Meraki Automation Using Python Workshop

Cisco Meraki

Introductions

Agenda

- Meraki Platform
- Meraki API Types
- Meraki API using Python
- Final Exercise
- Next Steps

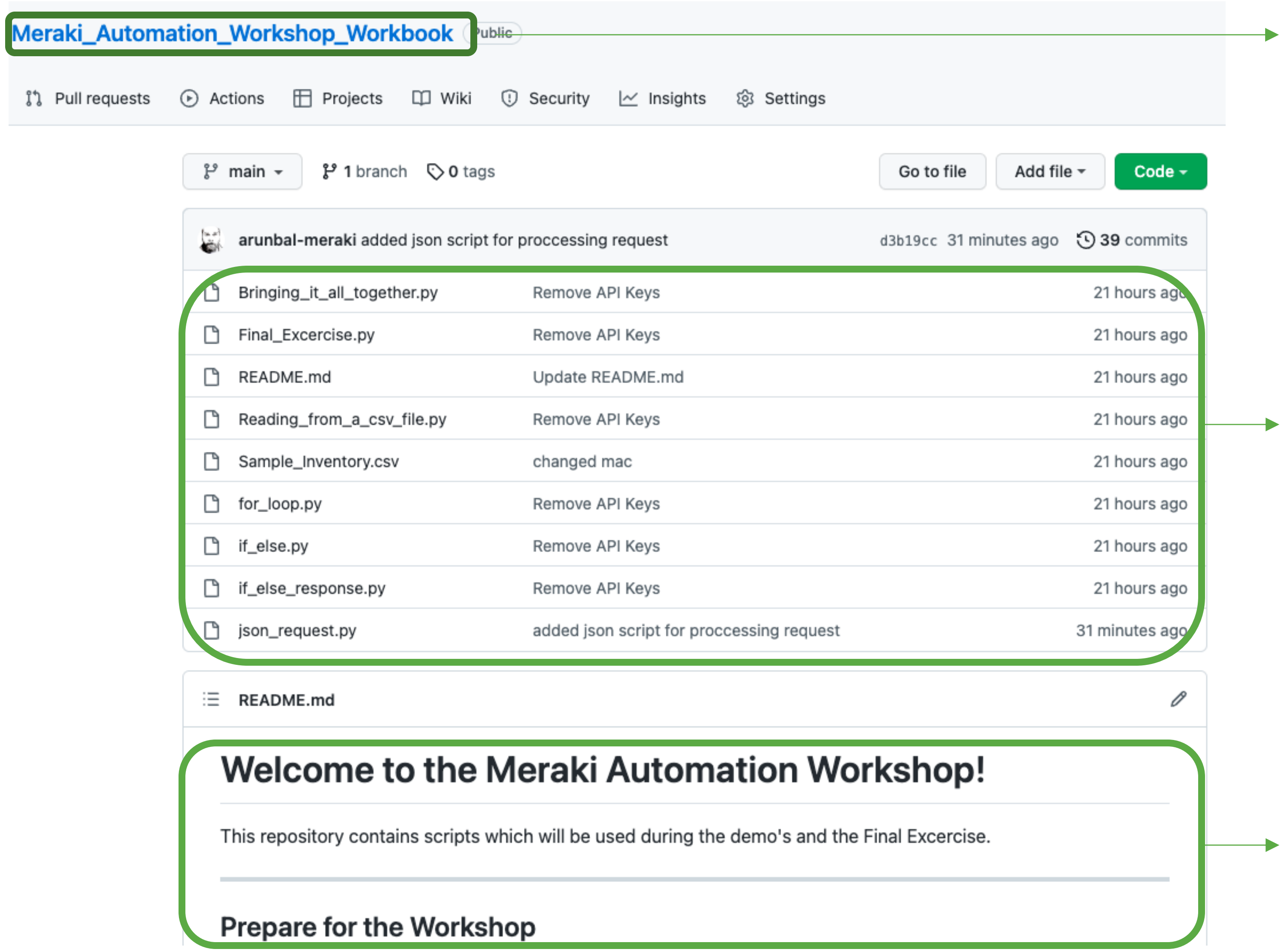
Recommended

Feel free to follow along in an environment of your choice

- Create an account on Meraki Developer Hub
- If already installed, use Python on your machine or V
- Use pythonanywhere
- Install Postman or use web version
- **Access the lab workbook on Github here**

Workshop Workbook

All the code which will be used in this workshop is available here. Feel free to follow along during the workshop or practice later.



arunbal-meraki added json script for proccessing request d3b19cc 31 minutes ago 39 commits

Bringing_it_all_together.py	Remove API Keys	21 hours ago
Final_Excercise.py	Remove API Keys	21 hours ago
README.md	Update README.md	21 hours ago
Reading_from_a_csv_file.py	Remove API Keys	21 hours ago
Sample_Inventory.csv	changed mac	21 hours ago
for_loop.py	Remove API Keys	21 hours ago
if_else.py	Remove API Keys	21 hours ago
if_else_response.py	Remove API Keys	21 hours ago
json_request.py	added json script for proccessing request	31 minutes ago

Welcome to the Meraki Automation Workshop!

This repository contains scripts which will be used during the demo's and the Final Excercise.

Prepare for the Workshop

Workbook repository

Code files

Workshop Guide

The Meraki Platform

The Meraki Platform

Connecting passionate people to their mission by **simplifying** the digital workspace

SIMPLE

Increased productivity and error reduction

SECURE

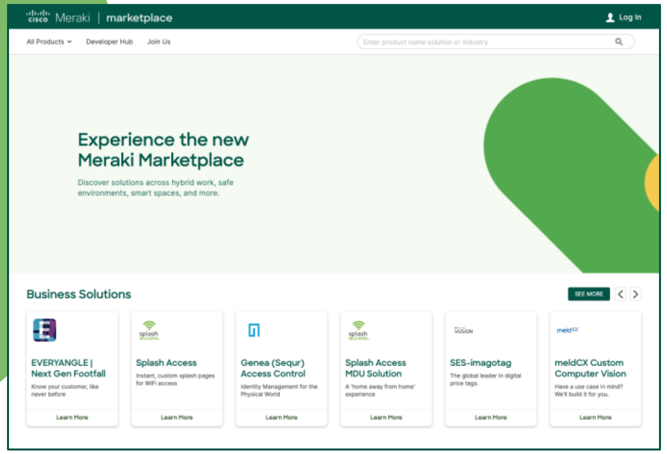
Efficient and reliable policy visibility

INTELLIGENT

Dynamic and scalable policy automation

Ease Through Convergence with the Meraki Platform

The **power of the platform** pulls together IT, IoT, and physical security domains



Experience the new Meraki Marketplace

Discover solutions across hybrid work, safe environments, smart spaces, and more.

Business Solutions

EVERYANGLE | Next Gen Footfall
Know your customer, like never before

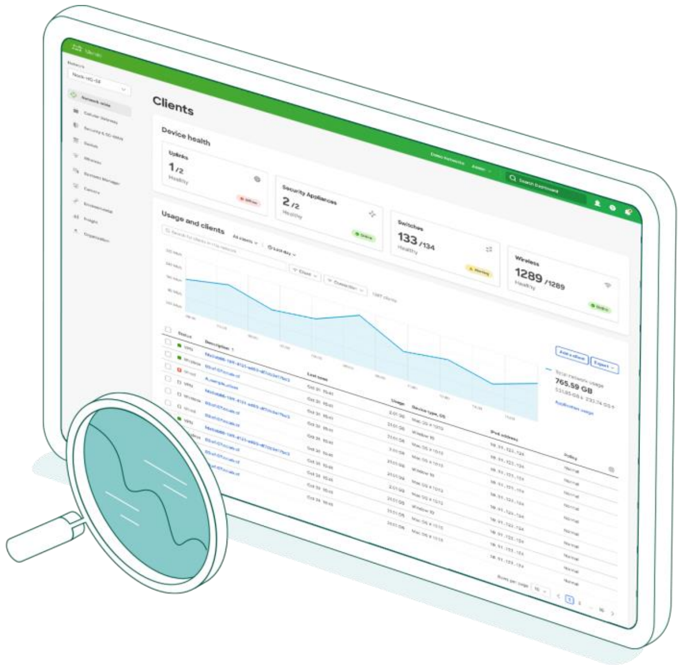
Splash Access
Instant, custom splash pages for WiFi access

Genea (Sequr) Access Control
Identity Management for the Physical World

Splash Access MDU Solution
A 'home away from home' experience

SES-imagotag
The global leader in digital price tags.

meldCX Custom Computer Vision
Have a use case in mind? We'll build it for you.



NEXT-GEN ACCESS | SASE | IoT | AI / ML

Meraki API Types

Meraki API Types

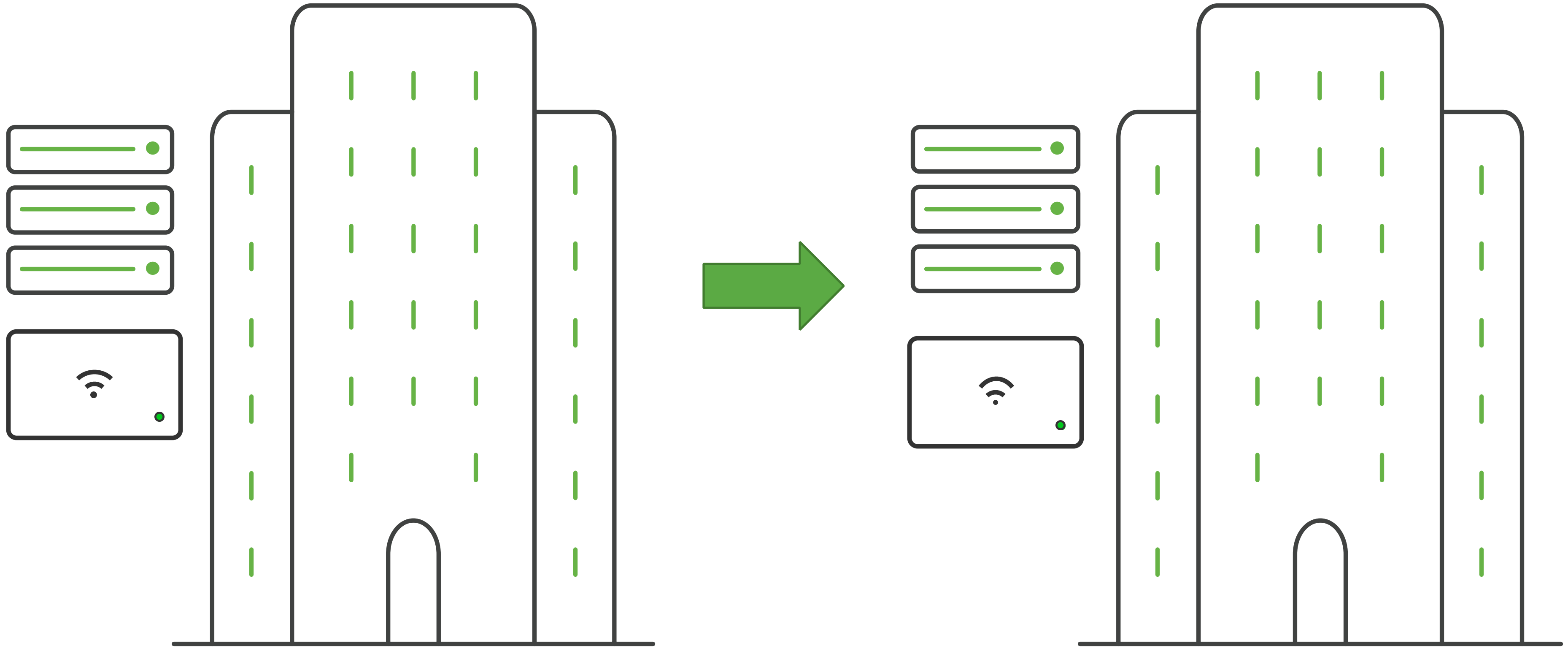
- Meraki Dashboard API
- Scanning API
- MV Sense
- Captive Portal
- Webhooks

How Do You Plan On Using Meraki APIs?

Scenario

Scenario

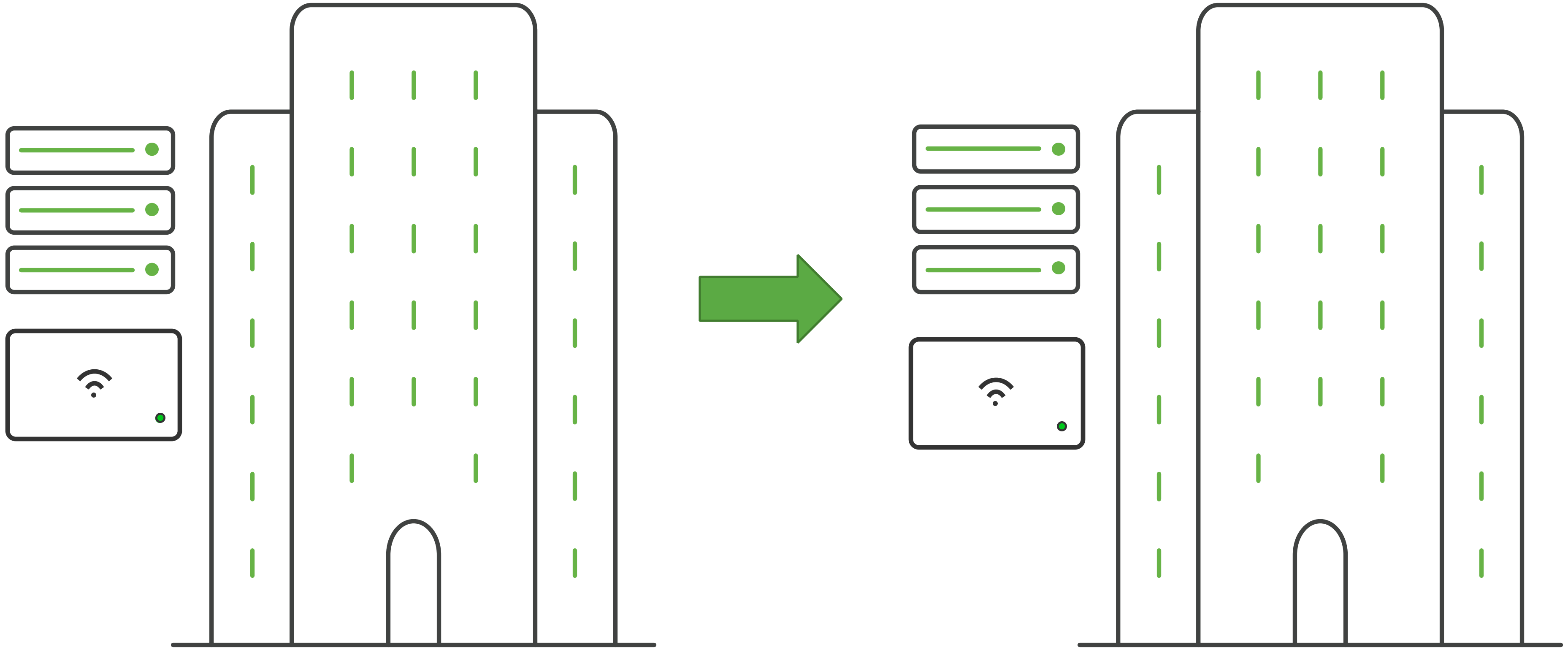
Imagine a customer has a current Meraki network with switches and wireless in one of their sites. They would like to clone the config from this site when they set up subsequent sites.



Scenario

What options are available for this customer?

- Network templates?
- Network clone?
- API?



