

Meraki API

Agenda

- Intro
 - Why programming
 - REST API
- Meraki Cloud platform
 - Dashboard APIs
 - Webhooks
 - Scanning APIs
 - EXCAP APIs
 - Camera & Analytics w/ MV Sense
- Demo Time

Why Network Programmability



Automation

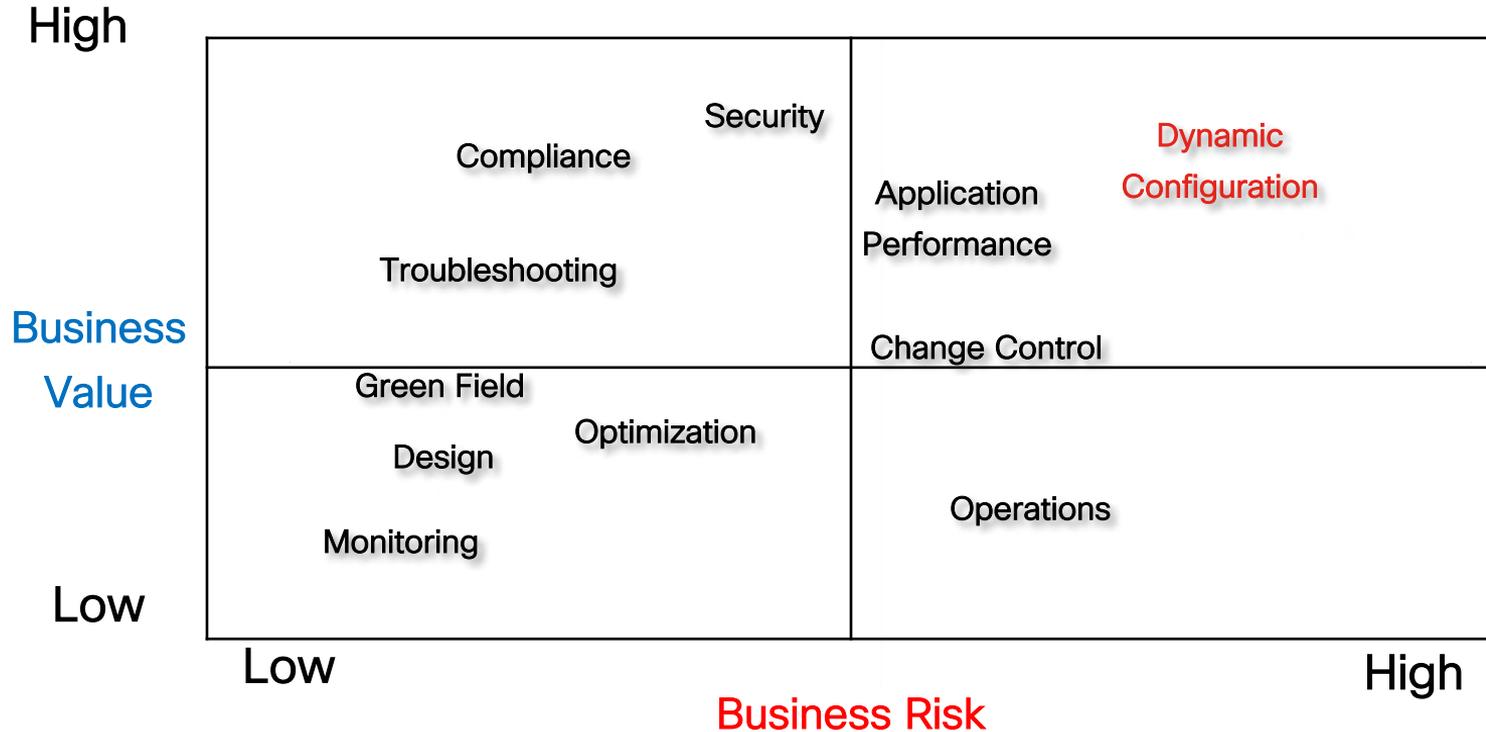


Integration



Innovation

Programmability Use Cases



Programmability Benefits

- Innovation and business agility
- Accelerated time to market
- Service delivery optimization
- Highly skilled architects and engineers focus on business
- Cost reduction and increased efficiencies
- Improved network availability due to reduced human error

REST Web service

- What is REST?
 - REpresentational State Transfer (REST)
 - API framework built on HTTP
- What is a REST Web Service?
 - REST is architecture style for designing networked applications.
 - Popular due to performance, scale, simplicity, and reliability

GET

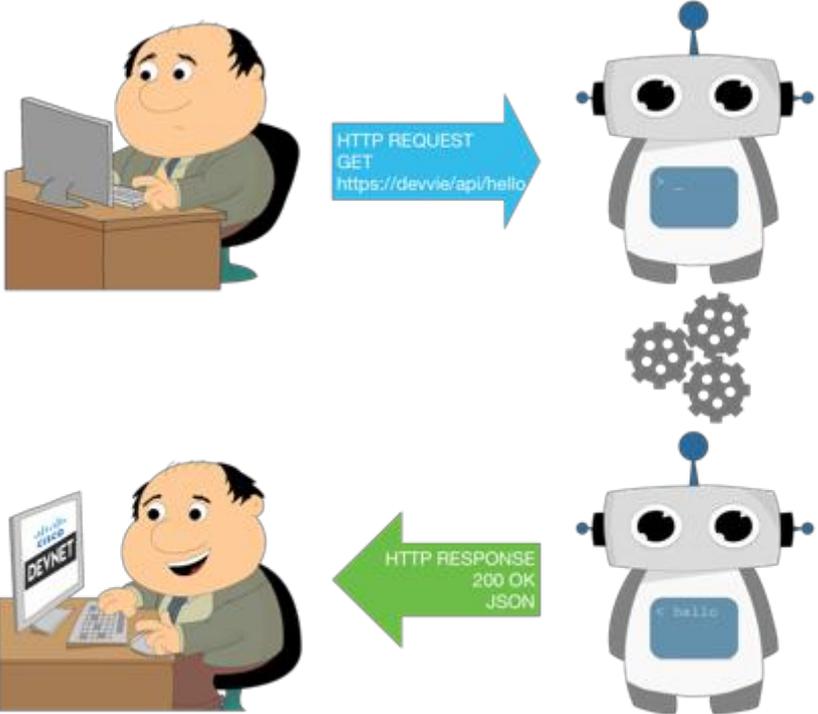
POST

PUT

DELETE

{REST}

Requests and Response REST API Flow



Overview of API

APIs help developers create apps that benefit the end user.



URI Structure

`http://maps.googleapis.com/maps/api/geocode/json?address=sanjose`

Server or Host Resource Parameters

Domain name

`http://` or `https://`

- Protocol over which data is sent between client and server

Server or Host

- Resolves to the IP and port to which to connect

Resource

- The location of the data or object of interest

Parameters

- Details to scope, filter, or clarify a request. Often optional.



Data: Sending and Receiving

- Contained in the message body
- POST, PUT, PATCH requests typically include a message body
- GET responses will include a message body
- Format typically JSON or XML
 - Specified in “Content-Type” header



HTTP Methods: What to do?

HTTP Verb	Typical Purpose (CRUD)	Description
POST	Create	Used to create a new object, or resource. Example: Create new feature template
GET	Read	Retrieve resource details from the system. Example: Get list of devices from the inventory
PUT	Update	Typically used to replace or update a resource. Can be used to modify or create. Example: Modify the existing prefix list
DELETE	Delete	Remove a resource from the system. Example: Delete a feature template

Reference: <https://restfulapi.net/http-methods/>

Response Status Codes: Did it work?

