



DEVNET

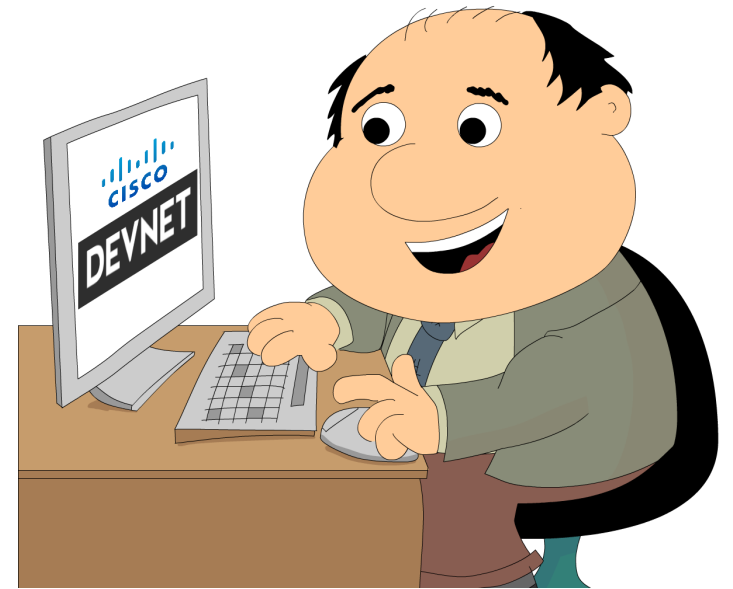
REST APIs Part 2: Making REST API Calls with Postman

A Network Programmability Basics Presentation

Hank Preston, ccie 38336
Developer Evangelist
@hfpreston 

Network Programmability Basics Modules

- Introduction: How to be a Network Engineer in a Programmable Age
- **Programming Fundamentals**
- Network Device APIs
- Network Controllers
- Application Hosting and the Network
- NetDevOps



Network Programmability Basics: The Lessons

Module: Programming Fundamentals

- Data Formats: Understanding and using JSON, XML and YAML
- APIs are Everywhere... but what are they?
- REST APIs Part 1: HTTP is for more than Web Browsing
- **REST APIs Part 2: Making REST API Calls with Postman**
- Python Part 1: Python Language and Script Basics
- Python Part 2: Working with Libraries and Virtual Environments
- Python Part 3: Useful Python Libraries for Network Engineers

Code and Develop Along

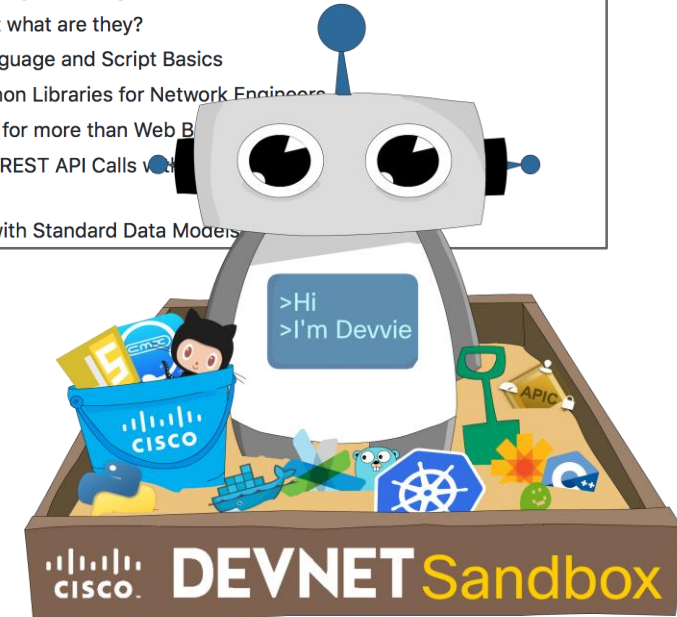
- Get the Code!
 - github.com/CiscoDevNet/netprog_basics
- Setup Lab Prerequisites
 - Each lab includes a README with details
- Access to Infrastructure
 - [DevNet Sandbox](#)
 - Specifics in lab README

Network Programmability Basics

Code, Examples, and Resources for the Network Programmability Basics Video Course

Table of Contents

- **Programming Fundamentals**
 - Data Formats: Understanding and using JSON, XML and YAML
 - APIs are Everywhere... but what are they?
 - Python Part 1: Python Language and Script Basics
 - Python Part 2: Useful Python Libraries for Network Engineers
 - REST APIs Part 1: HTTP is for more than Web Browsers
 - REST APIs Part 2: Making REST API Calls with Python
- **Network Device APIs**
 - Getting the "YANG" of it with Standard Data Models



Topics to Cover

- Why Postman and How to Get it?
- Sending an API Request
- API Collections
- Using Environments
- Postman to Code!

