Meraki API
Agenda

● Intro
  ○ Why programing
  ○ 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

<table>
<thead>
<tr>
<th>Business Value</th>
<th>Low</th>
<th>High</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business Risk</td>
<td>Low</td>
<td>High</td>
</tr>
<tr>
<td>Green Field</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Monitoring</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Design</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Optimization</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Application</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Performance</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Security</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Troubleshooting</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Compliance</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dynamic</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Configuration</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Change Control</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operations</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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
Requests and Response REST API Flow
Overview of API

APIs help developers create apps that benefit the end user.

- Map Server returns map data via API
- Restaurant Recommendation app asks a Map Server for map data
- Users see list of restaurants close to them
URI Structure

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.

Request: GET https://api.ciscospark.com/v1/people/me

Response: 200 OK + Data
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?

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

Reference: [https://restfulapi.net/http-methods/](https://restfulapi.net/http-methods/)
## Response Status Codes: Did it work?

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

Reference: [https://restfulapi.net/http-status-codes/](https://restfulapi.net/http-status-codes/)
### Headers:
What additional details and metadata can I use?

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

- 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 Platform

OUT-OF-THE-BOX MANAGEMENT & ANALYTICS

{ APIs }

INTEGRATIONS & SOLUTIONS
POWERED BY MERAKI

AUTOMATION
APPS
ANALYTICS
ENGAGEMENT
TRACKING
AND MORE

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

REST DASHBOARD API
Configuration & Metadata
- Programmability & Automation
  Make a configuration change to 1,100 ports
- Monitoring & Automation
  Status of all 2,200 Meraki devices in organization
- Reporting
  Visualize what % of clients are on WiFi vs Wired
- Data Insights (IQ)
  Processed data insights that deliver key new business insights

WEBHOOK API
Event-triggered POST notifications
- Event stream
- Automation trigger

LOCATION STREAMING API
Wi-Fi & BLE Location Telemetry
- Wayfinding
- Asset Tracking
- Location Analytics

CAPTIVE PORTAL API
Create custom guest Wi-Fi splash interactions
- Guest Wi-Fi
- Secure Onboarding
Driven by Powerful, Developer-friendly APIs

MV SENSE
Visual Sensor API

- Real-time (4Hz) data stream
  # People or vehicles in the frame

- Historical time-series via REST
  How many people were seen yesterday at 6pm
Dashboard API (REST)
Dashboard API
Programmatically manage and report on your network

<table>
<thead>
<tr>
<th>DASHBOARD UI</th>
<th>REST API</th>
</tr>
</thead>
<tbody>
<tr>
<td>20 2.55 MR52</td>
<td></td>
</tr>
<tr>
<td>21 2.25 MR52</td>
<td></td>
</tr>
<tr>
<td>22 2.54 MR52</td>
<td></td>
</tr>
<tr>
<td>23 2.53 MR52</td>
<td></td>
</tr>
<tr>
<td>24 5.10 MR42</td>
<td></td>
</tr>
<tr>
<td>25 5.36 - Axe MR42</td>
<td></td>
</tr>
<tr>
<td>26 5.12 MR53</td>
<td></td>
</tr>
</tbody>
</table>

```
"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"
```

```
"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"
```
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)

V0: Current GA Version
290+ Supported endpoints, 75K developers

V0.minor (Monthly)
New features, enhancements (non-breaking)

V1: In Development
New Major Release for Dashboard API
A new major release with a focus on consistency, intuitive structure, efficiency and scale (includes breaking changes)

Developer preview: November 2019

Planned GA Launch: February 2020

Dual-supported for 1 Year

View @ meraki.io/whats-new
Example 1: Org-wide Device Inventory (CMDB)

Object with nested array of all devices in global organization

Can support 25,000+ devices in synchronous response
Example 2: Org-wide Status Monitoring

Nested array containing all devices in global organization, with connectivity and alerting status.
Example 3: ActionBatch API (20 x Network Create)

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

<table>
<thead>
<tr>
<th>APs came up</th>
<th>Sample</th>
</tr>
</thead>
<tbody>
<tr>
<td>APs went down</td>
<td>Sample</td>
</tr>
<tr>
<td>Action Required: Upcoming firewall information changes</td>
<td>Sample</td>
</tr>
<tr>
<td>Air Marshal - Rogue AP detected</td>
<td>Sample</td>
</tr>
<tr>
<td>Bluetooth Clients have become visible</td>
<td>Sample</td>
</tr>
</tbody>
</table>

```
{
  "version": "0.1",
  "sharedSecret": "foo",
  "sentAt": "2019-07-18T22:41:27.571212Z",
  "organizationId": "00000000",
  "organizationName": "Miles Monitoring Inc.",
  "organizationUrl": "https://n1.meraki.com/o/manage/organization",
  "networkId": "N1111111111111",
  "networkName": "Main Office",
  "networkUrl": "https://n1.meraki.com/n/manage/nodes/list",
  "deviceSerial": "XXXX-XXXX-XXXX",
  "deviceMac": "00:00:00:00:00:00",
  "deviceName": "Device Foo Bar",
  "deviceUrl": "https://n1.meraki.com/n/manage/nodes/new_list",
  "alertType": "APs came up",
  "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 **5 GB** in **20 minutes**

**Wireless Alerts**
- A gateway goes offline for **5 minutes**
- A repeater goes offline for **5 minutes**
- A gateway becomes a repeater

**Security & SD-WAN Alerts**
- A security appliance goes offline for **5 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 **5 minutes**
- A new DHCP server is detected on the network
- Important goes down for more than **5 minutes**
- Any port detects a cable error
- Any port 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

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

1 post per minute

AP probe collector (per radio) → Cloud Aggregator Microservice → Client / Webserver
EXCAP (Splash) API
Captive Portal API

Create a custom WiFi login experience

MERAKI-HOSTED

EXCAP API
Setup & Configuration

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

```python
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 = ""
```

- **Sign-on**
  2. Configure the splash URL within the Meraki Dashboard for the desired SSID

![Meraki Dashboard](image)
Logic in MV Sense

Input
Lots & lots of video data

MV Computer Vision / Machine Learning Algorithm

Output

Historical Aggregate
How many were here at X time?

Current Snapshot
How many are here now?

Real Time Feed
Sub-second feed of objects and location

Third Party Applications

Request

Subscribe
MV object detection
MV Sense

Camera as a Sensor with built-in machine learning for object detection
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
Get Started on the DevNet Developer Hub!

https://meraki.io
Thank you