Status Code	Status Message	Meaning
200	OK	All looks good
204	No Content	Request succeeded, but no message body returned
400	Bad Request	Request was invalid
401	Unauthorized	Authentication missing or incorrect
403	Forbidden	Request was understood, but not allowed
404	Not Found	Resource not found
500	Internal Server Error	Something wrong with the server
503	Service Unavailable	Server is unable to complete request

Reference: <https://restfulapi.net/http-status-codes/>

Headers:

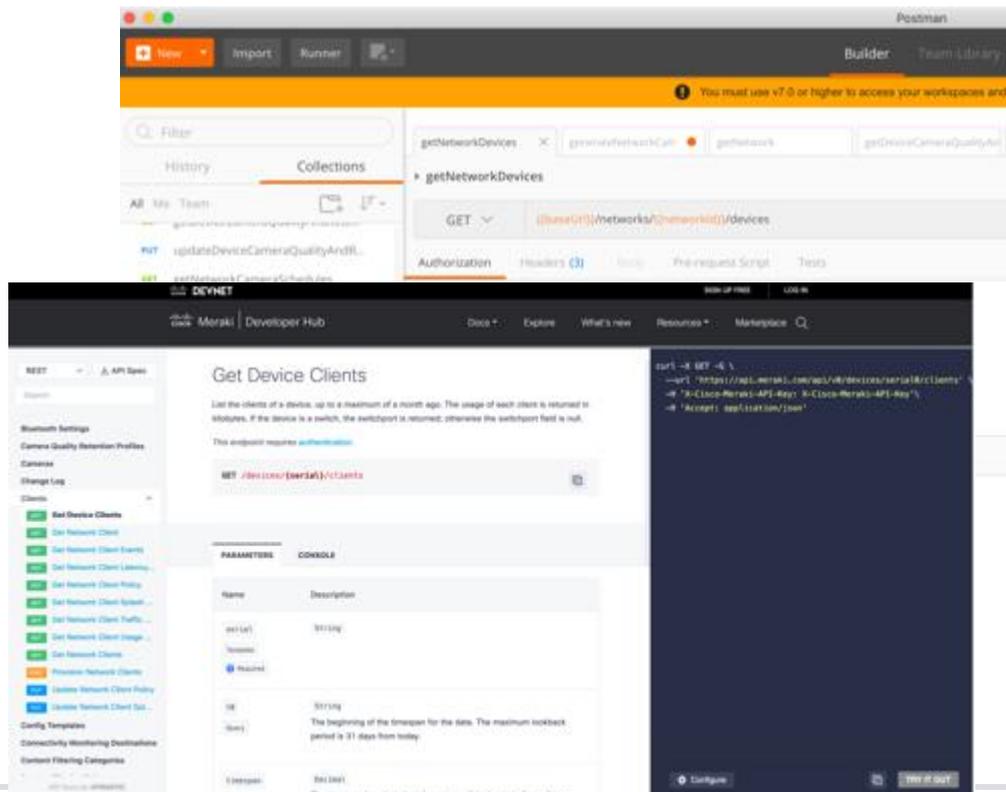
What additional details and metadata can I use?

Header	Example Value	Purpose
Content-Type	application/json	Specify the format of the data in the body
Accept	application/json	Specify the requested format for returned data
X-XSRF-TOKEN	1D76CEFD47B695EEA1CA92C4836AE772EFA	Provide token to authorize a request

- Used to pass information between client and server
- Included in both Request and Response

Options for Working with REST APIs

- curl
- Linux command line application
- Postman
- API testing application and framework
- Requests
- Python library for scripting
- OpenAPI/Swagger
- Dynamic API Documentation



Meraki Cloud

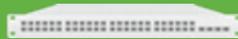
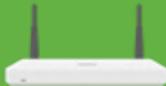
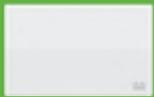
Meraki Cloud Platform

OUT-OF-THE-BOX MANAGEMENT & ANALYTICS



{ APIs }

INTEGRATIONS & SOLUTIONS POWERED BY MERAKI

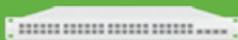
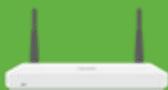
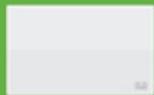
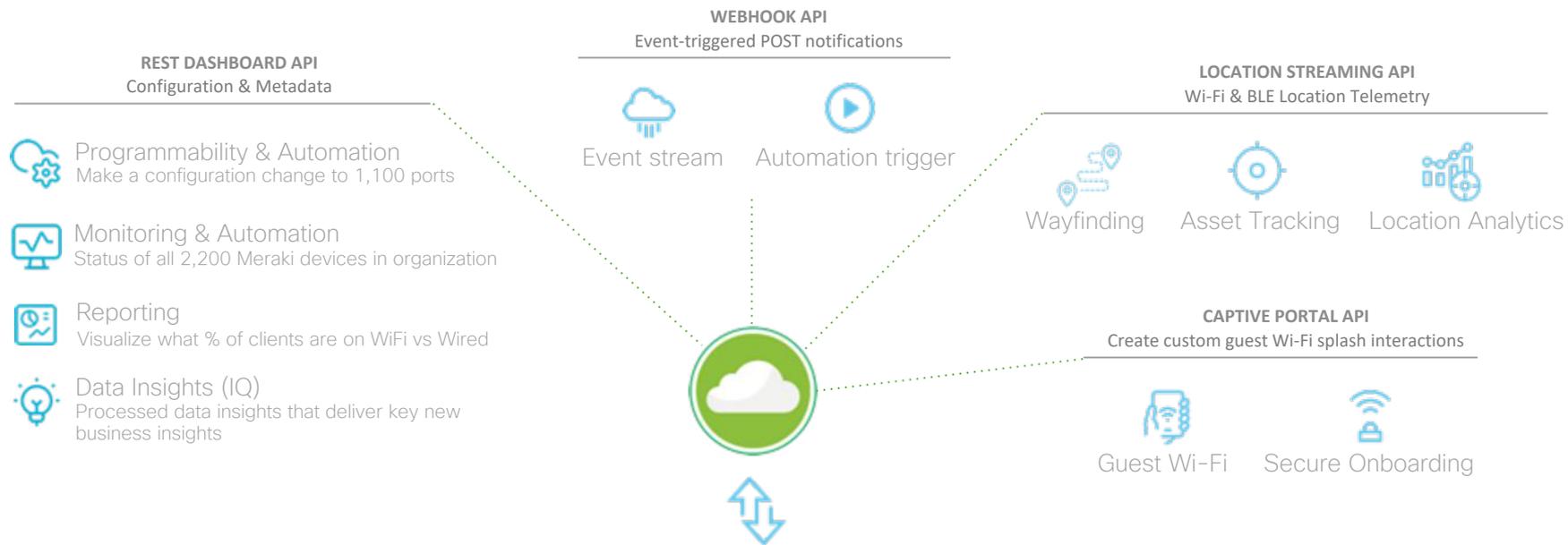


Automation of repetitive tasks
“1 click vs 1,328 clicks”

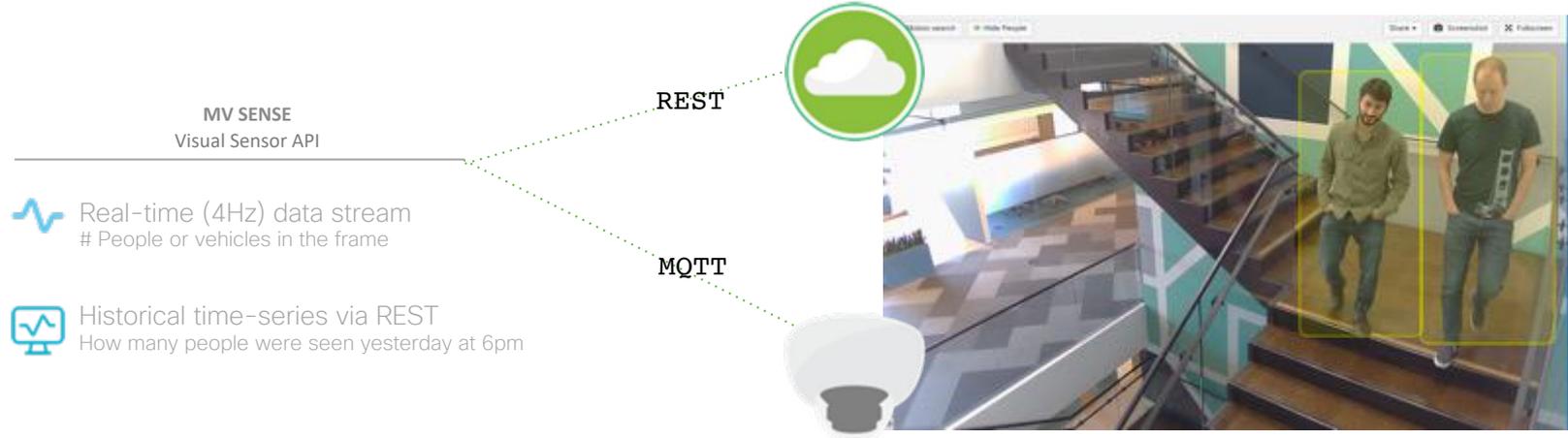
Platform Integrations
“Meraki plus _____”

Tailored solutions
“Seamless device onboarding”

Driven by Powerful, Developer-friendly APIs



Driven by Powerful, Developer-friendly APIs



Dashboard API (REST)