Why Postman and How to
Get it?

Postman: Powerful but Simple REST API Client

- Quickly test APIs in GUI
- Save APIs into Collections for reuse
- Manage multiple environments
- Auto generate code from API calls
- Standalone Application or Chrome Plugin



<https://www.getpostman.com>

Sending an API Request

The image shows a screenshot of the Postman REST client interface with several annotations pointing to specific features:

- URI**: Points to the URL bar at the top.
- Set Method**: Points to the HTTP method dropdown menu (currently set to GET).
- Manage request authorization, headers, and data (body)**: Points to the tabs for Authorization, Headers (1), and Body.
- Send Request**: Points to the Send button.
- Easily input Headers with auto-completion**: Points to the Headers table.
- Response Status Code**: Points to the status bar showing "Status: 200 OK".
- View Response details**: Points to the response body area.
- Response Body**: Points to the JSON response content.

The interface shows a request to `https://api.icndb.com/jokes/random` with the following headers:

Key	Value	Description
Accept	application/json	

The response status is **200 OK** and the response body is:

```
{
  "type": "success",
  "value": {
    "id": 403,
    "joke": "Chuck Norris crossed the road. No one has ever dared question his motives.",
    "categories": []
  }
}
```

Constructing a POST Request

- Choose method
- Enter URI
- Configure headers and authentication
- Provide data
- Send and verify status

Authorization

Headers (1)

Body

Pre-request Script

Tests

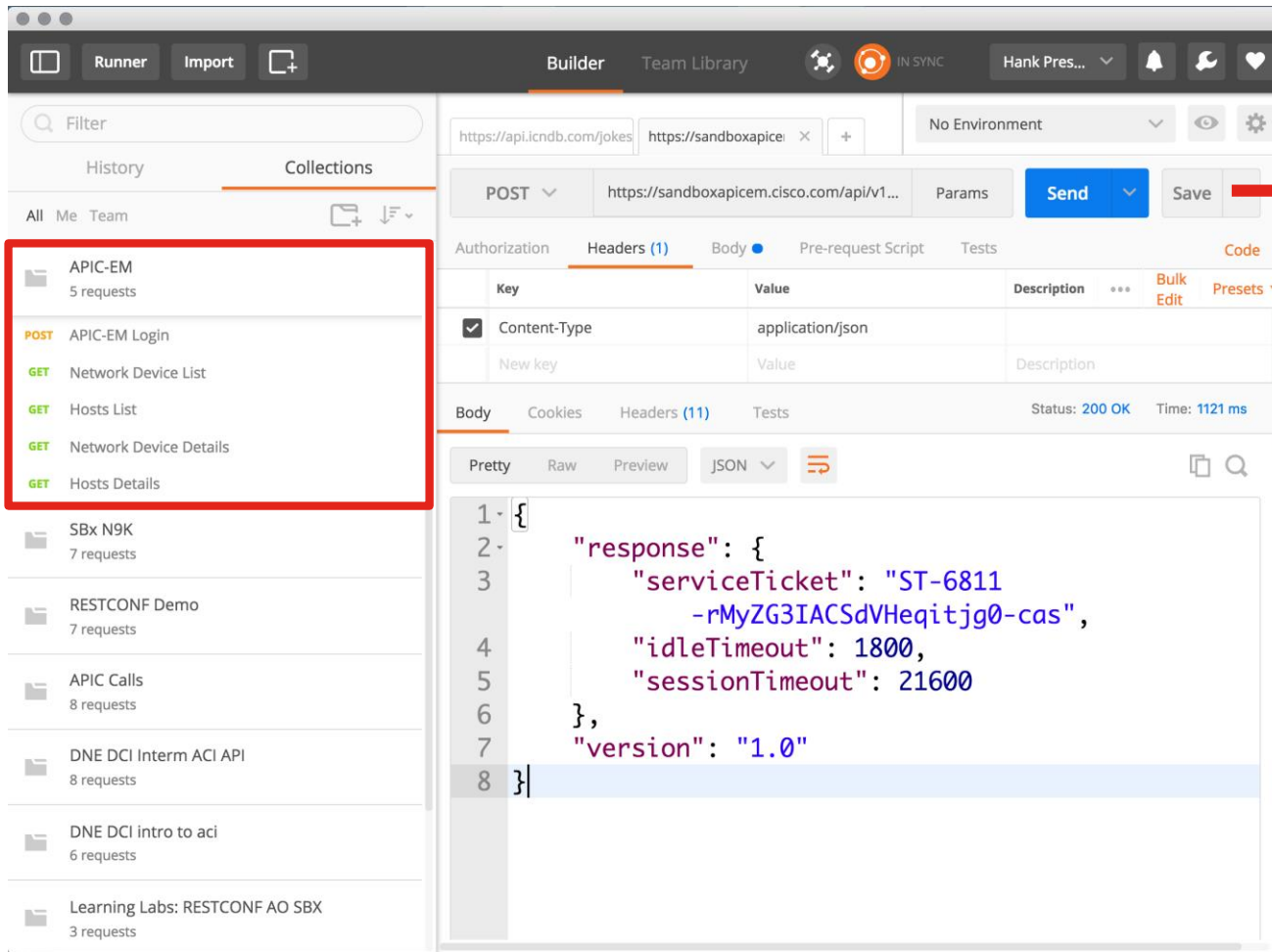
	Key	Value
<input checked="" type="checkbox"/>	Content-Type	application/json