Scenario Demo

Reasons to Use Meraki APIs

Can only purchase 50% of planned hardware because they are not staffed to deploy it fast enough

Engineers can't work on the next project because they are too busy supporting the last one

Change freeze because another outage last week caused by changes and/or human error

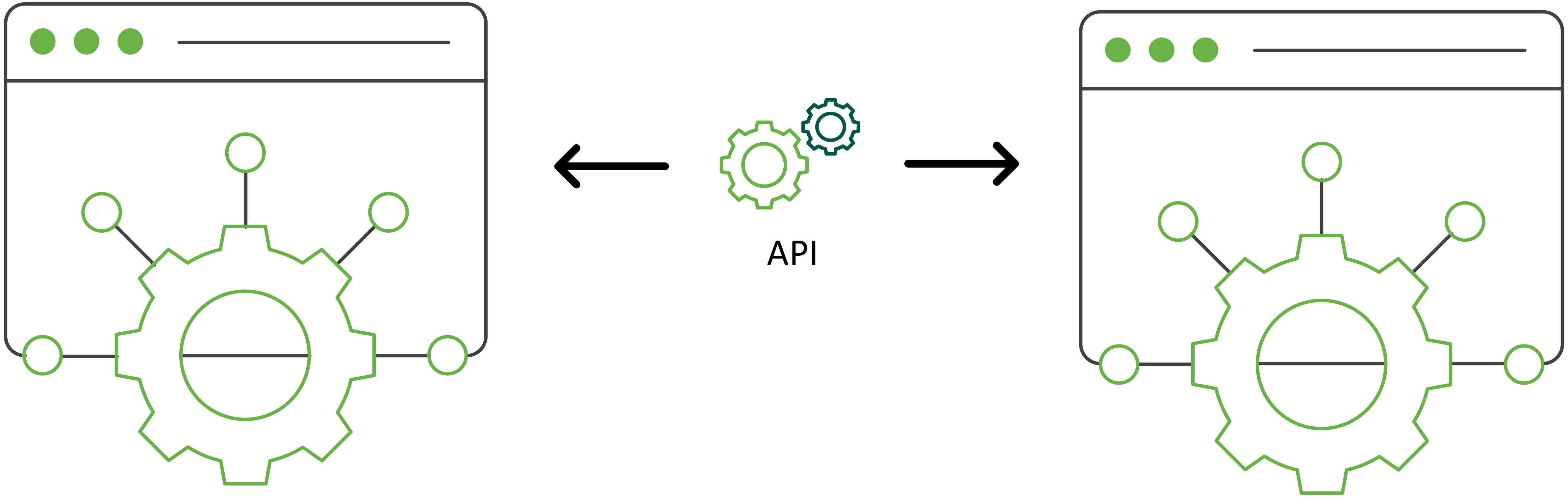
Developers moving everything to public cloud because it takes too long setting up new environments

Can't do X so we need to look at other options as well

Application Programming Interface (API)

Application Programming Interface (API)

A connection provided by an application to provide services.



RESTful API

RESTful API

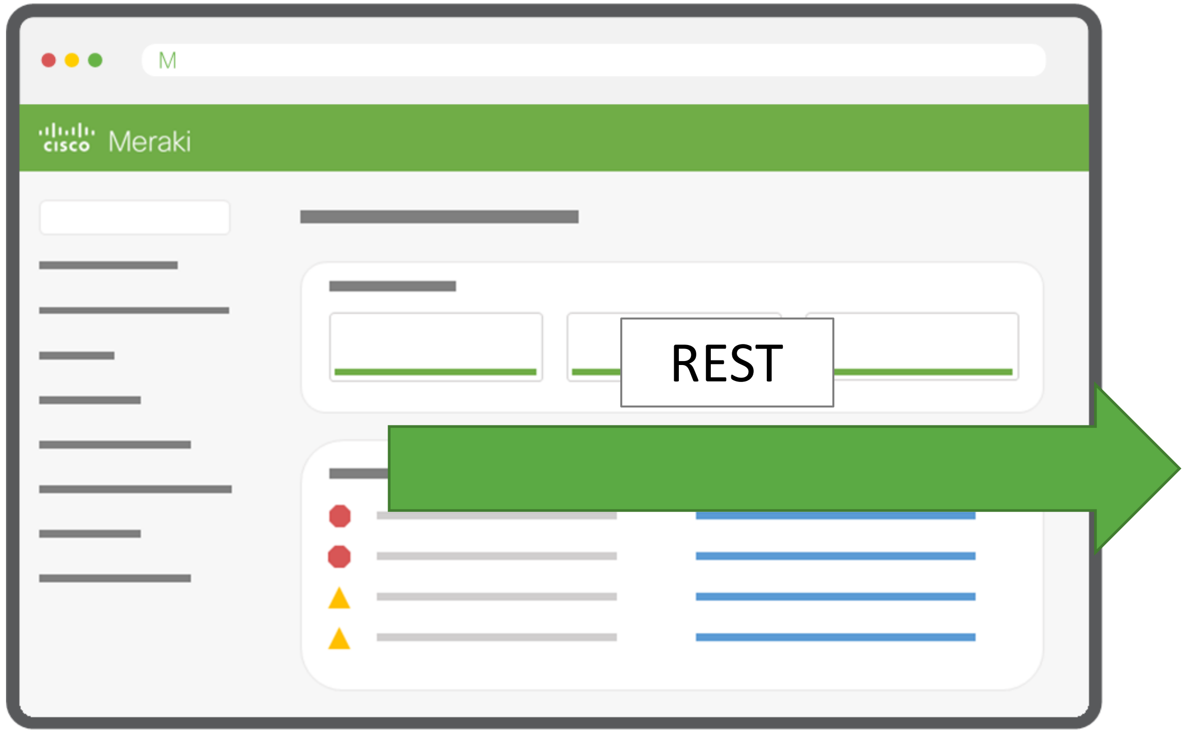
“REST is a set of **architectural constraints**, not a protocol or a standard. API developers can implement REST in a variety of ways.

When a client request is made via a RESTful API, it transfers a representation of the state of the resource to the requester or endpoint.”

https://www.redhat.com/en/topics/api/what-is-a-rest-api

RESTful API

The Meraki dashboard uses RESTful API

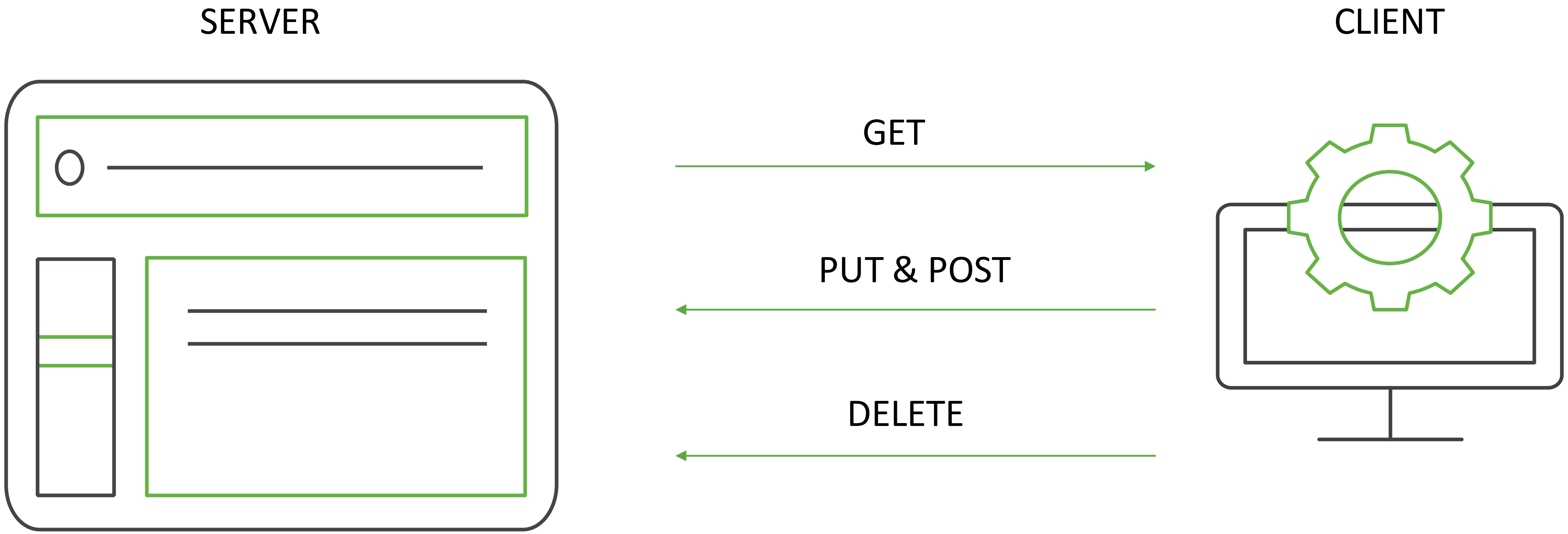


REST

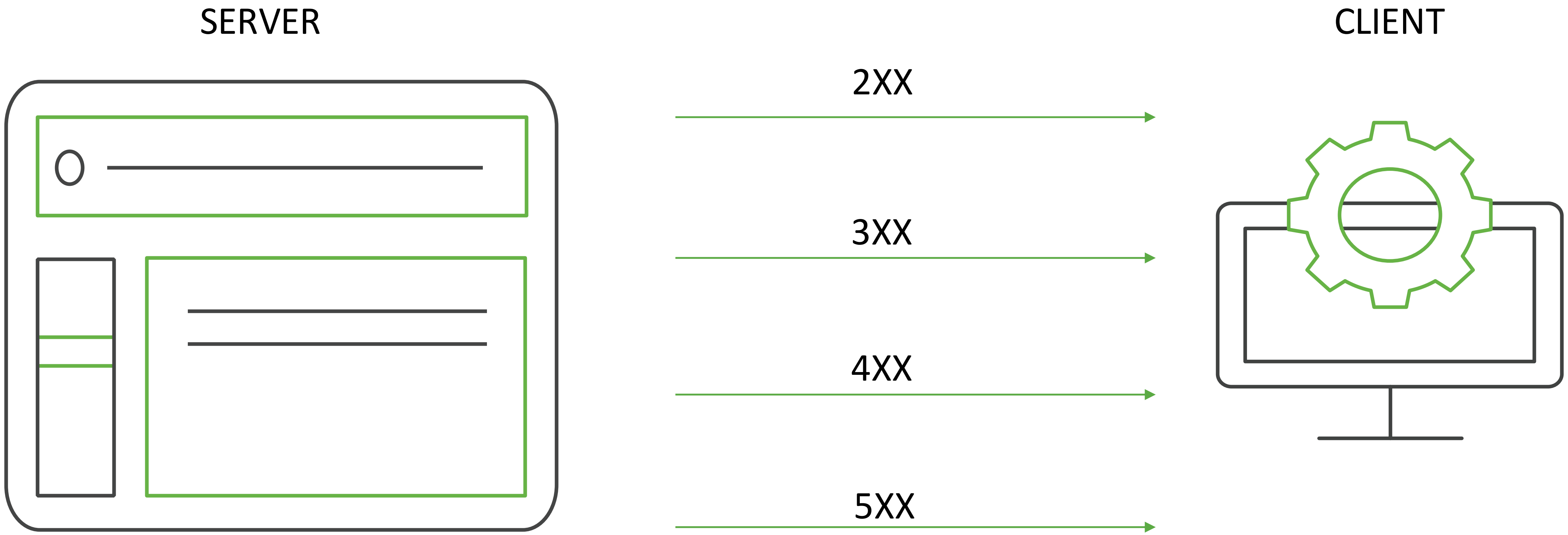
HTTP & JSON

HTTP

HTTP Methods



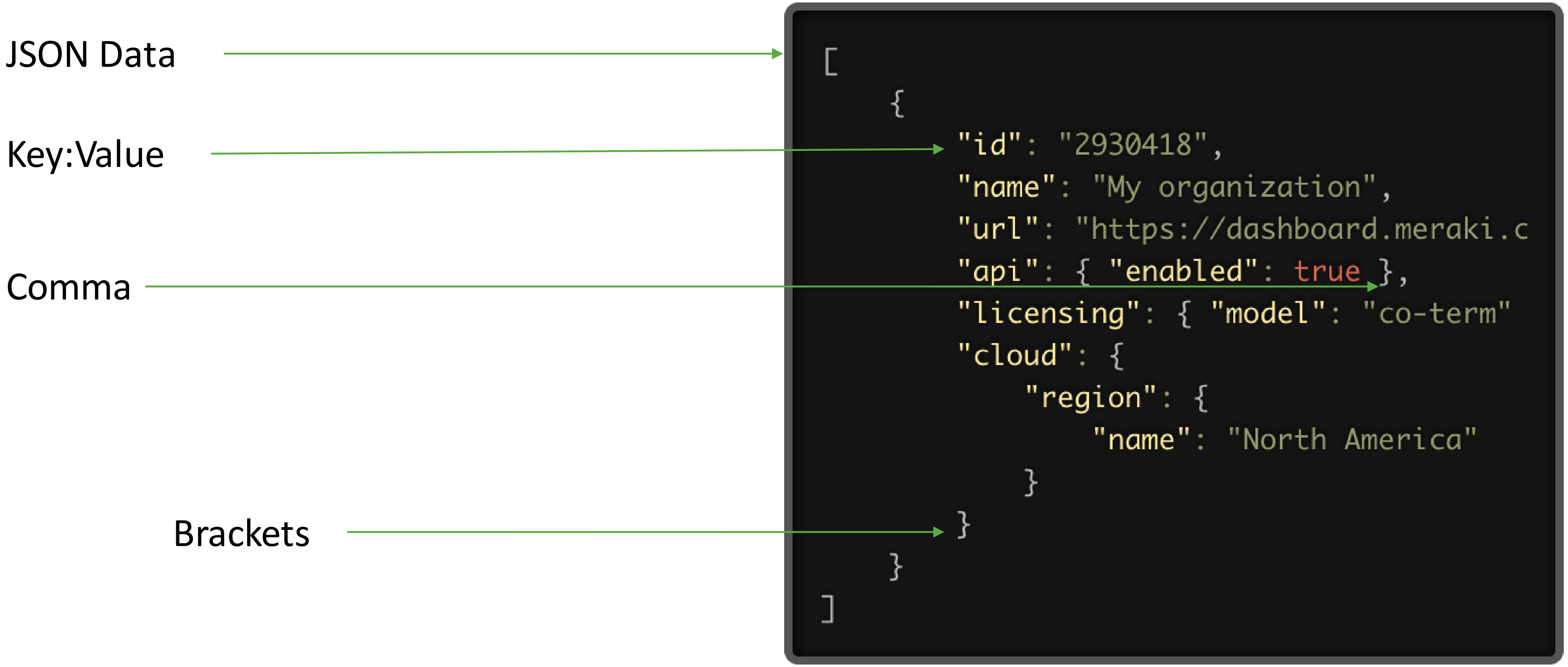
HTTP Response Codes



JSON

JavaScript Object Notation (JSON)

REST APIs send and receive data in JSON format.