Dashboard API

Programmatically manage and report on your network

DASHBOARD UI				REST API
20	●	<u>2.55</u>	MR52	<pre>"name": "2.55", "serial": "Q2TD-LS8S-J6YJ", "mac": "0c:8d:db:7a:01:64", "publicIp": "67.188.23.251", "networkId": "L_599541700393699125", "status": "alerting", "lanIp": "192.168.1.174"</pre>
21	●	<u>2.25</u>	MR52	
22	●	<u>2.54</u>	MR52	
23	●	<u>2.53</u>	MR52	
24	●	<u>5.10</u>	MR42	
25	●	<u>5.36 - Axe</u>	MR42	
26	●	<u>5.12</u>	MR53	
				<pre>"name": "2.25", "serial": "Q2DK-2H9N-JSTE", "mac": "00:18:0a:5b:0c:b0", "publicIp": "67.188.23.251", "networkId": "L_599541700393699125", "status": "online", "lanIp": "172.16.0.5"</pre>

Authenticating with the API

- A Bearer Token is required to use the Meraki Dashboard API
 - 1) Enable API access
 - 2) Generate a new bearer token via user profile
 - 3) Copy ticket and add to **X-Auth-Token** Header
 - 4) Use in subsequent API calls
- Dashboard APIs use JSON format for exchange of data between the Meraki Cloud and the REST application (API client)

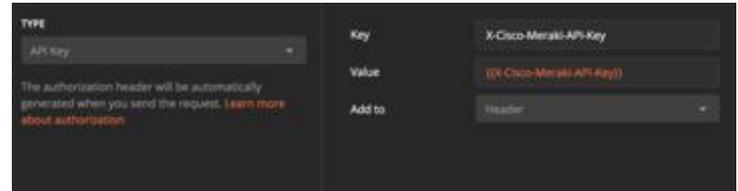
Step 1: Request service token (Organization > Dashboard API)



Step 2: Generate a new API token via User Profile

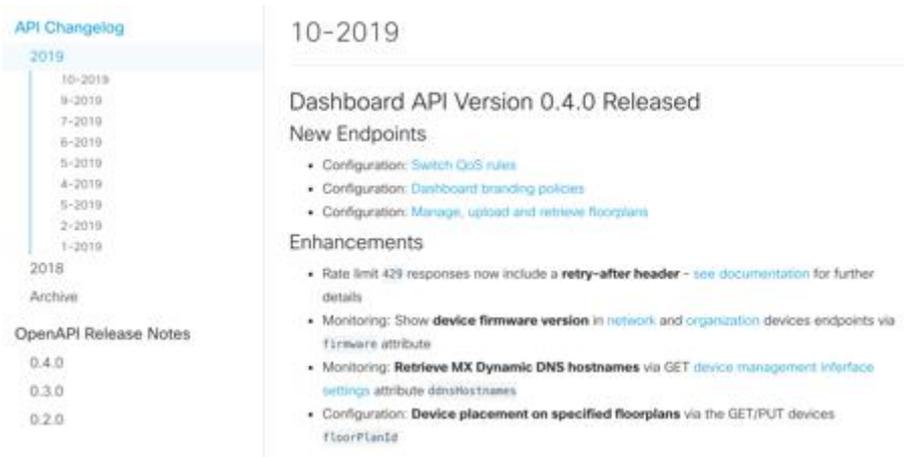


Step 3: Add ticket to the X-Auth-Token header



API Lifecycle for Dashboard API

API follows semantic versioning (major.minor.patch)

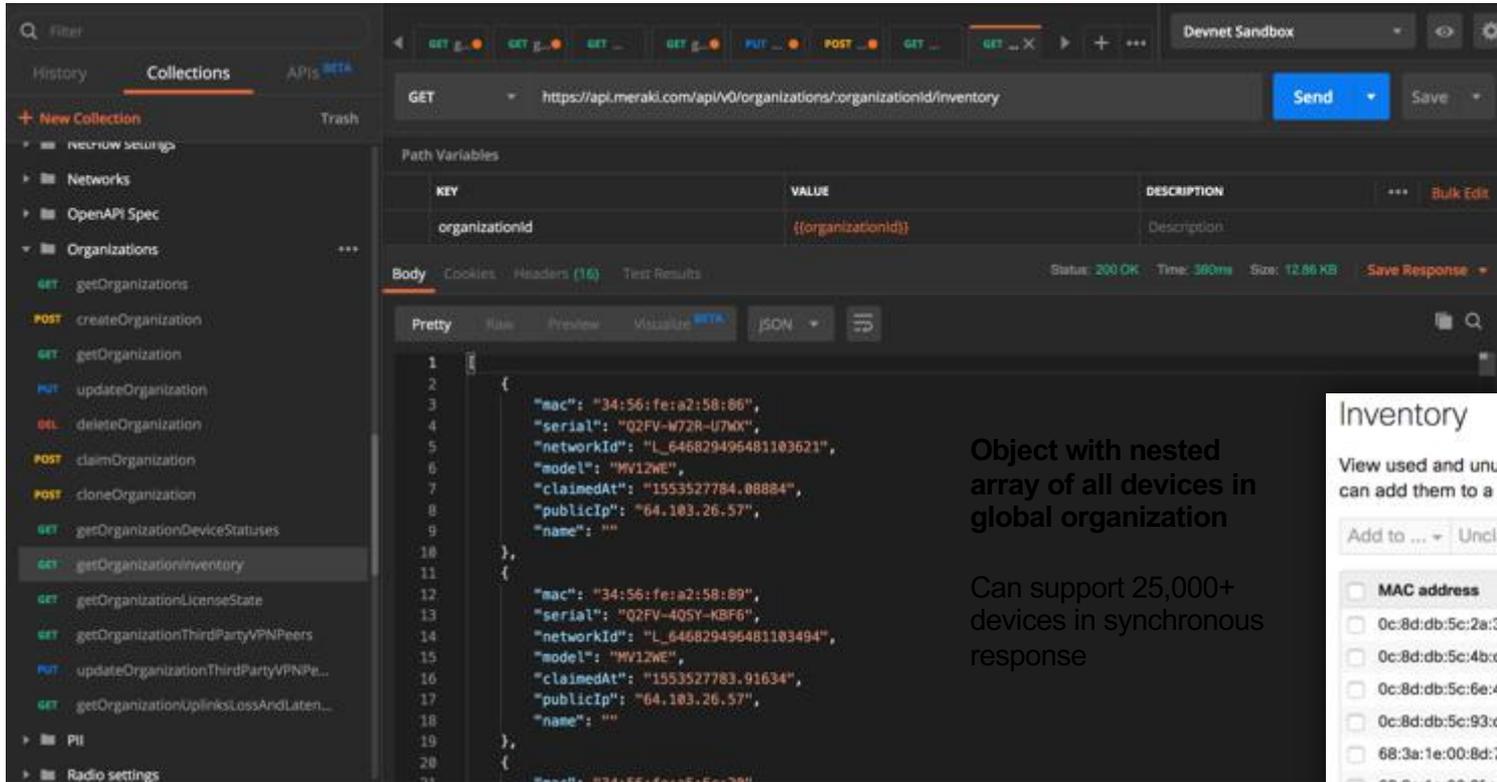


The screenshot shows the 'API Changelog' page for the Dashboard API. The main content is for the month of 10-2019, titled 'Dashboard API Version 0.4.0 Released'. It lists 'New Endpoints' and 'Enhancements'. The 'New Endpoints' section includes three items: 'Configuration: Switch QoS rules', 'Configuration: Dashboard branding policies', and 'Configuration: Manage, upload and retrieve floorplans'. The 'Enhancements' section includes four items: 'Rate limit 429 responses now include a retry-after header', 'Monitoring: Show device firmware version in network and organization devices endpoints via firmware attribute', 'Monitoring: Retrieve MX Dynamic DNS hostnames via GET device management interface settings attribute dnshostnames', and 'Configuration: Device placement on specified floorplans via the GET/PUT devices floorPlans'.

