**- NIL
  - **Response Body:**
    - (Check Response on REST Client)

Note:

- Please note down the LocationID for location specific response
- Location Id is a system generated Unique Identifier of each location in the location hierarchy.
- Location in the Location Hierarchy is of three type: Root Location, Parent Location, Child Location
- Every Location has a unique LocationId

ARISTA

Examples 3: (3/3)

- Fetch Clients Details
  - API call is used to fetch the details of all clients connected the specific location.
  - **URL**: https://awm17001.srv.wifi.arista.com/new/webservice/v2/devices/clients/0/25?capability=2&locationid=30&sortcolumn=devicename&sortascending=false
  - **Method:** Get
  - **Status Code**: 200
  - **Request Body**- NIL
  - **Response Body:**
    - (Check Response on REST Client)

Thank You

www.arista.com

ARISTA