JSON Data

Key:Value

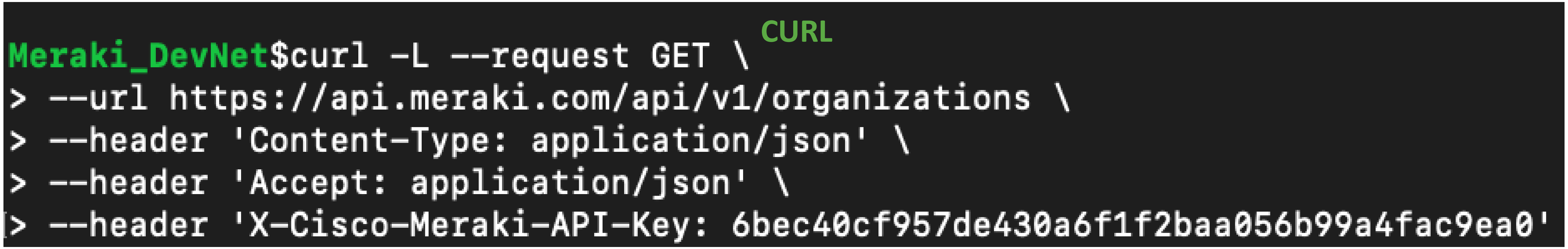
Comma

Brackets

```
[
    {
        "id": "2930418",
        "name": "My organization",
        "url": "https://dashboard.meraki.c
        "api": { "enabled": true },
        "licensing": { "model": "co-term"
        "cloud": {
            "region": {
                "name": "North America"
            }
        }
    }
]
```

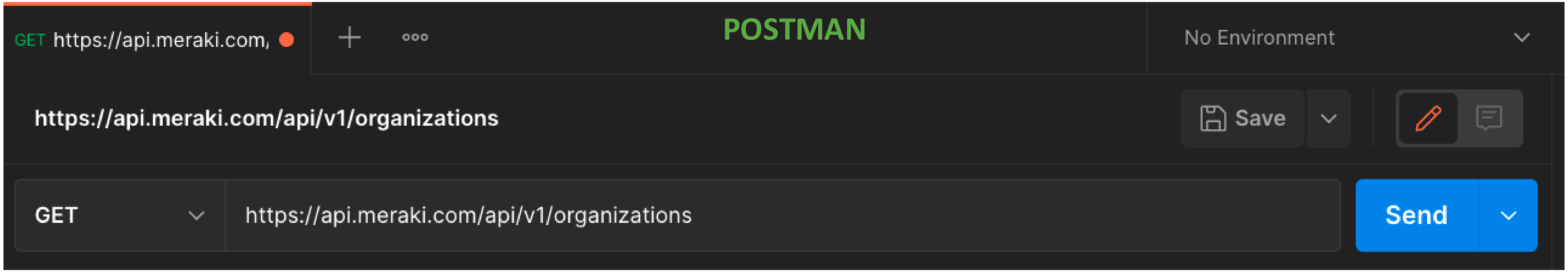
Tools for REST API

CURL and Postman are popular tools for learning, testing, and building API.



CURL

```
Meraki_DevNet$curl -L --request GET \
> --url https://api.meraki.com/api/v1/organizations \
> --header 'Content-Type: application/json' \
> --header 'Accept: application/json' \
> --header 'X-Cisco-Meraki-API-Key: 6bec40cf957de430a6f1f2baa056b99a4fac9ea0'
```



POSTMAN

GET https://api.meraki.com,

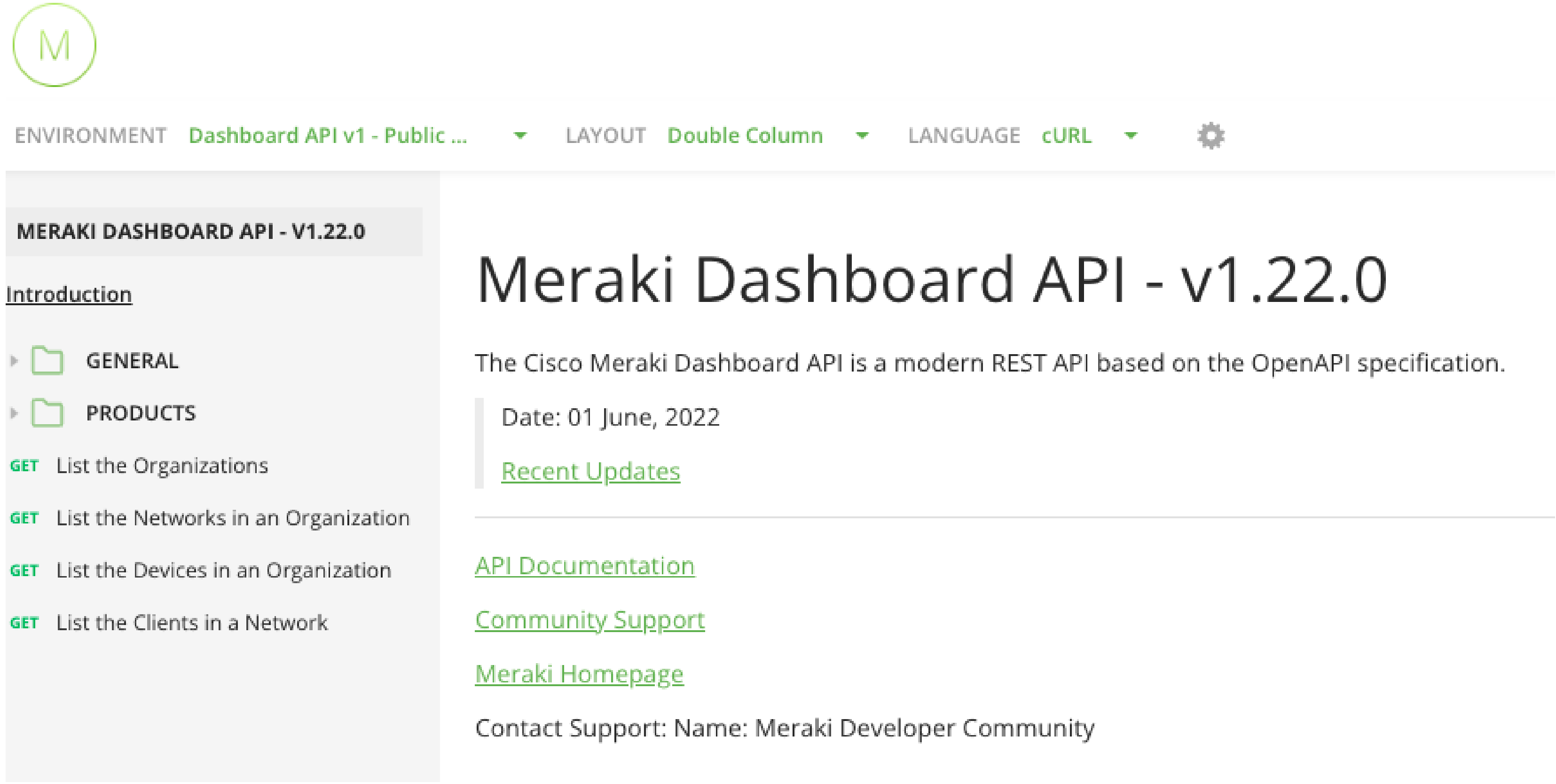
https://api.meraki.com/api/v1/organizations

GET https://api.meraki.com/api/v1/organizations

Meraki Postman Collection

You can import the Meraki Postman collection.

https://documenter.getpostman.com/view/897512/SzYXYfmJ



ENVIRONMENT Dashboard API v1 - Public ... LAYOUT Double Column LANGUAGE cURL

MERAKI DASHBOARD API - V1.22.0

Introduction

- GENERAL
- PRODUCTS
- GET List the Organizations
- GET List the Networks in an Organization
- GET List the Devices in an Organization
- GET List the Clients in a Network

Meraki Dashboard API - v1.22.0

The Cisco Meraki Dashboard API is a modern REST API based on the OpenAPI specification.

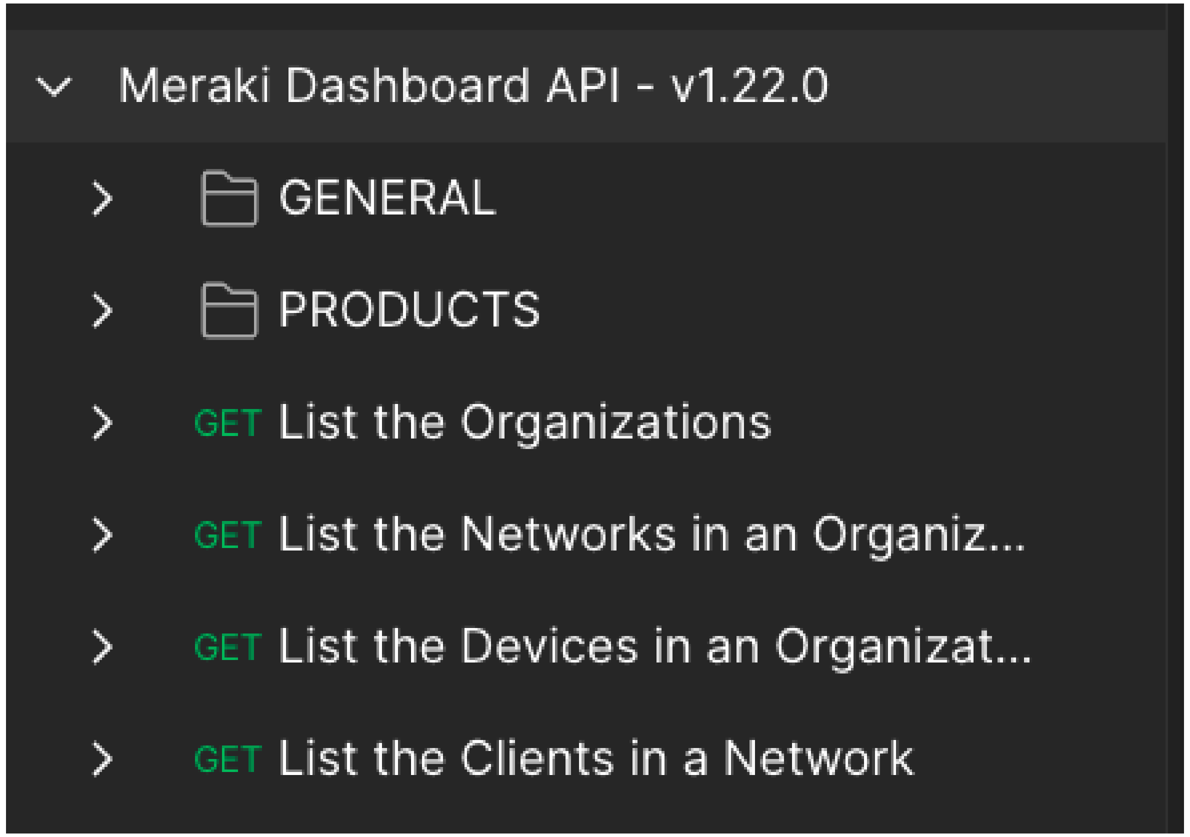
> Date: 01 June, 2022
>
> Recent Updates

API Documentation

Community Support

Meraki Homepage

Contact Support: Name: Meraki Developer Community



Meraki Dashboard API - v1.22.0

- GENERAL
- PRODUCTS
- GET List the Organizations
- GET List the Networks in an Organiz...
- GET List the Devices in an Organizat...
- GET List the Clients in a Network

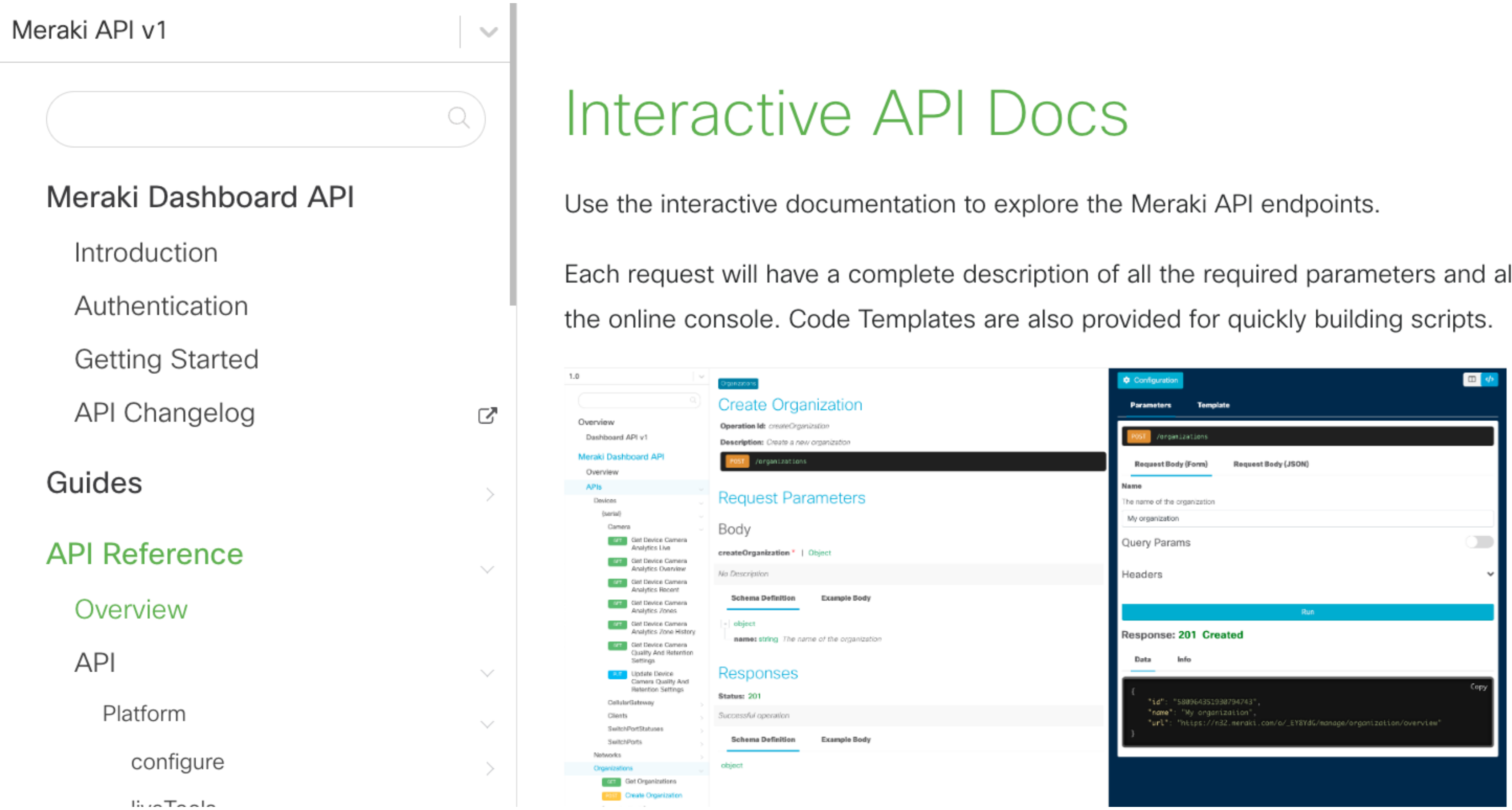
API Requests Using cURL and Postman Demo

Slido Quiz

API Documentation

API Documentation

Use the extensive API documentation available online and on the dashboard.



Interactive API Docs

Use the interactive documentation to explore the Meraki API endpoints.

Each request will have a complete description of all the required parameters and also allow you to instantly try it out in the online console. Code Templates are also provided for quickly building scripts.