View @ meraki.io/whats-new



Example 1: Org-wide Device Inventory (CMDB)



The screenshot shows a REST client interface with the following details:

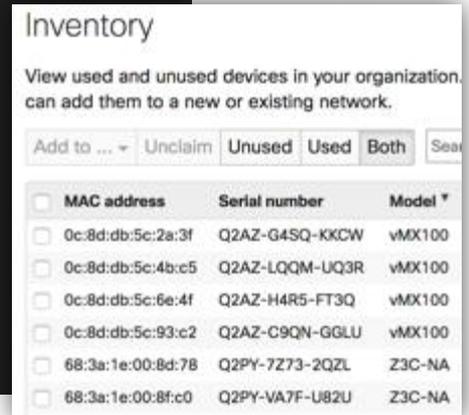
- Method: GET
- URL: `https://api.meraki.com/api/v0/organizations/organizationId/inventory`
- Path Variables:

KEY	VALUE	DESCRIPTION
organizationId	{{organizationId}}	Description
- Status: 200 OK, Time: 360ms, Size: 12.86 KB
- Response Body (Pretty):

```
1 {
2   "mac": "34:56:fe:a2:58:86",
3   "serial": "Q2FV-W72R-U7WX",
4   "networkId": "L_646829496481103621",
5   "model": "MV12WE",
6   "claimedAt": "1553527784.08884",
7   "publicIp": "64.103.26.57",
8   "name": ""
9 },
10
11 {
12   "mac": "34:56:fe:a2:58:89",
13   "serial": "Q2FV-4Q5Y-KBF6",
14   "networkId": "L_646829496481103494",
15   "model": "MV12WE",
16   "claimedAt": "1553527783.91634",
17   "publicIp": "64.103.26.57",
18   "name": ""
19 },
20
21 {
```

Object with nested array of all devices in global organization

Can support 25,000+ devices in synchronous response



Inventory
View used and unused devices in your organization. You can add them to a new or existing network.

Buttons: Add to ... Unclaim Unused Used Both Search

<input type="checkbox"/>	MAC address	Serial number	Model
<input type="checkbox"/>	0c:8d:db:5c:2a:3f	Q2AZ-G4SQ-KKCW	vMX100
<input type="checkbox"/>	0c:8d:db:5c:4b:c5	Q2AZ-LQQM-UQ3R	vMX100
<input type="checkbox"/>	0c:8d:db:5c:6e:4f	Q2AZ-H4R5-FT3Q	vMX100
<input type="checkbox"/>	0c:8d:db:5c:93:c2	Q2AZ-C9QN-GGLU	vMX100
<input type="checkbox"/>	68:3a:1e:00:8d:78	Q2PY-7Z73-2QZL	Z3C-NA
<input type="checkbox"/>	68:3a:1e:00:8f:c0	Q2PY-VA7F-U82U	Z3C-NA

Example 2: Org-wide Status Monitoring

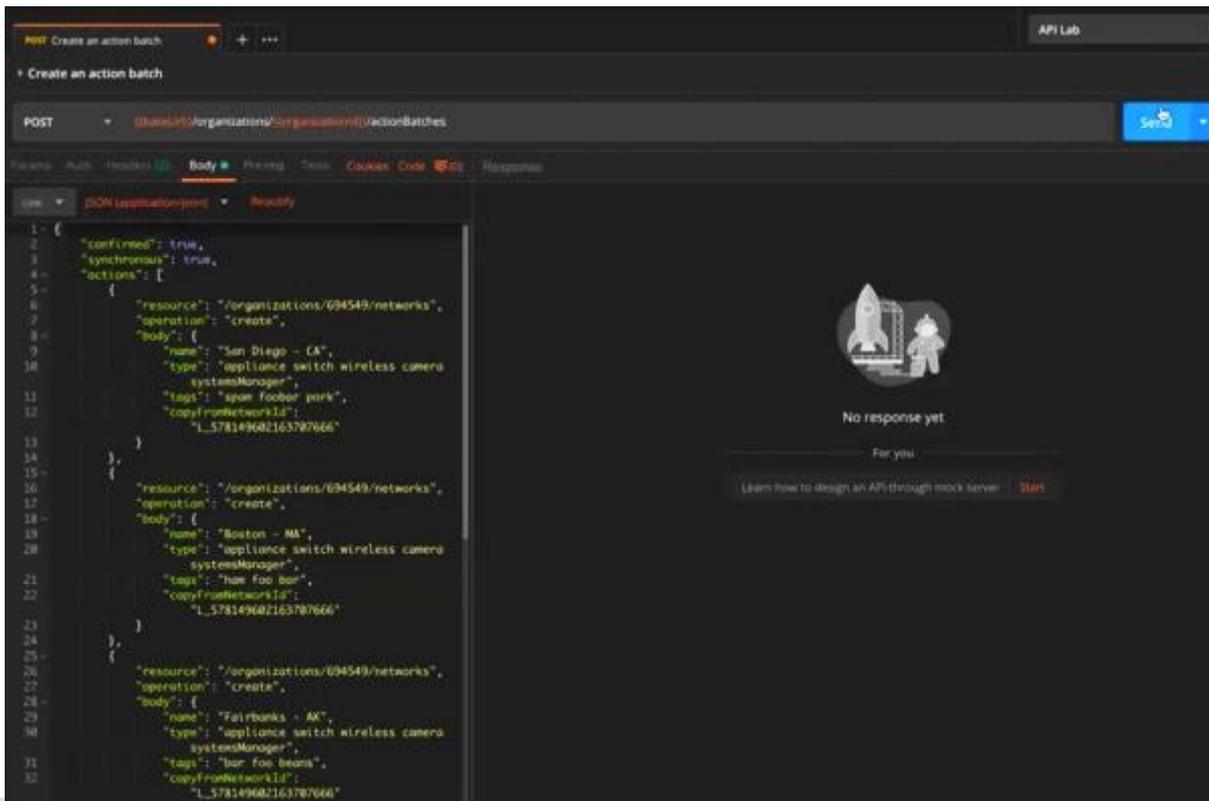
GET `https://api.meraki.com/api/v0/organizations/{organizationId}/deviceStatuses`

Body: `[[{"name": null, "serial": "Q2QN-9J8L-SLP0", "mac": "e0:55:3d:17:d4:23", "publicIp": "64.103.26.57", "networkId": "L_646829496481099586", "status": "online", "lastReportedAt": "2019-10-06 23:29:52.207", "usingCellularFailover": false, "wan1Ip": "10.10.10.106", "wan2Ip": null}, {"name": null, "serial": "Q2HP-F5K5-R88R", "mac": "88:15:44:df:f3:af", "publicIp": "64.103.26.57", "networkId": "L_646829496481099586", "status": "offline", "lastReportedAt": "2019-03-22 21:27:01.323", "lanIp": "192.168.128.2"}]]`

Nested array containing all devices in global organization, with connectivity and alerting status

Model	Uplink IP (Port 1)	Public IP	Connectivity
MX100	185.53.227.85	185.53.227.85	Online
MX100	10.3.0.16	184.23.135.132	Offline
MR52	10.11.0.179	184.23.135.132	Online
MR52	10.255.29.103	193.117.158.138	Online

Example 3: ActionBatch API (20 x Network Create)



```
POST /organizations/694549/actionBatches

{
  "confirmed": true,
  "synchronous": true,
  "actions": [
    {
      "resource": "/organizations/694549/networks",
      "operation": "create",
      "body": {
        "name": "San Diego - CA",
        "type": "appliance switch wireless camera systemsManager",
        "tags": "span foobar park",
        "copyFromNetworkID": "L_378149682163787666"
      }
    },
    {
      "resource": "/organizations/694549/networks",
      "operation": "create",
      "body": {
        "name": "Boston - MA",
        "type": "appliance switch wireless camera systemsManager",
        "tags": "ham foo bar",
        "copyFromNetworkID": "L_378149682163787666"
      }
    },
    {
      "resource": "/organizations/694549/networks",
      "operation": "create",
      "body": {
        "name": "Fairbanks - AK",
        "type": "appliance switch wireless camera systemsManager",
        "tags": "bar foo beams",
        "copyFromNetworkID": "L_378149682163787666"
      }
    }
  ]
}
```

BATCH POST NETWORK CREATE

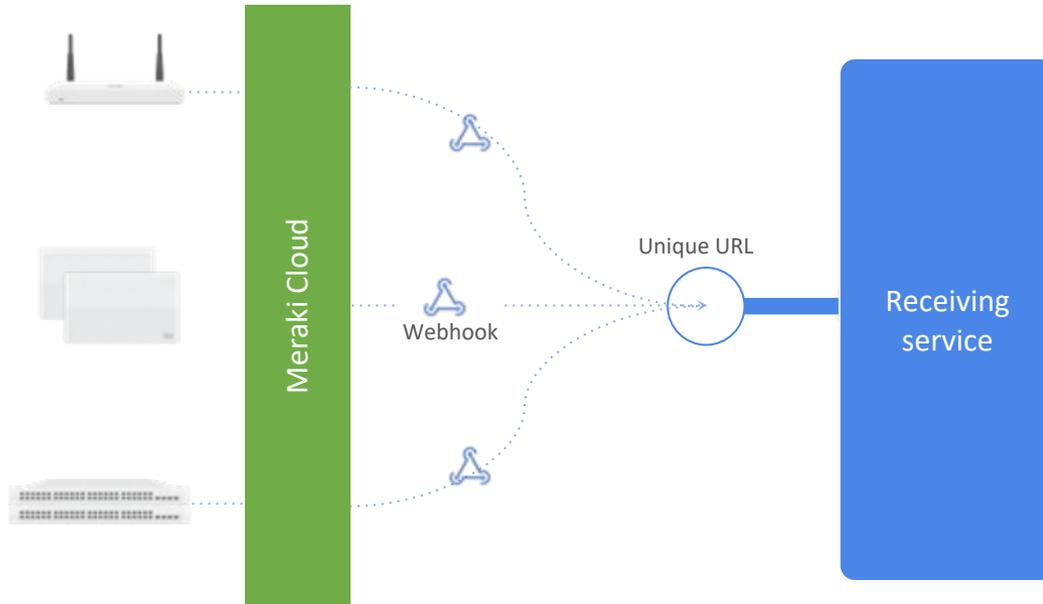
Can support 20 synchronous or 100 asynchronous constructive & destructive actions in a single POST

Meraki Webhooks

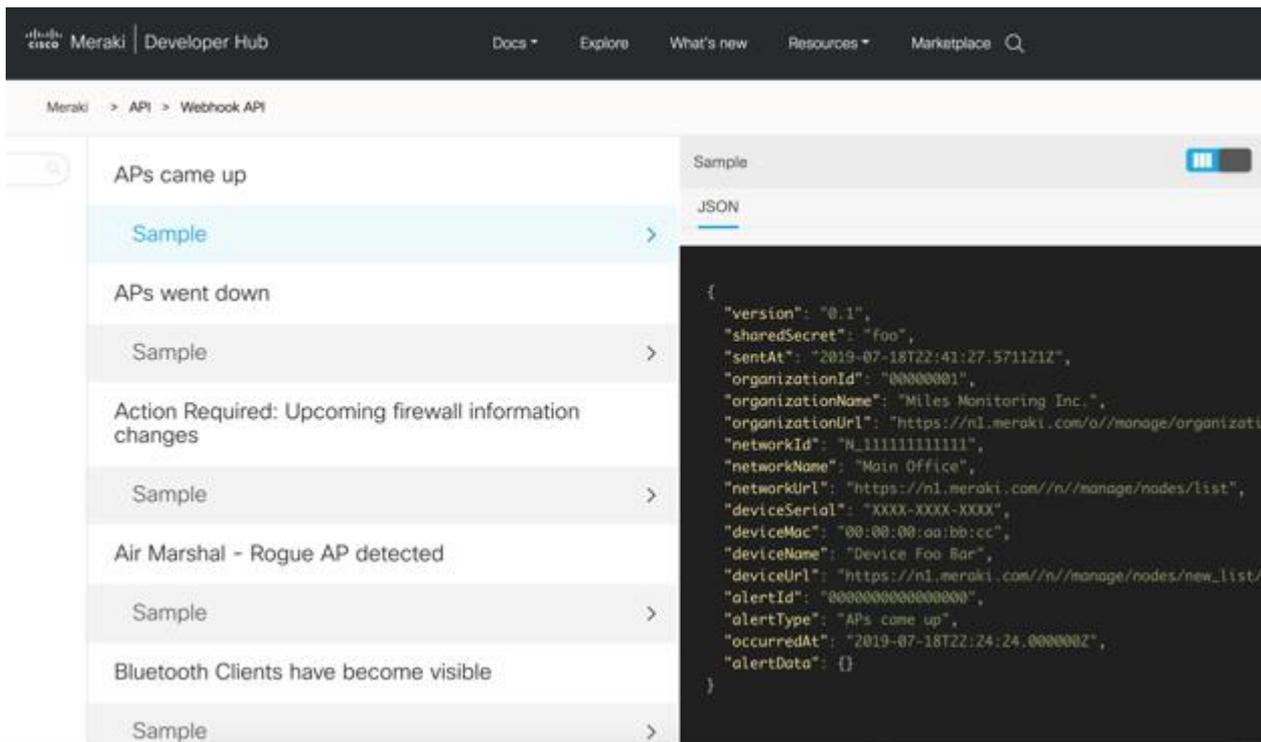


Webhook API

Push notifications for events from the Meraki Cloud



Cloud Alerts via Webhook to any Web Client



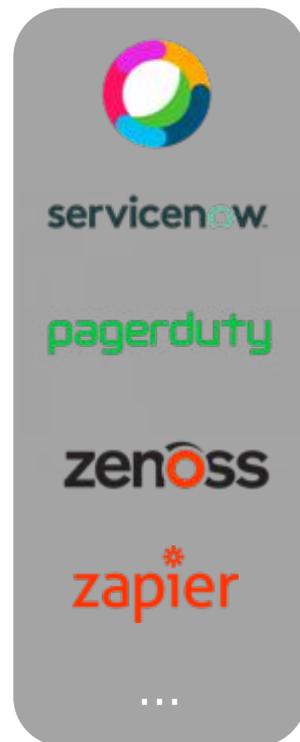
The screenshot shows the Meraki Developer Hub interface for the Webhook API. The left sidebar lists various alert types, and the main content area displays a sample JSON payload for an alert.

Alerts listed in the sidebar:

- APs came up
- APs went down
- Action Required: Upcoming firewall information changes
- Air Marshal - Rogue AP detected
- Bluetooth Clients have become visible

Sample JSON payload:

```
{
  "version": "0.1",
  "sharedSecret": "foo",
  "sentAt": "2019-07-18T22:41:27.571121Z",
  "organizationId": "00000001",
  "organizationName": "Miles Monitoring Inc.",
  "organizationUrl": "https://n1.meraki.com/o//manage/organizati",
  "networkId": "N_11111111111111",
  "networkName": "Main Office",
  "networkUrl": "https://n1.meraki.com//n//manage/nodes/list",
  "deviceSerial": "XXXX-XXXX-XXXX",
  "deviceMac": "00:00:00:aa:bb:cc",
  "deviceName": "Device Foo Bar",
  "deviceUrl": "https://n1.meraki.com//n//manage/nodes/new_list",
  "alertId": "0000000000000000",
  "alertType": "APs came up",
  "occurredAt": "2019-07-18T22:24:24.000000Z",
  "alertData": {}
}
```



40+ Event Triggers and Categories

Network Alerts

- Configuration settings are changed
- A VPN connection comes up or goes down ⓘ
- A rogue AP is detected
- Network usage exceeds GB in

Wireless Alerts

- A gateway goes offline for minutes
- A repeater goes offline for minutes
- A gateway becomes a repeater ⓘ

Security & SD-WAN Alerts

- A security appliance goes offline for minutes
- The primary uplink status changes ⓘ
- The DHCP lease pool is exhausted ⓘ
- An IP conflict is detected ⓘ
- Cellular connection state changes ⓘ
- A rogue DHCP server is detected ⓘ
- A warm spare failover occurs ⓘ
- Malware is blocked ⓘ
- Malware is downloaded ⓘ
- Clients connect or disconnect from the LAN ⓘ

Switch Alerts

- A switch goes offline for minutes
- A new DHCP server is detected on the network ⓘ
- goes down for more than minutes
- detects a cable error
- changes link speed
- A power supply goes down
- A redundant power supply is powering a switch ⓘ
- Unidirectional link detection (UDLD) errors exist on a port
- A switch is operating at critical temperature ⓘ



Setup & Authentication

1. Add one or more HTTP servers (receivers)
Include a name, unique HTTPS (TLS) URL and shared secret

Name	URL	Shared secret	
Zapier	https://hooks.zapier.com	*****	✕
Built.io	https://built.io/hooks/ca	*****	✕
Add an HTTP server			

2. Configured HTTP servers can now be selected as a recipient for any alert within dashboard, or via the alert configuration endpoint via API

Alerts Settings

Default recipients

webhook | jane.doe@meraki.com | Webhook: Zapier | webhook | Webhook: Built.io

Network-wide

Configuration settings are changed

Hide additional recipients

Webhook: Built.io | +

Setup & Authentication via REST

```
getNetworkHttpServers
List the HTTP servers for a network

GET https://api.meraki.com/api/v0/networks/:networkId/httpServers

Params Authorization Headers (1) Body Pre-request Script Tests

Body Headers

Pretty Raw Preview JSON

1 - [
2 -   {
3     "id": "ABC123",
4     "networkId": "N_123",
5     "name": "My HTTP server",
6     "url": "https://www.example.com/webhooks",
7     "sharedSecret": "foobar"
8   }
9 ]
```