1.0
Overview
Dashboard API v1
Meraki Dashboard API
Overview
APIs
Devices
{serial}
Camera
GET Get Device Camera Analytics Live
GET Get Device Camera Analytics Overview
GET Get Device Camera Analytics Recent
GET Get Device Camera Analytics Zones
GET Get Device Camera Analytics Zone History
GET Get Device Camera Quality And Retention Settings
PUT Update Device Camera Quality And Retention Settings
CellularGateway
Clients
SwitchPortStatuses
SwitchPorts
Networks
Organizations
GET Get Organizations
POST Create Organization
Organizations

Create Organization

Operation Id: *createOrganization*

Description: *Create a new organization*

```
POST /organizations
```

Request Parameters

Body

createOrganization * | Object

No Description

object

name: string *The name of the organization*

Responses

Status: 201

Successful operation

object

```
POST /organizations
```

Name

The name of the organization

My organization

Response: 201 Created

```
{
    "id": "580964351938794743",
    "name": "My organization",
    "url": "https://n32.meraki.com/o/_EY8YdG/manage/organization/overview"
}
```

OPENAPI INITIATIVE

https://developer.cisco.com/meraki/api-v1/

API Documentation Demo

Enabling API on the Dashboard

Enabling API on the Dashboard

- ✓ Login
- ✓ Go to your profile
- ✓ Generate Key

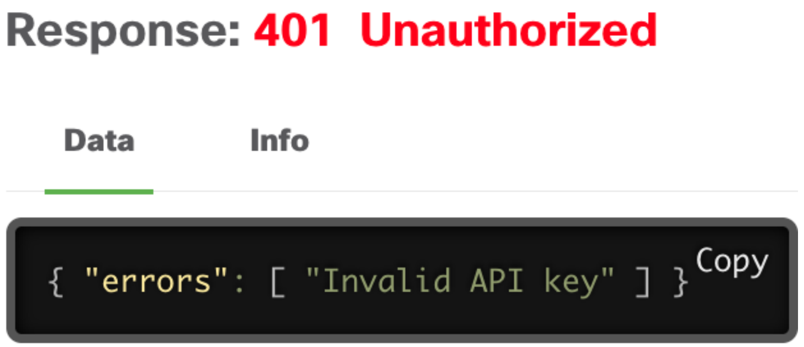
Enabling API on the Dashboard Demo

API Requests

- Every request must specify an API key via a request header.
- The API key must be specified in the URL.
- The API version must be specified in the URL
- 401 response for invalid key.

X-Cisco-Meraki-API-Key: <secret key>

https://api.meraki.com/api/v1/<resource>



Response: 401 Unauthorized

```
{ "errors": [ "Invalid API key" ] }
```

Interacting with the Dashboard API using Python

Python

“Python is a programming language that lets you work quickly and integrate systems more effectively.”

https://www.python.org/

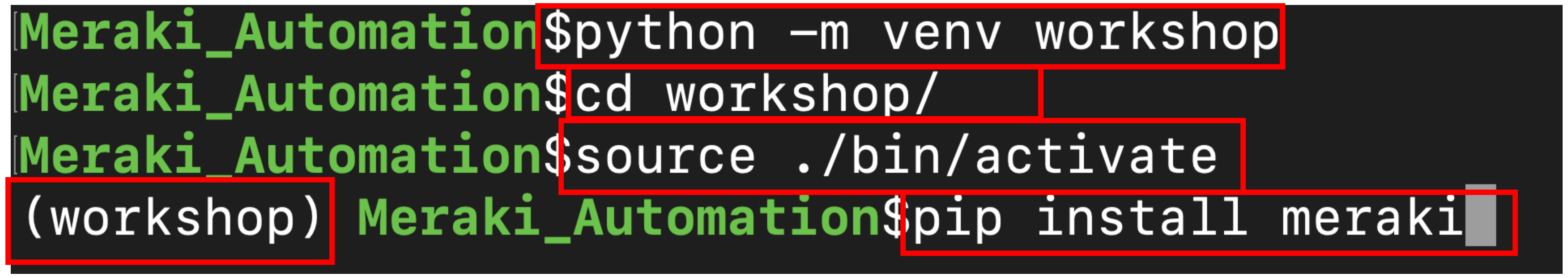
Python Editors

- Visual Studio Code
- PyCharm
- Atom
- IDLE
- Text Editor - vim/nano

Virtual Environment (venv)

Use virtual environments to avoid dependencies and run in isolation.

- Create a venv
- Navigate to the directory
- Activate the venv
- Prompt shows venv
- Install packages in venv

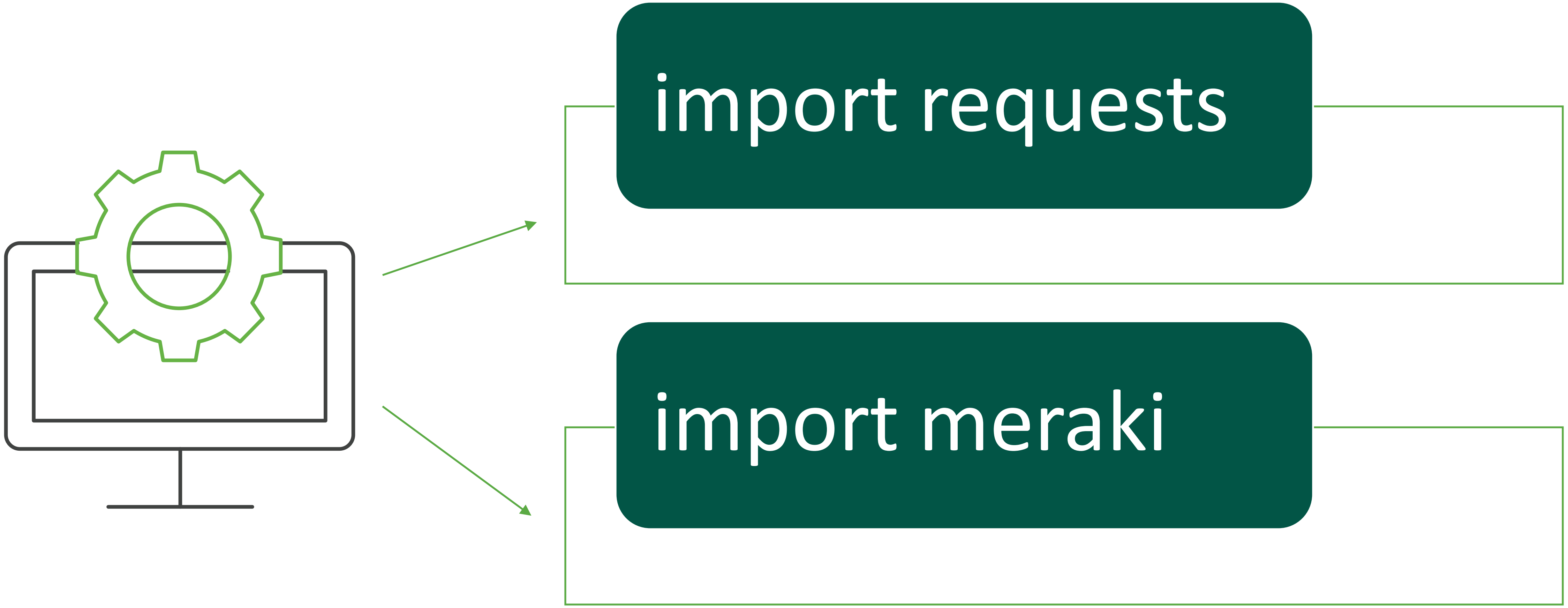


```
[Meraki_Automation$python -m venv workshop
[Meraki_Automation$cd workshop/
[Meraki_Automation$source ./bin/activate
(workshop) Meraki_Automation$pip install meraki
```

Python Virtual Environment Demo

Python with Meraki

Two libraries

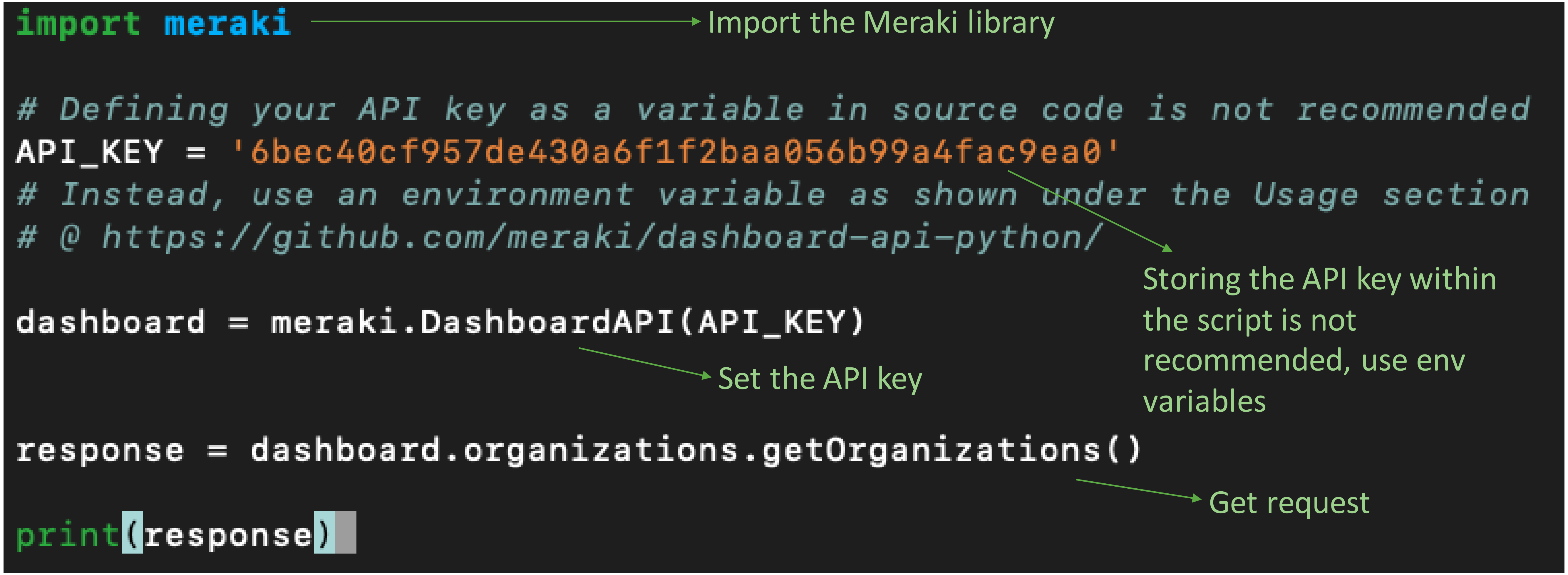


import requests

import meraki

*Note: Install the Meraki library using “pip install meraki”

Python Request and Meraki Library



```
import meraki

# Defining your API key as a variable in source code is not recommended
API_KEY = '6bec40cf957de430a6f1f2baa056b99a4fac9ea0'
# Instead, use an environment variable as shown under the Usage section
# @ https://github.com/meraki/dashboard-api-python/

dashboard = meraki.DashboardAPI(API_KEY)

response = dashboard.organizations.getOrganizations()

print(response)
```

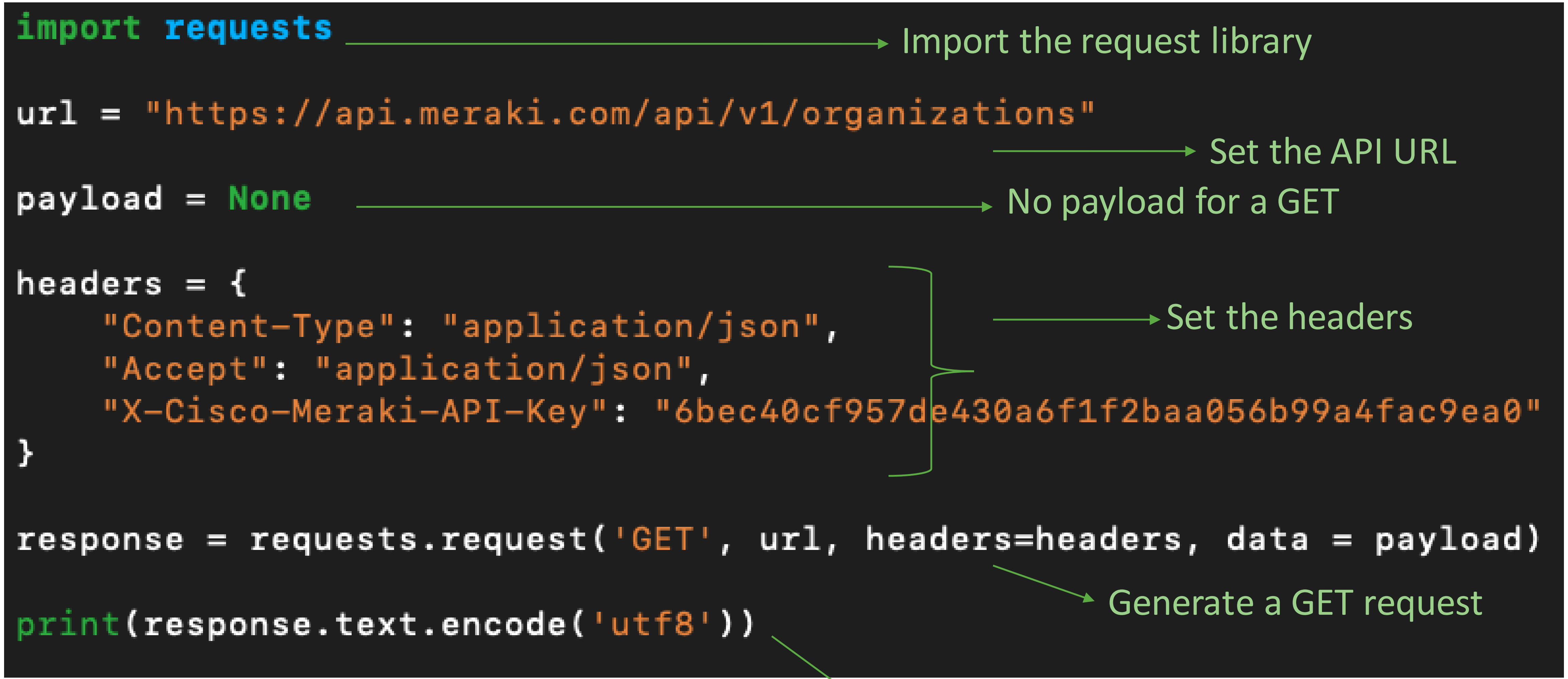
Import the Meraki library

Storing the API key within the script is not recommended, use env variables

Set the API key

Get request

Python Request and Requests Library



```
import requests

url = "https://api.meraki.com/api/v1/organizations"

payload = None

headers = {
    "Content-Type": "application/json",
    "Accept": "application/json",
    "X-Cisco-Meraki-API-Key": "6bec40cf957de430a6f1f2baa056b99a4fac9ea0"
}

response = requests.request('GET', url, headers=headers, data = payload)

print(response.text.encode('utf8'))
```