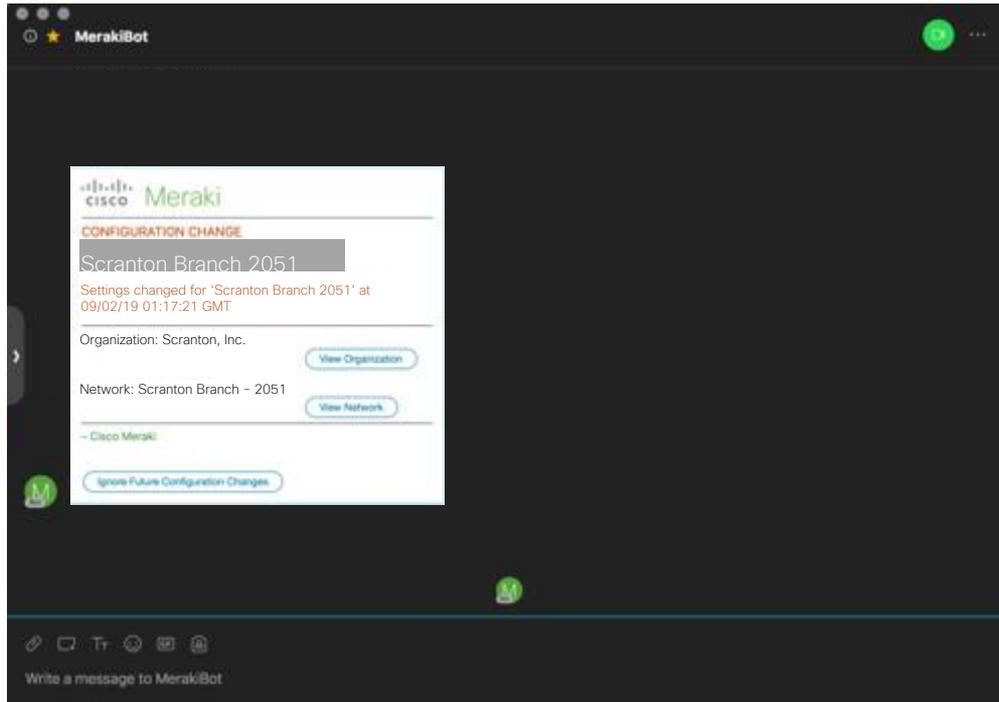


```
getNetworkAlertSettings

GET https://api.meraki.com/api/v0/networks/:networkId/alertSettings

1 - {
2   "defaultDestinations": {
3     "emails": [
4       "miles@meraki.com"
5     ],
6     "allAdmins": true,
7     "snmp": true
8   },
9   "alerts": [
10    {
11      "type": "gatewayDown",
12      "enabled": true,
13      "alertDestinations": {
14        "emails": [
15          "miles@meraki.com"
16        ],
17        "allAdmins": false,
18        "snmp": false
19      },
20      "filters": {
21        "timeout": 60
22      }
23    }
24  ]
25 }
```

Enables Powerful Cloud-to-cloud Integrations

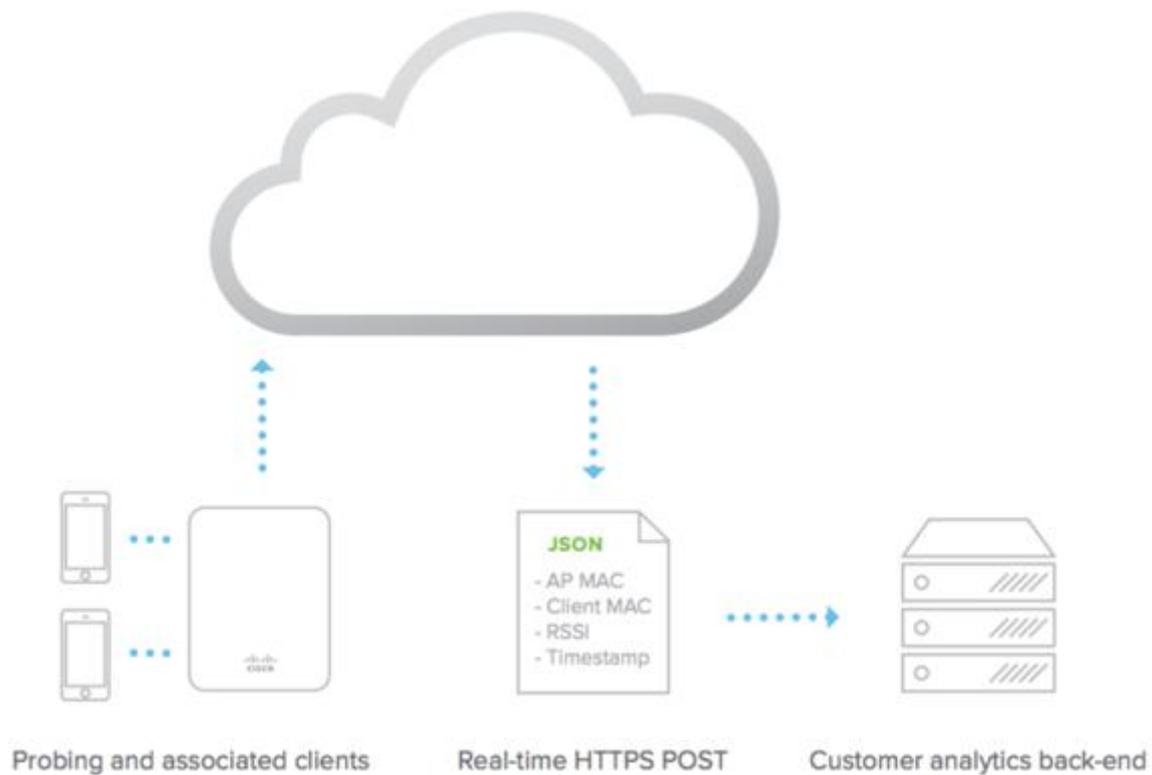


Current prototype of native
WebEx Teams Integration



Meraki Location API

Real time based data



Location API

Provides telemetry for WiFi and Bluetooth devices

DASHBOARD UI



LOCATION API

```
{  
  "ipv4": "/192.168.0.63",  
  "location": {  
    "lat": 51.5355157,  
    "lng": -0.06990350000000944,  
    "unc": 0.20755340376944298,  
    "x": ,  
    "y": ,  
  },  
  "seenTime": "2016-09-24T00:06:27Z",  
  "ssid": ".interwebs",  
  "os": null,  
  "clientMac": "18:fe:34:e1:b4:7a",  
  "seenEpoch": 1474675587,  
  "rssi": 56,  
  "ipv6": null,  
  "manufacturer": "Espressif"  
}
```

Payload

Access point metadata:

- AP
- AP tag(s)
- AP floor #'s

Client probe observations:

- IPv4 / IPv6 addy
- Cloud-calculated location
 - lat / long
 - x / y
- Seen time
- RSSI
- manufacturer

```
{
  {
    "version": "2.0",
    "secret": "supersecret",
    "type": "DevicesSeen",
    "data": {
      "apMac": "00:18:0a:13:dd:b0",
      "apFloors": [],
      "apTags": [
        "dev",
        "entrance",
        "office"
      ],
      "observations": [
        {
          "ipv4": "/192.168.0.38",
          "location": {
            "lat": 51.5355157,
            "lng": -0.06990350000000944,
            "unc": 1.233417960754815,
            "x": [],
            "y": []
          },
          "seenTime": "2016-09-24T00:06:23Z",
          "ssid": ".interwebs",
          "os": null,
          "clientMac": "18:fe:34:fc:5a:7f",
          "seenEpoch": 1474675583,
          "rssi": 47,
          "ipv6": null,
          "manufacturer": "Espressif"
        }
      ]
    }
  }
}
```

Setup & Authentication

1. Turn on the API by selecting Scanning API enabled in the dropdown box
2. Specify a post URL and the authentication secret
3. Specify which Scanning API version your HTTP server is prepared to receive and process
4. Configure and host your HTTP server to receive JSON objects
5. Upon the first connection, the Meraki cloud will perform a single HTTP GET to perform server validation

SUCCESS! Data should begin streaming

Location and scanning @

Analytics:

Scanning API:

Validator @

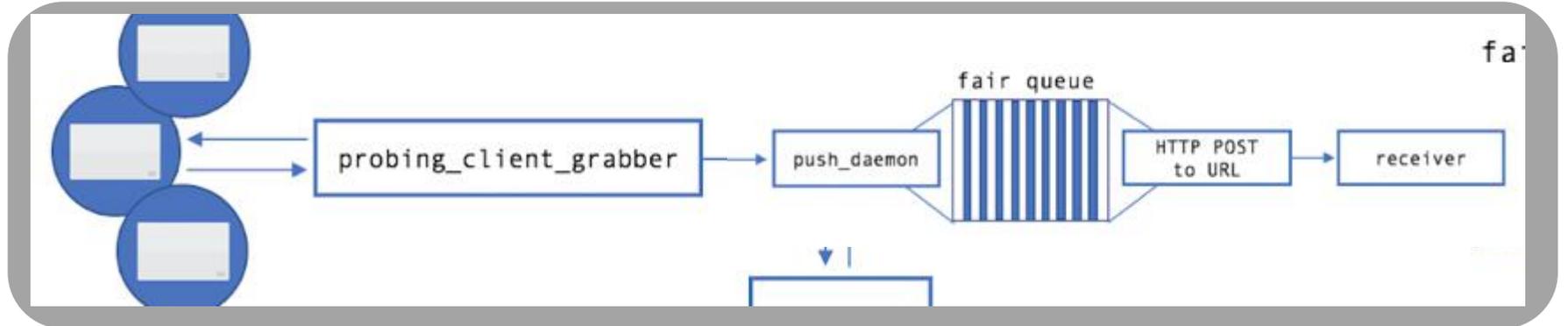
Post URLs @

Status	Post URL	Secret	API Version	
	cmsapi.example.com	Show secret 2.0	Validate X
	example.cmsapi.com	Show secret 2.0	Validate X

[Add a Post URL](#)



Architecture & Frequency



1 post per minute





EXCAP (Splash) API

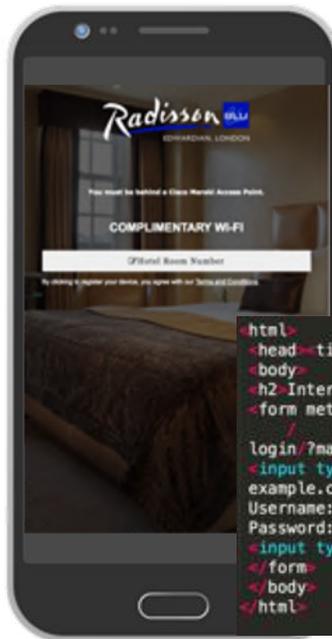
Captive Portal API

Create a custom WiFi login experience

MERAKI-HOSTED



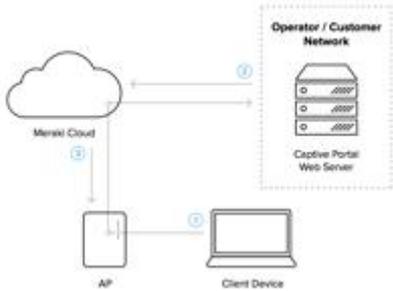
EXCAP API



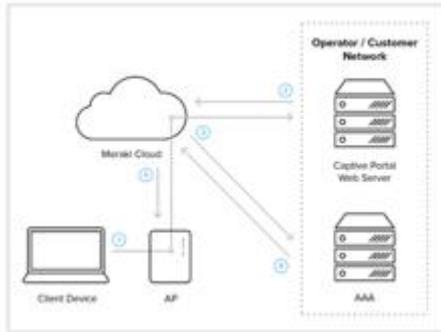
```
<html>
<head>title=Internet Access Login</title></head>
<body>
<h2>Internet Access Login</h2>
<form method=POST action="https://example.meraki.com/splash
/
login/?mauth=ABCDEFG123456">
<input type="hidden" name="success_url" value="http://www.
example.com/success/" />
Username: <input type="text" name="username" />
Password: <input type="text" name="password" />
<input type="submit" value="Login" />
</form>
</body>
</html>
```

Setup & Configuration

- Click-through



- Sign-on

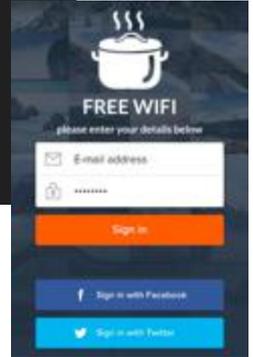


1. Configure a publicly-accessible webserver that is hosting a click-through or sign-on splash page

```
from flask import Flask, request, render_template, redirect, url_for, json

app = Flask(__name__)

global base_grant_url
base_grant_url = ""
global user_continue_url
user_continue_url = ""
global success_url
success_url = ""
```



2. Configure the splash URL within the Meraki Dashboard for the desired SSID

Custom splash URL

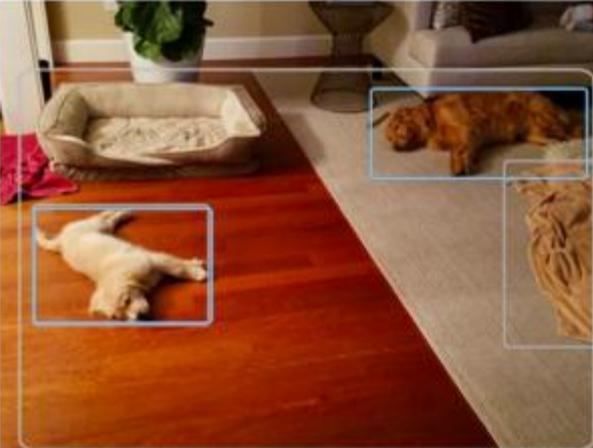
Or provide a URL where users will be redirected:

[What is this?](#)

MV Sense

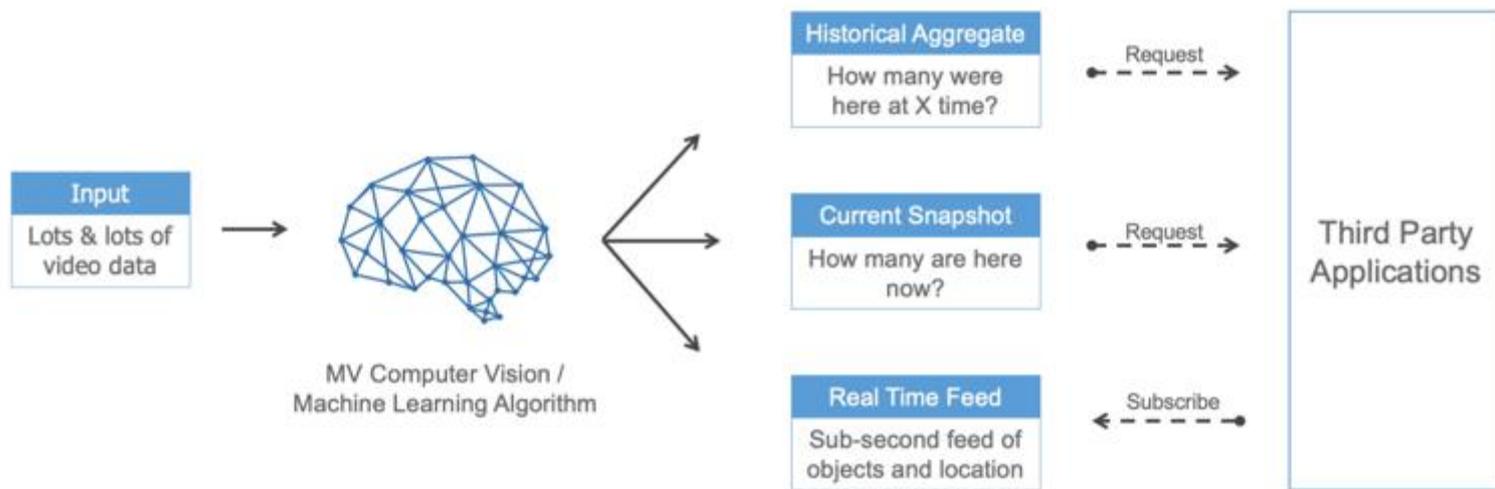


Object and scene detection
Intelligence automatically labels objects, concepts and scenes in your images, and provides a confidence score.



```
{"confidence": 97.9327281889936,  
  "instances": [  
    {  
      "boundingBox": {  
        "height": 249.6222449188822,  
        "left": 218.92277448294967,  
        "top": 178.24284117484712,  
        "width": 248.16880008922895  
      },  
      "confidence": 97.9327281889936  
    },  
    {  
      "boundingBox": {  
        "height": 234.2277288573368,  
        "left": 239.713813818882275,  
        "top": 46.515817164889225,  
        "width": 237.9341202544458  
      },  
      "confidence": 92.2781124392024  
    }  
  ],  
  "scene": "Dog",  
  "sceneScore": {  
    "scene": "Dog"  
  }  
}
```

Logic in MV Sense



MV object detection



MV Sense



GET

Get Device Camera Analy...