Import the request library

Set the API URL

No payload for a GET

Set the headers

Generate a GET request

Get the data portion of the response

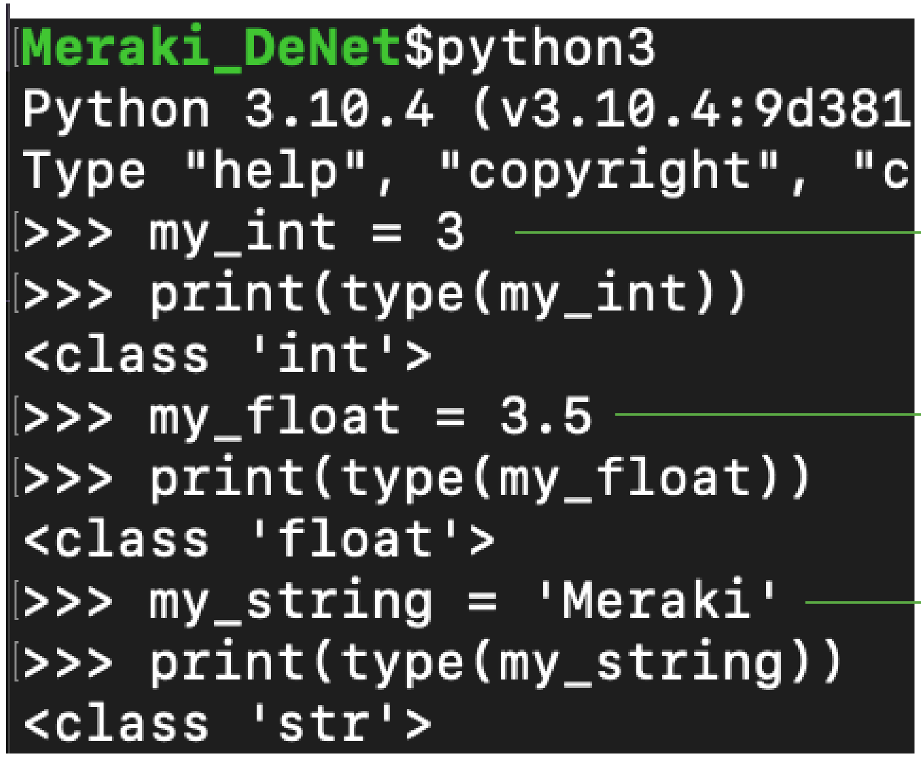
Key Python Concepts

- Data Types
- Variables
- if…else
- for and while loops

Data Types

Numeric	• int • float
Boolean	• True • False
Text	• String
Sequence	• List
Mapping	• Dictionary

Data Types

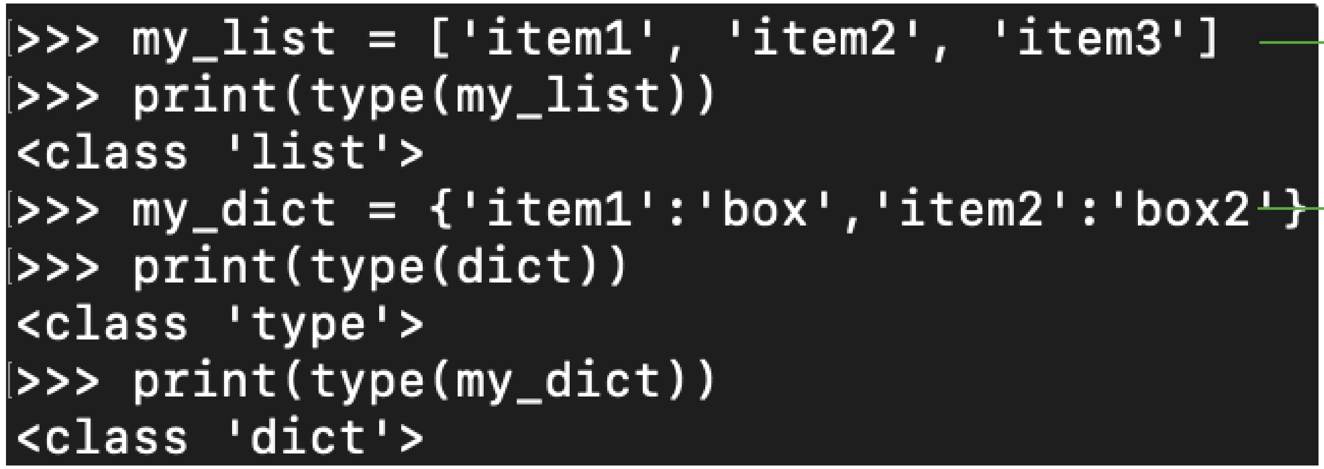


```
[Meraki_DeNet$python3
Python 3.10.4 (v3.10.4:9d381
Type "help", "copyright", "c
[>>> my_int = 3
[>>> print(type(my_int))
<class 'int'>
[>>> my_float = 3.5
[>>> print(type(my_float))
<class 'float'>
[>>> my_string = 'Meraki'
[>>> print(type(my_string))
<class 'str'>
```

int

float

string



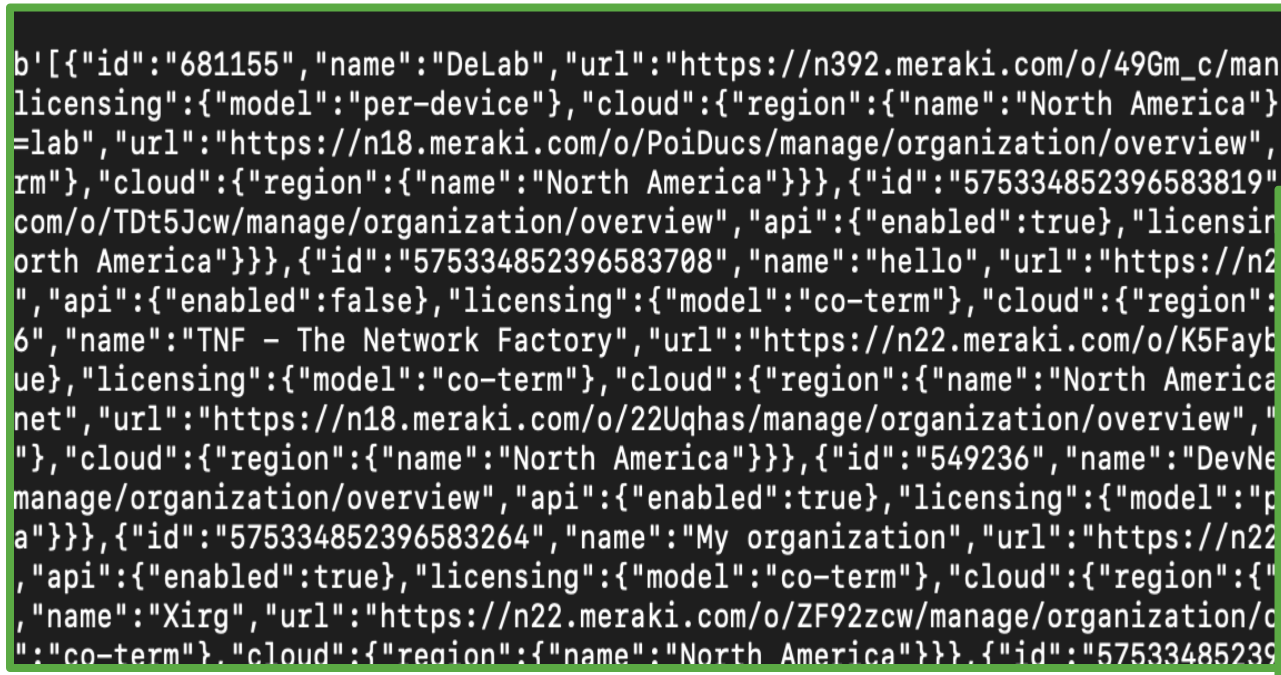
```
[>>> my_list = ['item1', 'item2', 'item3']
[>>> print(type(my_list))
<class 'list'>
[>>> my_dict = {'item1':'box','item2':'box2'}
[>>> print(type(dict))
<class 'type'>
[>>> print(type(my_dict))
<class 'dict'>
```

list

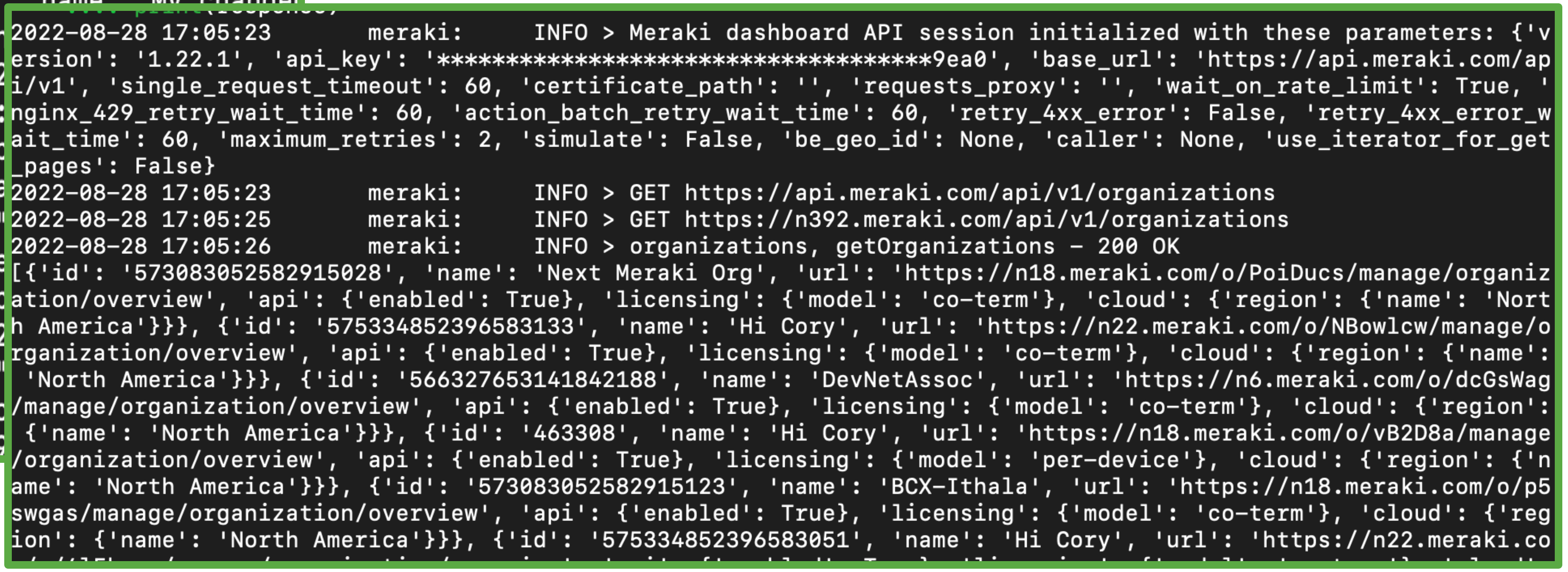
dictionary

Data Types

These are the outputs, notice any difference?

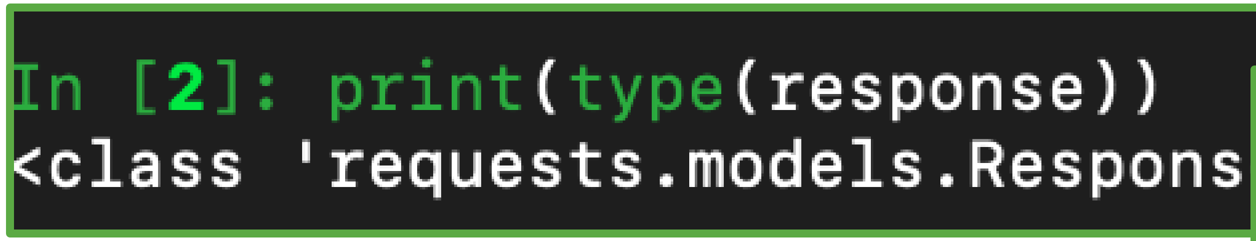


```
b'[{"id":"681155","name":"DeLab","url":"https://n392.meraki.com/o/49Gm_c/manage/organization/o
licensing":{"model":"per-device"},"cloud":{"region":{"name":"North America"}}},{"id":"57308305
=lab","url":"https://n18.meraki.com/o/PoiDucs/manage/organization/overview","api":{"enabled":fa
rm"},"cloud":{"region":{"name":"North America"}}},{"id":"575334852396583819","name":"My change
com/o/TDt5Jcw/manage/organization/overview","api":{"enabled":true},"licensin
orth America"}}},{"id":"575334852396583708","name":"hello","url":"https://n2
","api":{"enabled":false},"licensing":{"model":"co-term"},"cloud":{"region":
6","name":"TNF - The Network Factory","url":"https://n22.meraki.com/o/K5Fayb
ue},"licensing":{"model":"co-term"},"cloud":{"region":{"name":"North America
net","url":"https://n18.meraki.com/o/22Uqhas/manage/organization/overview","
"},"cloud":{"region":{"name":"North America"}}},{"id":"549236","name":"DevNe
manage/organization/overview","api":{"enabled":true},"licensing":{"model":"p
a"}}},{"id":"575334852396583264","name":"My organization","url":"https://n22
,"api":{"enabled":true},"licensing":{"model":"co-term"},"cloud":{"region":{"
,"name":"Xirg","url":"https://n22.meraki.com/o/ZF92zcw/manage/organization/o
":"co-term"},"cloud":{"region":{"name":"North America"}}},{"id":"57533485239
```

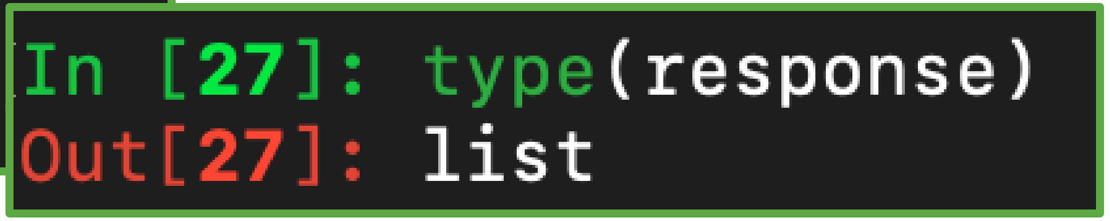


```
2022-08-28 17:05:23       meraki:     INFO > Meraki dashboard API session initialized with these parameters: {'v
ersion': '1.22.1', 'api_key': '************************************9ea0', 'base_url': 'https://api.meraki.com/ap
i/v1', 'single_request_timeout': 60, 'certificate_path': '', 'requests_proxy': '', 'wait_on_rate_limit': True, '
nginx_429_retry_wait_time': 60, 'action_batch_retry_wait_time': 60, 'retry_4xx_error': False, 'retry_4xx_error_w
ait_time': 60, 'maximum_retries': 2, 'simulate': False, 'be_geo_id': None, 'caller': None, 'use_iterator_for_get
_pages': False}
2022-08-28 17:05:23       meraki:     INFO > GET https://api.meraki.com/api/v1/organizations
2022-08-28 17:05:25       meraki:     INFO > GET https://n392.meraki.com/api/v1/organizations
2022-08-28 17:05:26       meraki:     INFO > organizations, getOrganizations - 200 OK
[{'id': '573083052582915028', 'name': 'Next Meraki Org', 'url': 'https://n18.meraki.com/o/PoiDucs/manage/organiz
ation/overview', 'api': {'enabled': True}, 'licensing': {'model': 'co-term'}, 'cloud': {'region': {'name': 'Nort
h America'}}}, {'id': '575334852396583133', 'name': 'Hi Cory', 'url': 'https://n22.meraki.com/o/NBowlcw/manage/o
rganization/overview', 'api': {'enabled': True}, 'licensing': {'model': 'co-term'}, 'cloud': {'region': {'name':
 'North America'}}}, {'id': '566327653141842188', 'name': 'DevNetAssoc', 'url': 'https://n6.meraki.com/o/dcGsWag
/manage/organization/overview', 'api': {'enabled': True}, 'licensing': {'model': 'co-term'}, 'cloud': {'region':
 {'name': 'North America'}}}, {'id': '463308', 'name': 'Hi Cory', 'url': 'https://n18.meraki.com/o/vB2D8a/manage
/organization/overview', 'api': {'enabled': True}, 'licensing': {'model': 'per-device'}, 'cloud': {'region': {'n
ame': 'North America'}}}, {'id': '573083052582915123', 'name': 'BCX-Ithala', 'url': 'https://n18.meraki.com/o/p5
swgas/manage/organization/overview', 'api': {'enabled': True}, 'licensing': {'model': 'co-term'}, 'cloud': {'reg
ion': {'name': 'North America'}}}, {'id': '575334852396583051', 'name': 'Hi Cory', 'url': 'https://n22.meraki.co
```

Let’s compare the type of the response variable.



```
In [2]: print(type(response))
<class 'requests.models.Respons
```

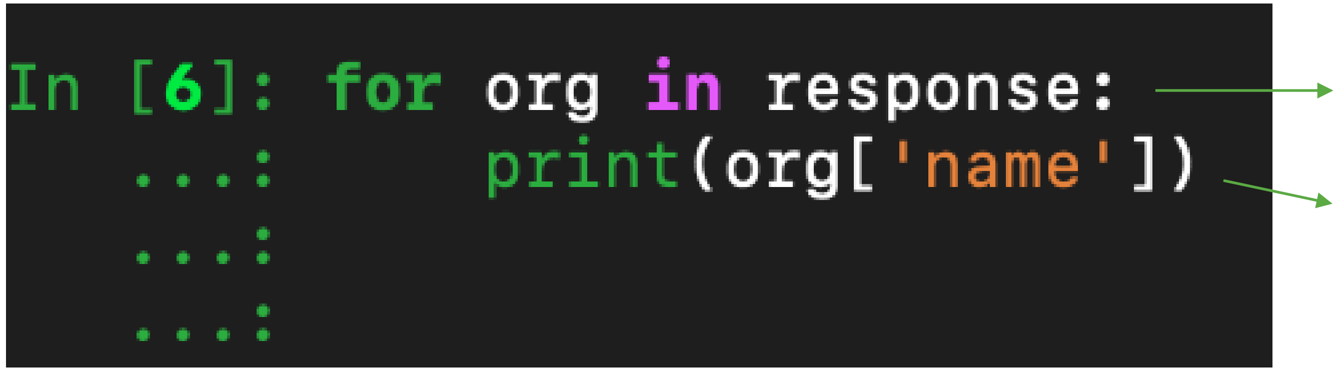


```
In [27]: type(response)
Out[27]: list
```

Data Types Demo

For and While Loops

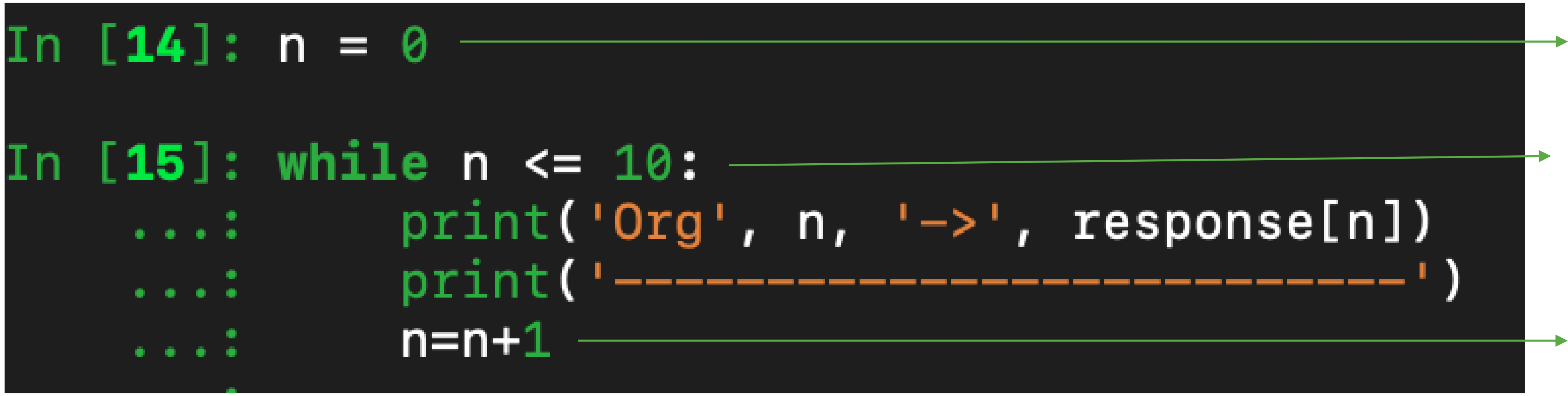
What would the output be for each of the following?



```
In [6]: for org in response:
   ...:     print(org['name'])
   ...:
   ...:
```

Take each item in “response” and store in org

Print the current value of org



```
In [14]: n = 0

In [15]: while n <= 10:
    ...:     print('Org', n, '->', response[n])
    ...:     print('--------------------------')
    ...:     n=n+1
```

Set counter to 0

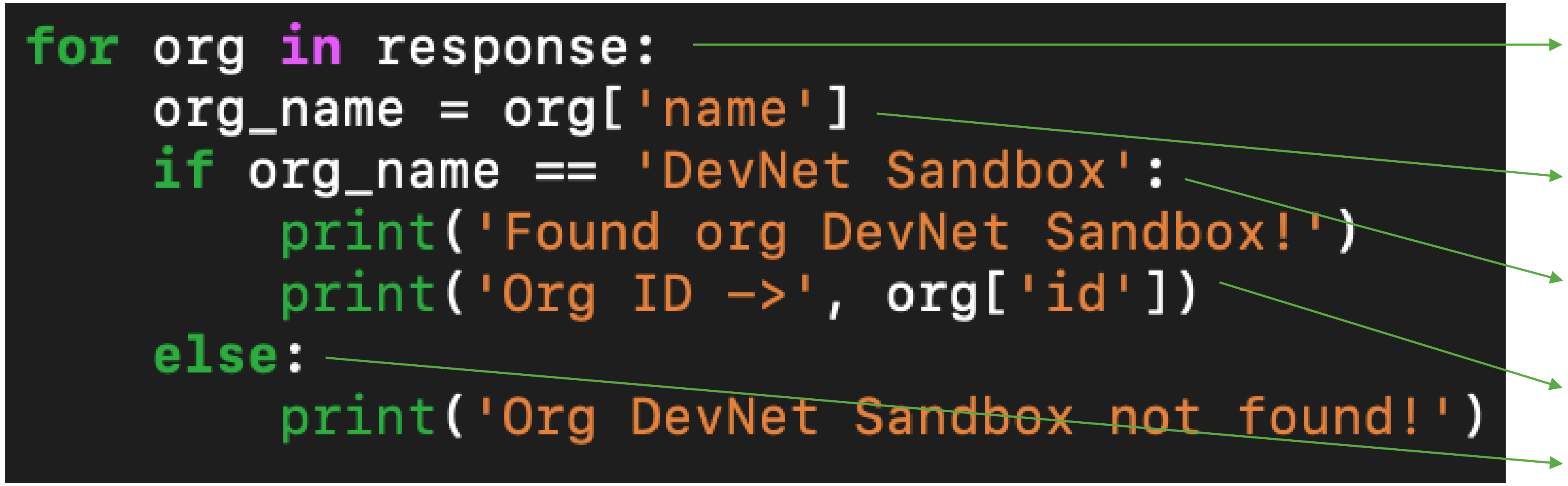
Set condition to stop the loop

Increment counter

*Hint: Try it out yourself!

For and While loops Demo

If...Else Statements



```
for org in response:
    org_name = org['name']
    if org_name == 'DevNet Sandbox':
        print('Found org DevNet Sandbox!')
        print('Org ID ->', org['id'])
    else:
        print('Org DevNet Sandbox not found!')
```

Iterate through each item in response and store in org

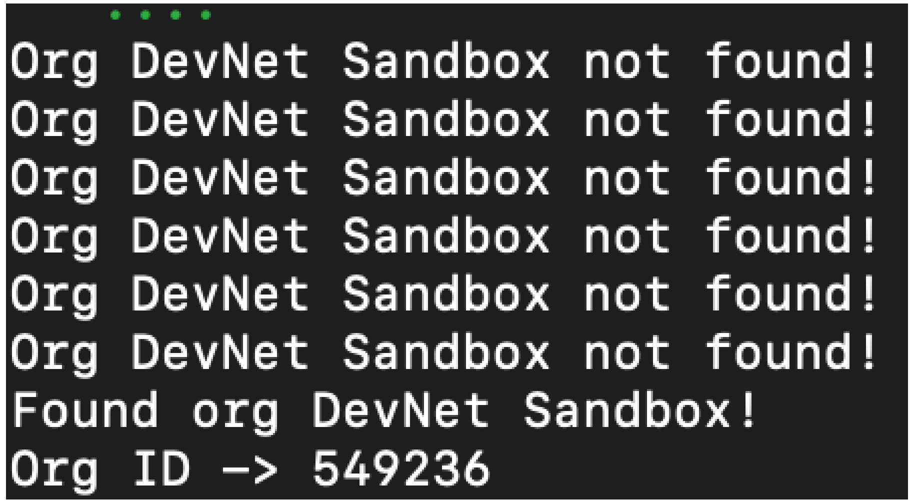
Store the org name in org_name

Check org name is DevNet Sandbox

If True then execute the print statements

If False, then execute this print

Can you guess the output?

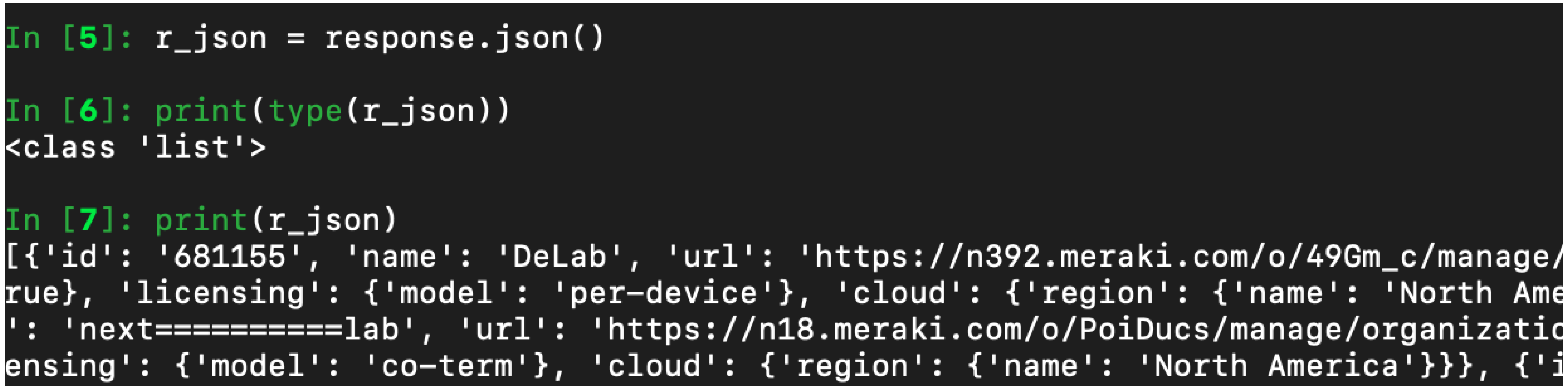


```
Org DevNet Sandbox not found!
Org DevNet Sandbox not found!
Org DevNet Sandbox not found!
Org DevNet Sandbox not found!
Org DevNet Sandbox not found!
Org DevNet Sandbox not found!
Found org DevNet Sandbox!
Org ID -> 549236
```

- Most common use case for if...else would be to check the response code.
- For example, if response is 2xx, then continue else if response code is 4xx then generate error

JSON with Python

Let’s convert the *requests* response data to something python can process.

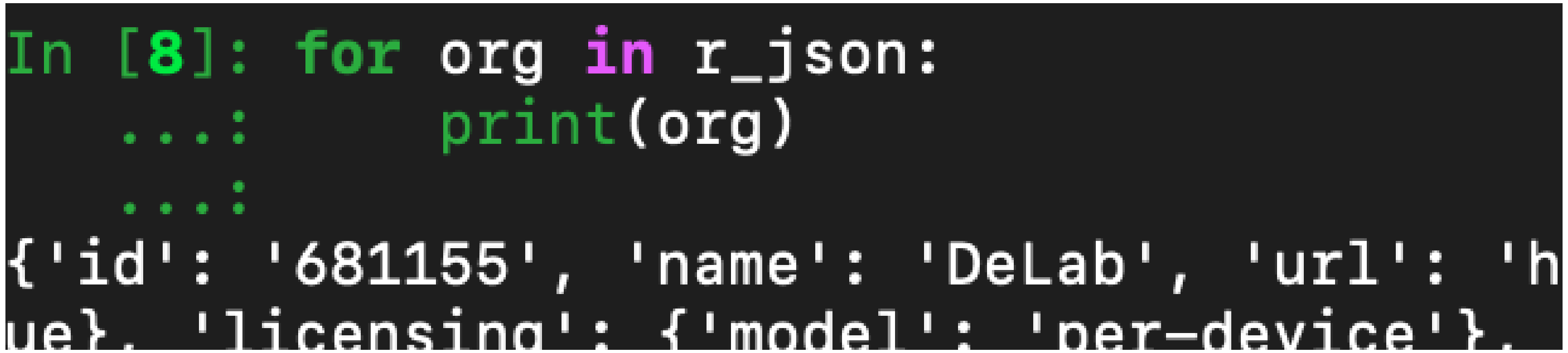


```
In [5]: r_json = response.json()

In [6]: print(type(r_json))
<class 'list'>

In [7]: print(r_json)
[{'id': '681155', 'name': 'DeLab', 'url': 'https://n392.meraki.com/o/49Gm_c/manage/
rue}, 'licensing': {'model': 'per-device'}, 'cloud': {'region': {'name': 'North Ame
': 'next==========lab', 'url': 'https://n18.meraki.com/o/PoiDucs/manage/organizatio
ensing': {'model': 'co-term'}, 'cloud': {'region': {'name': 'North America'}}}, {'i
```

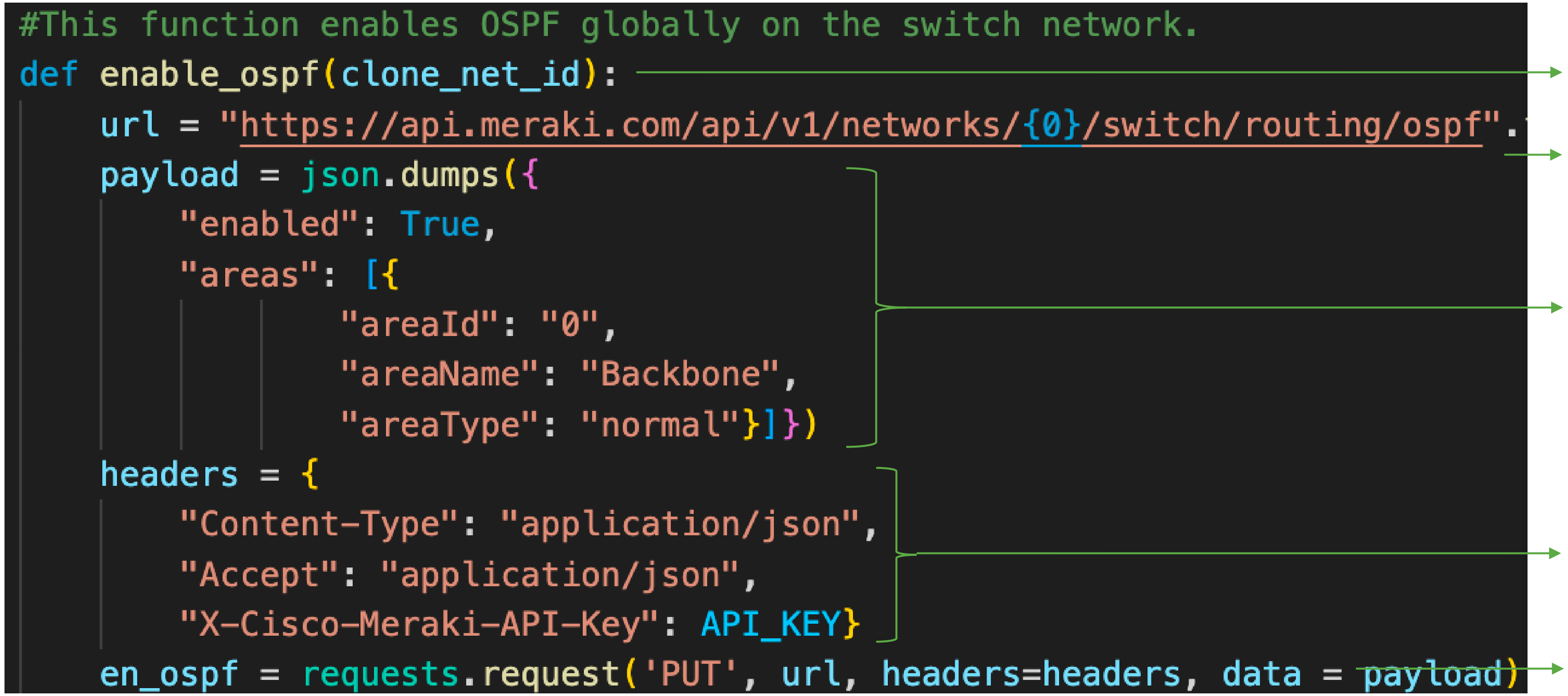
Now that it is a *list* we can use *for* loop to parse the data!



```
In [8]: for org in r_json:
   ...:     print(org)
   ...:
{'id': '681155', 'name': 'DeLab', 'url': 'h
ue}, 'licensing': {'model': 'per-device'},
```