GET

Get Device Camera Analyti...

Cameras



POST

Generate Network Camera ...

GET

Get Device Camera Quality...

GET

Get Device Camera Video ...

GET

Get Network Camera Sche...

GET

Get Network Camera Video...

PUT

Update Device Camera Qu...

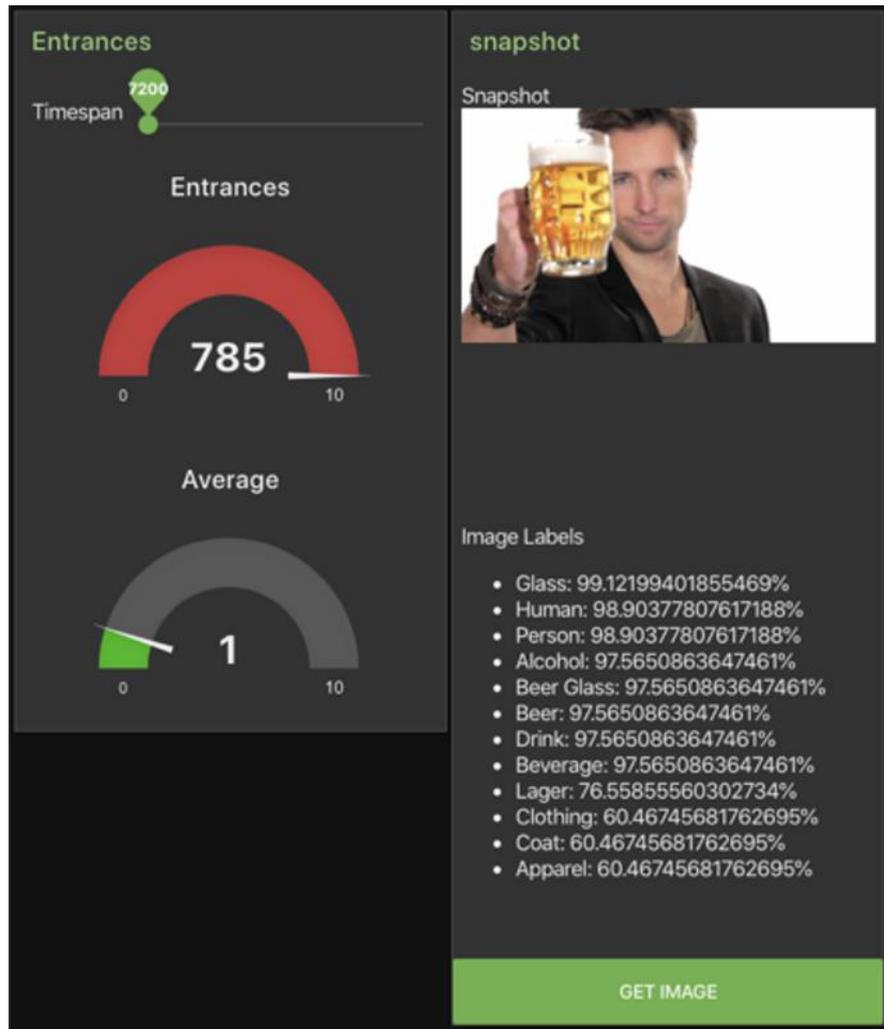
PUT

Update Device Camera Vid...

Custom Dashboard

Use Camera data with 3rd-party ML

- MV Sense
- AWS Rekognition
- Node-RED

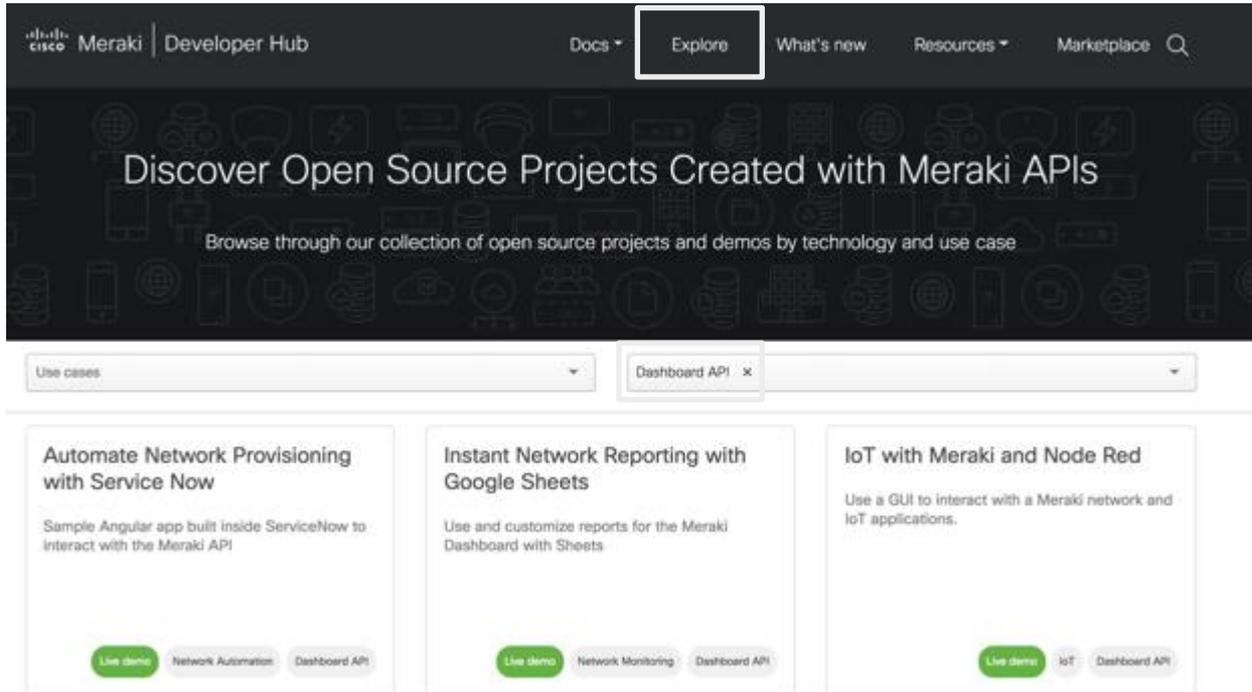


Demo Time

How to start?

Get Started on the DevNet Developer Hub!

 <https://meraki.io/explore> 



Meraki | Developer Hub Docs **Explore** What's new Resources Marketplace

Discover Open Source Projects Created with Meraki APIs

Browse through our collection of open source projects and demos by technology and use case

Use cases Dashboard API

Automate Network Provisioning with Service Now

Sample Angular app built inside ServiceNow to interact with the Meraki API

[Live demo](#) [Network Automation](#) [Dashboard API](#)

Instant Network Reporting with Google Sheets

Use and customize reports for the Meraki Dashboard with Sheets

[Live demo](#) [Network Monitoring](#) [Dashboard API](#)

IoT with Meraki and Node Red

Use a GUI to interact with a Meraki network and IoT applications.

[Live demo](#) [IoT](#) [Dashboard API](#)

Get Started on the DevNet Developer Hub!



https://meraki.io



Meraki Developer Hub

Meraki > API > Webhook API

Overview

Meraki Webhooks are a powerful and lightweight new way to subscribe to alerts sent from the Meraki Cloud when an event occurs. These include a JSON formatted message and are sent to a unique URL, where they can be processed, stored or used to trigger powerful automations. For a comprehensive list of alerts, please see our general documentation on [Monitoring and Alerting](#).

Setup and Configuration

1. Add one or more **HTTP servers**. Include a name, unique HTTPS (TLS) URL and shared secret (see **Security** below for more details). [network-wide > alerts]

HTTP server	Name	URL	Shared secret
1	Super	https://example.com	
2	Sub	https://example.com	
Add an HTTP server			

```
JS
if (meraki.sharedSecret === "asdf1234"){
  saveData(meraki);
}else{
  // invalid secret
}
```

- Getting started
- Dashboard API
- Webhooks
- Captive Portal API
- Location Services
- MV Sense

- Learning Labs
- Sandbox
- Meraki GitHub
- Code Exchange
- Ecosystem Exchange
- Meraki Community

Thank you