Python Functions

Have a look at the Python function below…



```
#This function enables OSPF globally on the switch network.
def enable_ospf(clone_net_id):
    url = "https://api.meraki.com/api/v1/networks/{0}/switch/routing/ospf".
    payload = json.dumps({
        "enabled": True,
        "areas": [{
                "areaId": "0",
                "areaName": "Backbone",
                "areaType": "normal"}]})
    headers = {
        "Content-Type": "application/json",
        "Accept": "application/json",
        "X-Cisco-Meraki-API-Key": API_KEY}
    en_ospf = requests.request('PUT', url, headers=headers, data = payload)
```

Define a function

Request URL

Data in JSON sent to the dashboard

Headers sent in request

PUT request

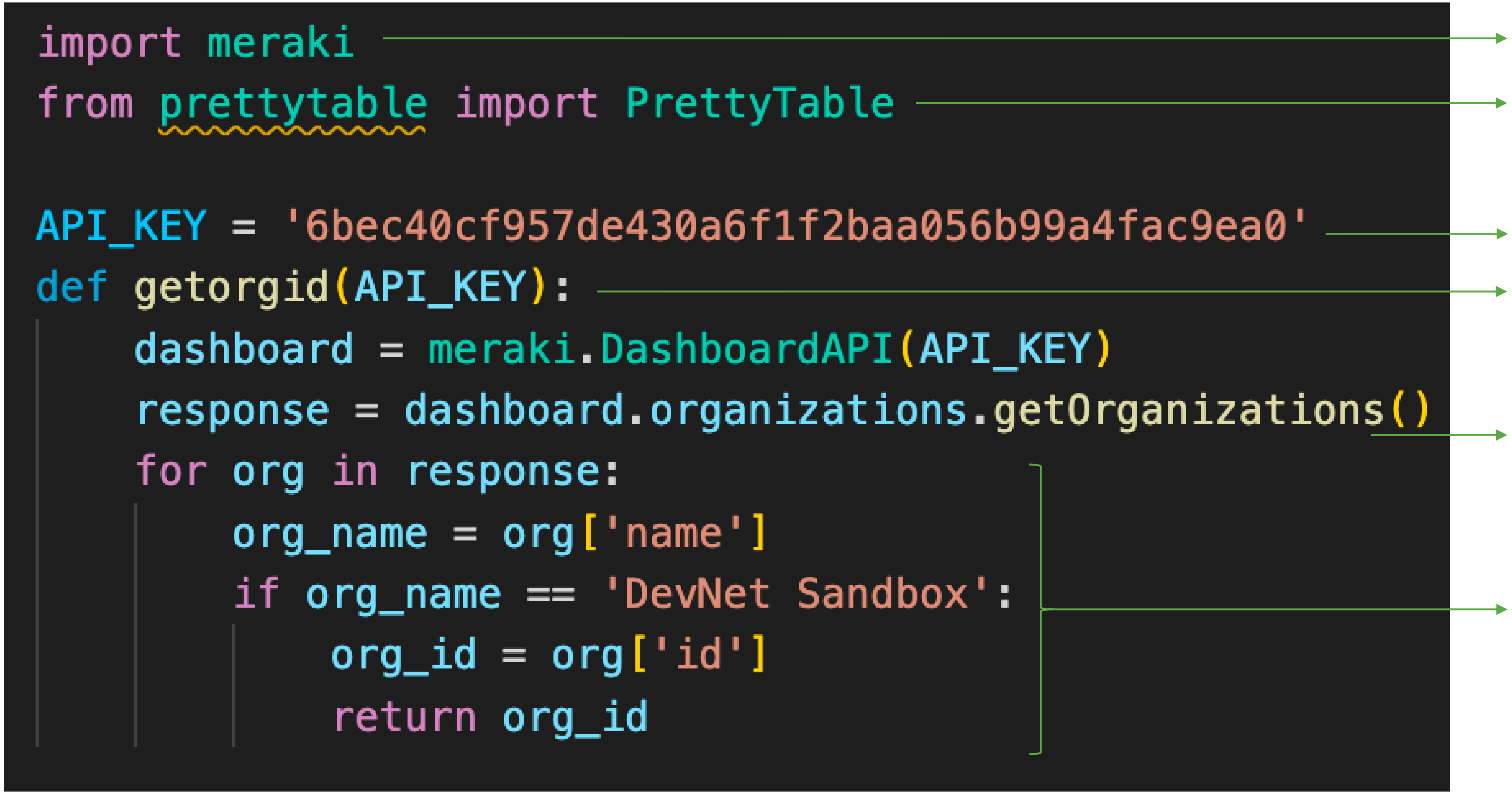
Now I can reuse this function whenever I need to enable OSPF!

Slido Quiz

Bringing it all together
Demo

Bringing It All Together

Here is the first half of the code.



```
import meraki
from prettytable import PrettyTable

API_KEY = '6bec40cf957de430a6f1f2baa056b99a4fac9ea0'
def getorgid(API_KEY):
    dashboard = meraki.DashboardAPI(API_KEY)
    response = dashboard.organizations.getOrganizations()
    for org in response:
        org_name = org['name']
        if org_name == 'DevNet Sandbox':
            org_id = org['id']
            return org_id
```

Import the Meraki library

Import prettytable

Set the API key

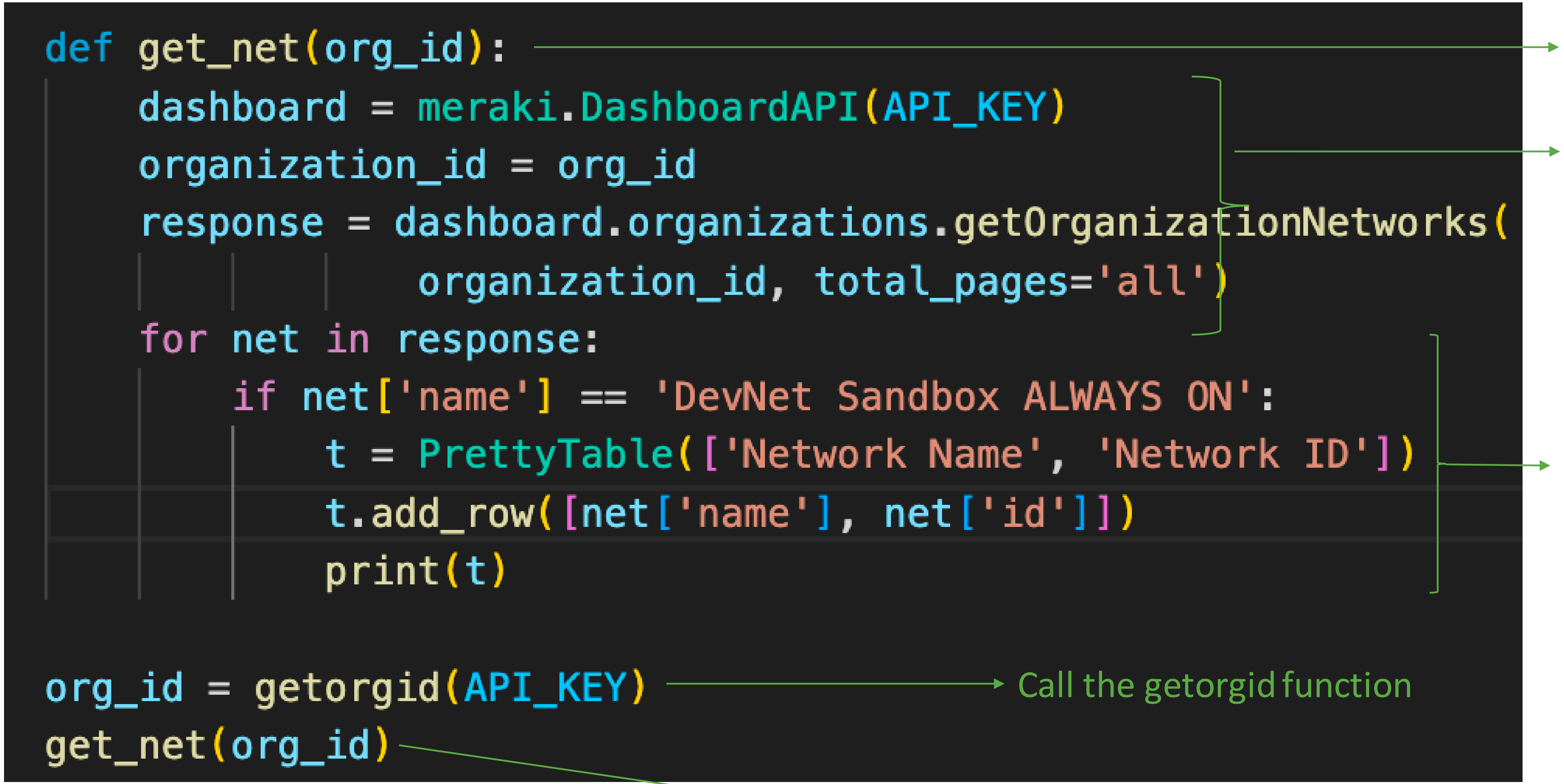
Define a function getorgid

Run the GET request

Find the org DevNet Sandbox and return the org id

Bringing It All Together

Here is the second half of the code.



```
def get_net(org_id):
    dashboard = meraki.DashboardAPI(API_KEY)
    organization_id = org_id
    response = dashboard.organizations.getOrganizationNetworks(
                organization_id, total_pages='all')
    for net in response:
        if net['name'] == 'DevNet Sandbox ALWAYS ON':
            t = PrettyTable(['Network Name', 'Network ID'])
            t.add_row([net['name'], net['id']])
            print(t)

org_id = getorgid(API_KEY)
get_net(org_id)
```

Define a function get_net

GET request for networks

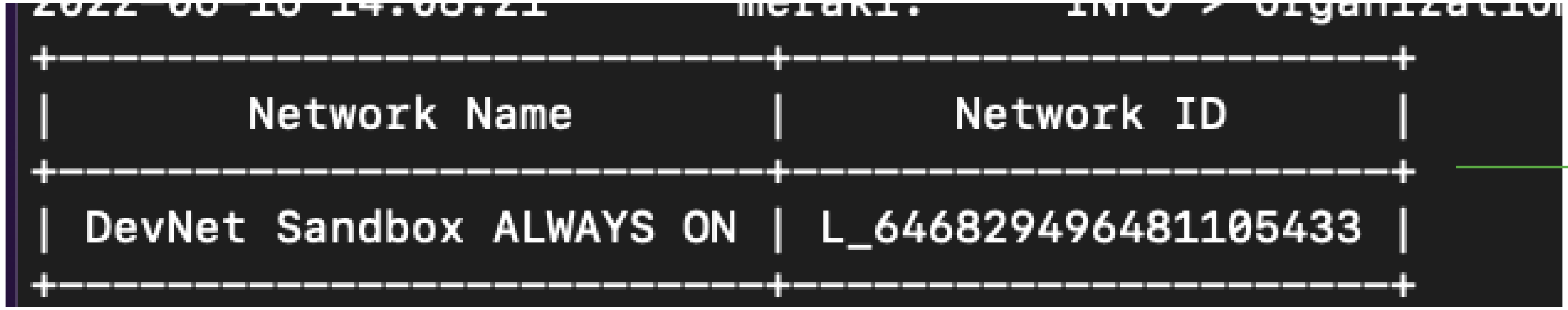
Check network name and then print its name and id

Call the getorgid function

Call the get_net function with org id

Bringing It All Together: Output

Here is the output.



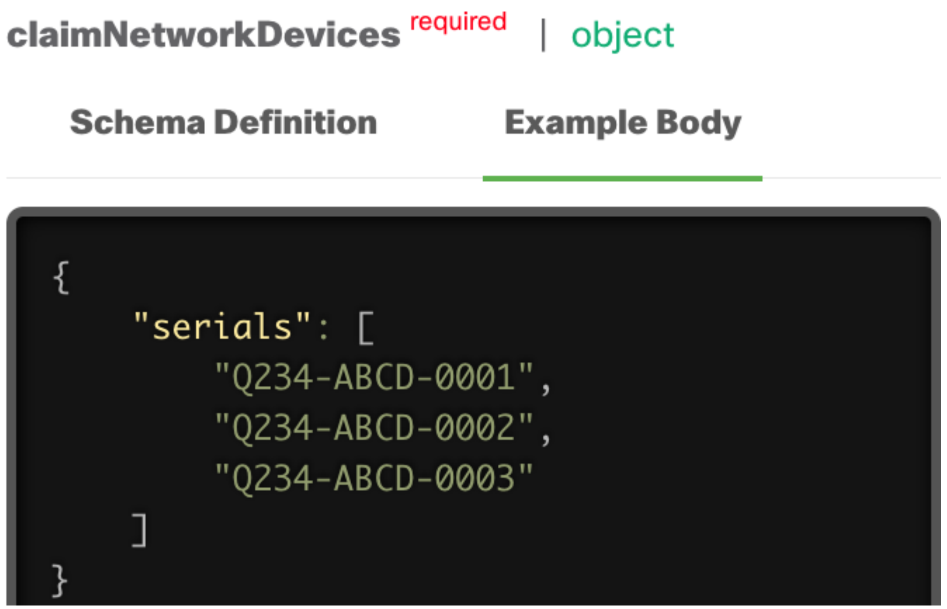
```
2022-06-16 14:08:21       meraki:     INFO > Organization
+--------------------------+----------------------+
|       Network Name       |      Network ID      |
+--------------------------+----------------------+
| DevNet Sandbox ALWAYS ON | L_646829496481105433 |
+--------------------------+----------------------+
```

Output printed using prettytable

Using the Network ID

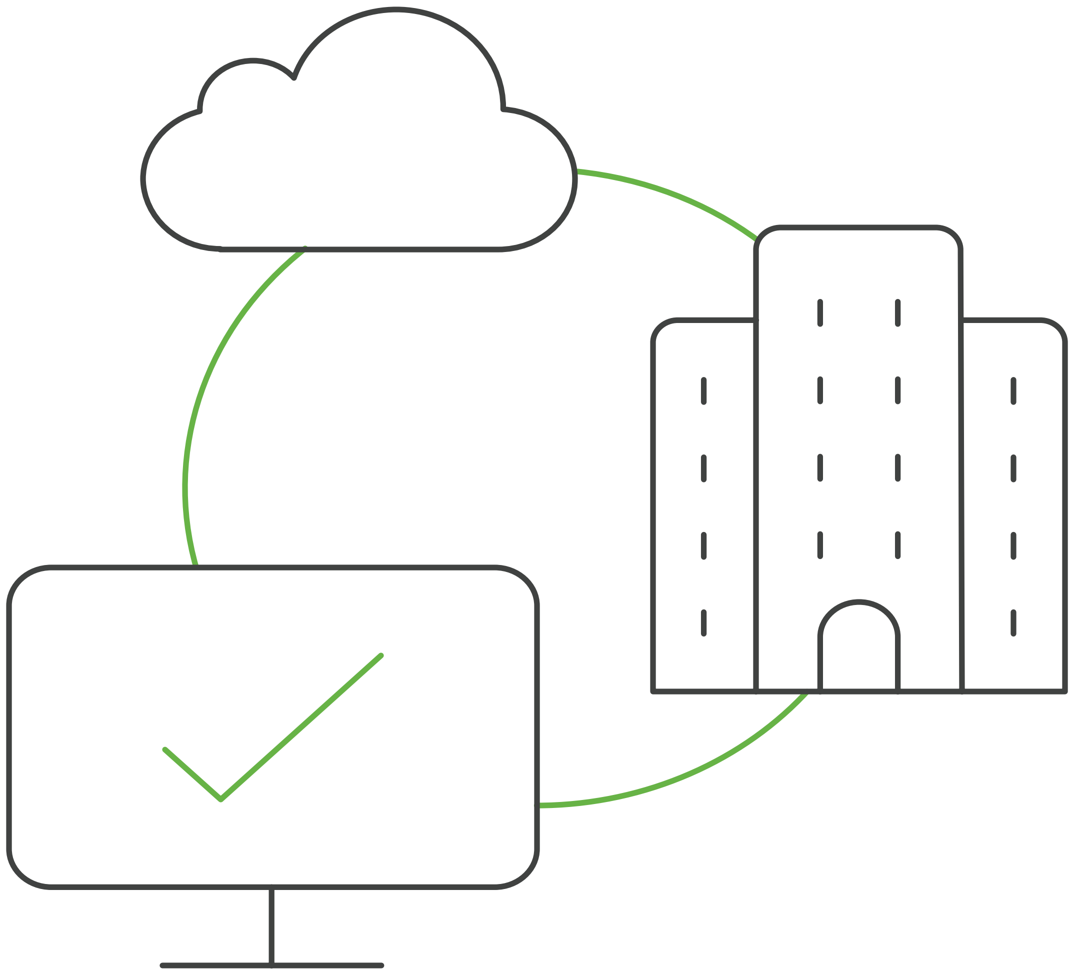
Now that we have a network ID, what can we do with it?

- One of the common tasks is to claim devices into a network
- But we need a list of serial numbers for that, as can be seen from the API docs.



claimNetworkDevices required | object

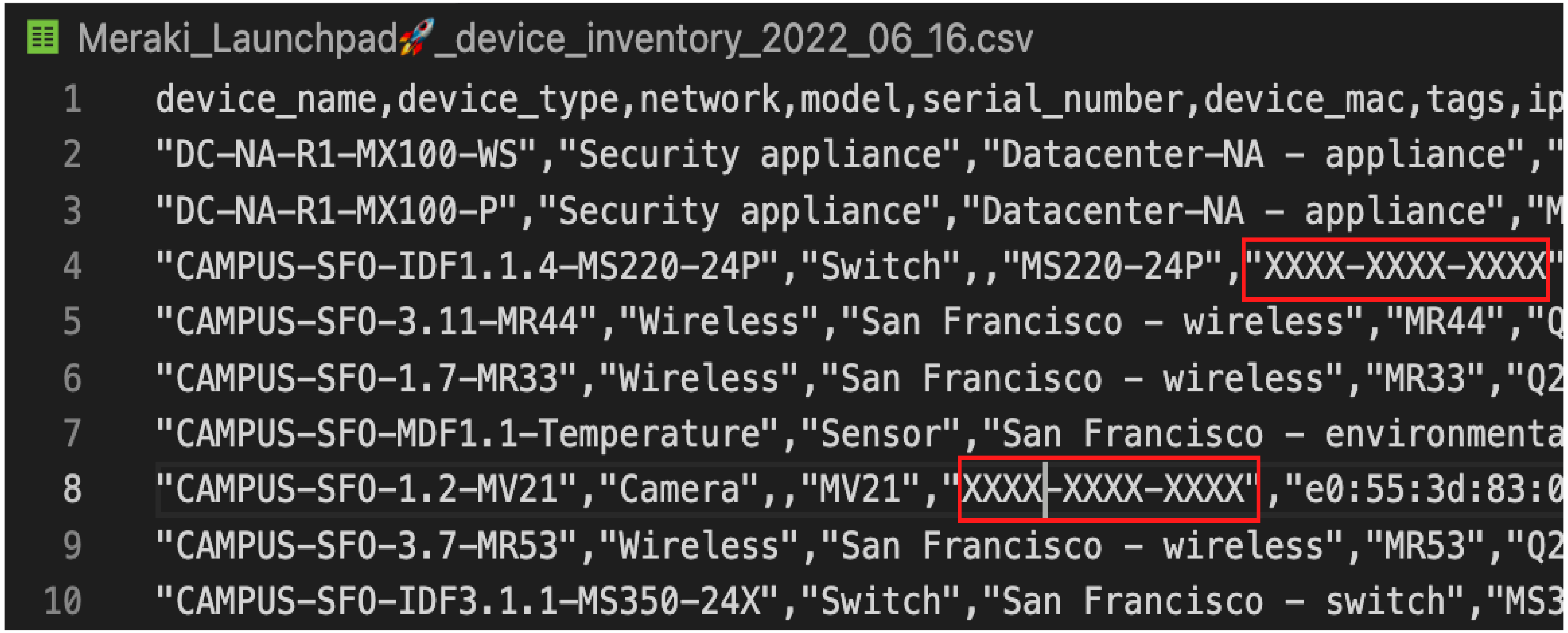
```
{
    "serials": [
        "Q234-ABCD-0001",
        "Q234-ABCD-0002",
        "Q234-ABCD-0003"
    ]
}
```



Reading from a CSV File

Reading from a CSV File

This is a screenshot of the inventory CSV file downloaded from the dashboard.



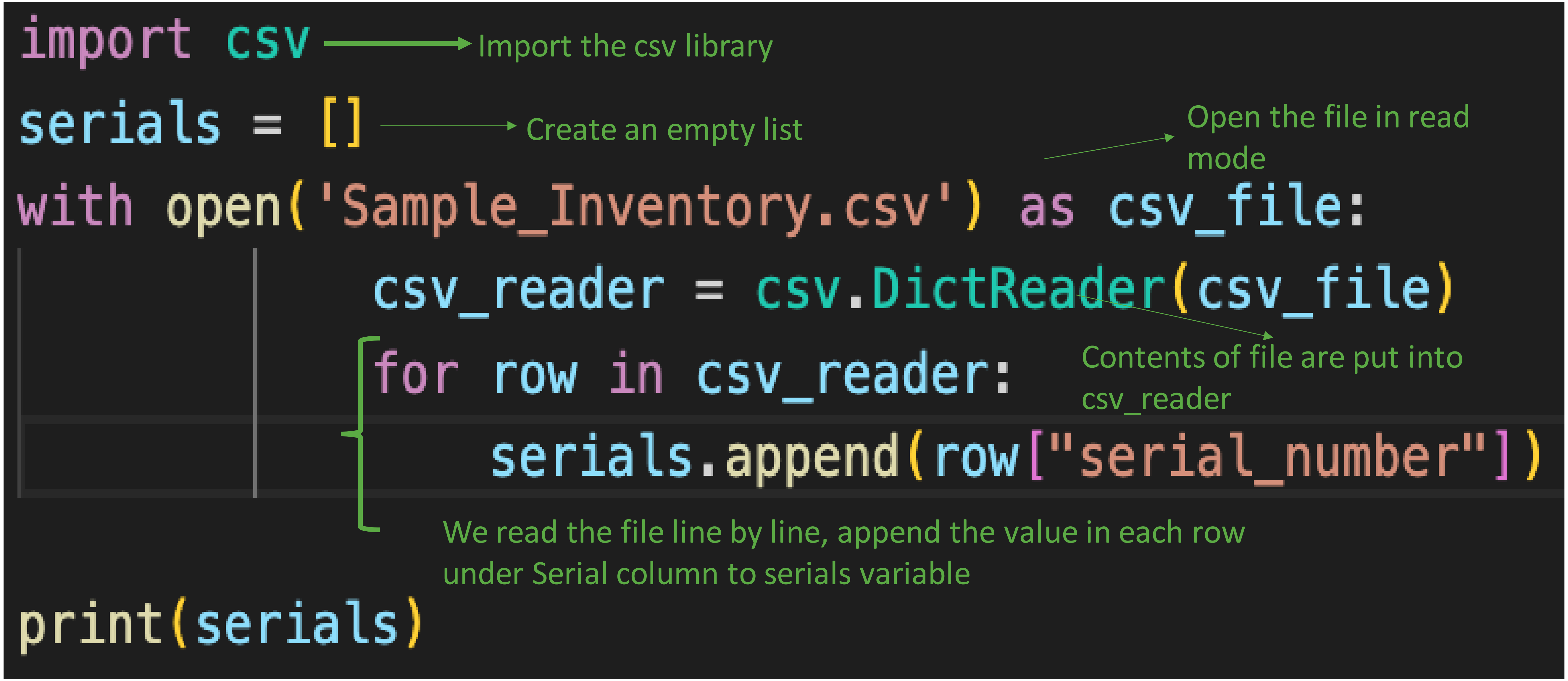
Meraki_Launchpad🚀_device_inventory_2022_06_16.csv

```
device_name,device_type,network,model,serial_number,device_mac,tags,ip
"DC-NA-R1-MX100-WS","Security appliance","Datacenter-NA - appliance","
"DC-NA-R1-MX100-P","Security appliance","Datacenter-NA - appliance","M
"CAMPUS-SFO-IDF1.1.4-MS220-24P","Switch",,"MS220-24P","XXXX-XXXX-XXXX"
"CAMPUS-SFO-3.11-MR44","Wireless","San Francisco - wireless","MR44","Q
"CAMPUS-SFO-1.7-MR33","Wireless","San Francisco - wireless","MR33","Q2
"CAMPUS-SFO-MDF1.1-Temperature","Sensor","San Francisco - environmenta
"CAMPUS-SFO-1.2-MV21","Camera",,"MV21","XXXX-XXXX-XXXX","e0:55:3d:83:0
"CAMPUS-SFO-3.7-MR53","Wireless","San Francisco - wireless","MR53","Q2
"CAMPUS-SFO-IDF3.1.1-MS350-24X","Switch","San Francisco - switch","MS3
```

How can we get just the serial number from this file to add into a network?

We can use the CSV Python library.

Reading from a CSV file (2)



```
import csv
serials = []
with open('Sample_Inventory.csv') as csv_file:
        csv_reader = csv.DictReader(csv_file)
        for row in csv_reader:
            serials.append(row["serial_number"])
print(serials)
```

Import the csv library

Create an empty list

Open the file in read mode

Contents of file are put into csv_reader

We read the file line by line, append the value in each row under Serial column to serials variable

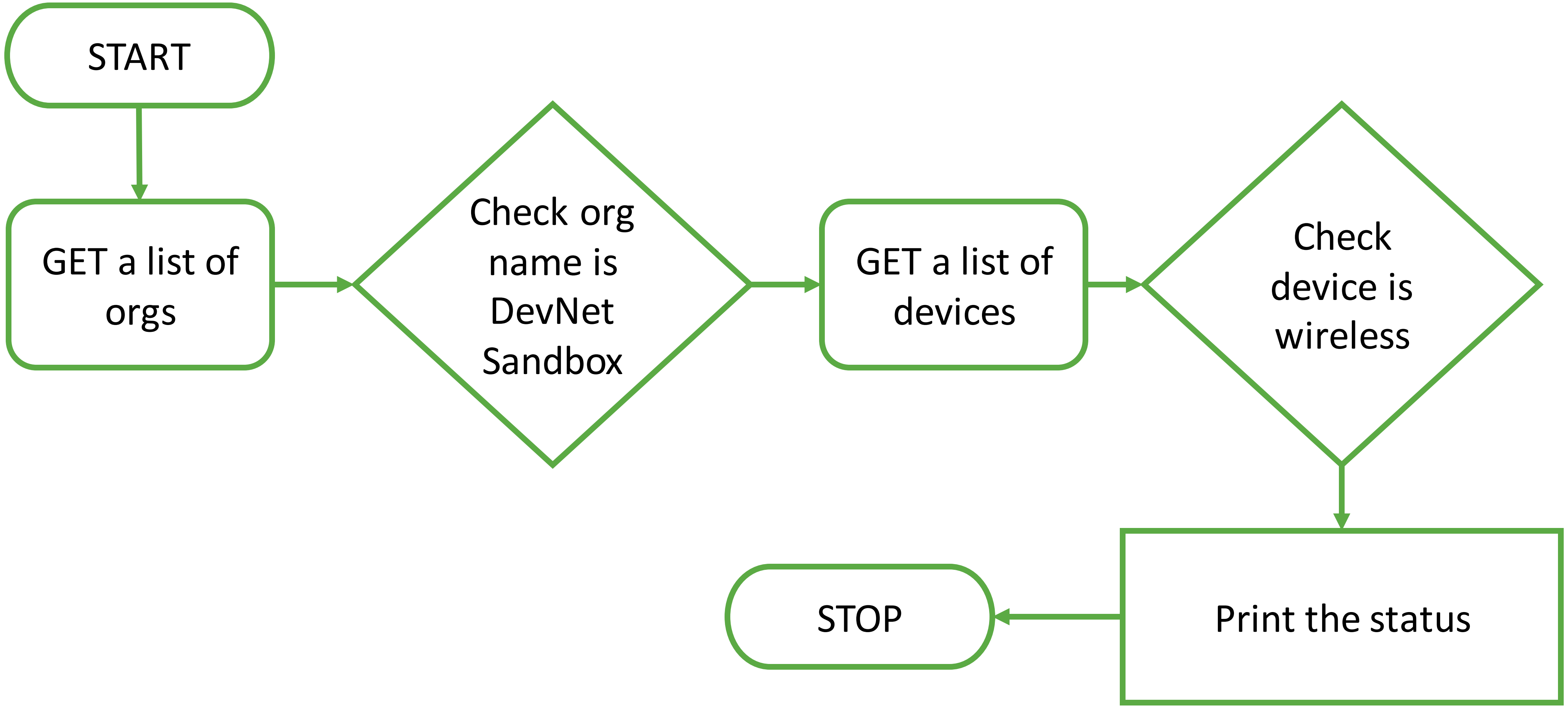
Reading from a CSV file
Demo

Slido Quiz

Final Exercise

Final Exercise

List the status of wireless AP's in DevNet Sandbox organization



Next Steps

https://developer.cisco.com/meraki/

Thank you!

Meraki