The screenshot shows a REST client interface with a POST request to `https://sandboxapicem.cisco.com/api/v1/ticket`. The request body is a JSON object with `username: "devnetuser"` and `password: "Cisco123!"`. The response status is 200 OK, and the response body is a JSON object containing a `serviceTicket`, `idleTimeout`, `sessionTimeout`, and `version`.

```
1 {  
2   "username": "devnetuser",  
3   "password": "Cisco123!"  
4 }  
  
1 {  
2   "response": {  
3     "serviceTicket": "ST-6811-rMyZG3IACsdVHeqitjg0-cas",  
4     "idleTimeout": 1800,  
5     "sessionTimeout": 21600  
6   },  
7   "version": "1.0"  
8 }
```

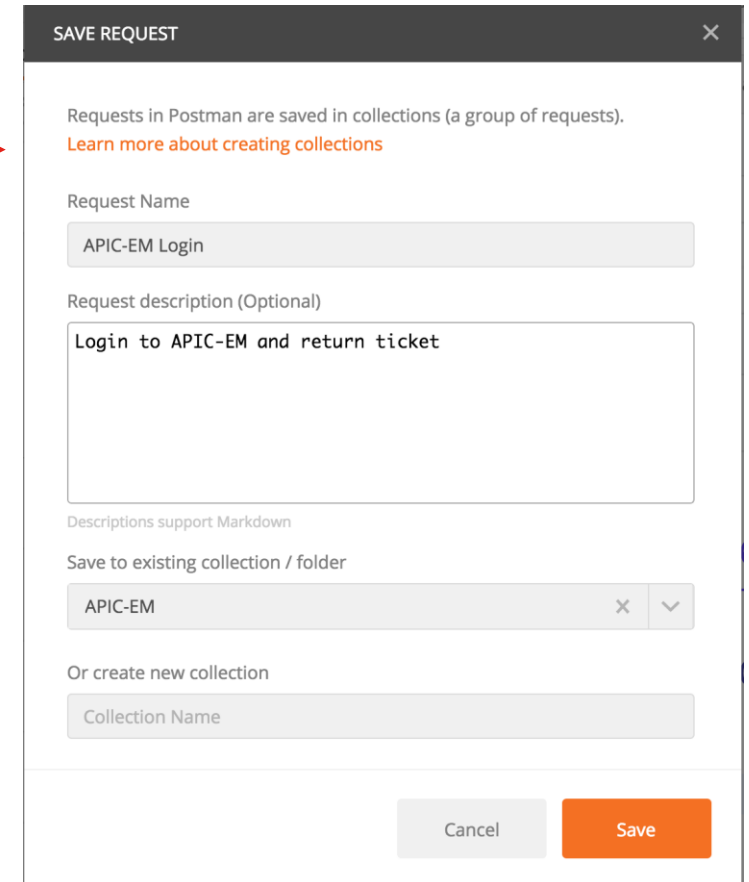
API Collections

Save and Organize API Calls into Collections



The screenshot shows the Postman interface. On the left, the 'Collections' sidebar is visible, with a red box highlighting the 'APIC-EM' collection, which contains 5 requests. The main panel shows a POST request to 'https://sandboxapicem.cisco.com/api/v1...' with a 'Content-Type' header of 'application/json'. The response is a JSON object with a 'serviceTicket' and a 'version'.

```
1 {  
2   "response": {  
3     "serviceTicket": "ST-6811  
4       -rMyZG3IACsDVHeqitjg0-cas",  
5     "idleTimeout": 1800,  
6     "sessionTimeout": 21600  
7   },  
8   "version": "1.0"  
}
```



SAVE REQUEST

Requests in Postman are saved in collections (a group of requests).
[Learn more about creating collections](#)

Request Name
APIC-EM Login

Request description (Optional)
Login to APIC-EM and return ticket

Descriptions support Markdown

Save to existing collection / folder
APIC-EM

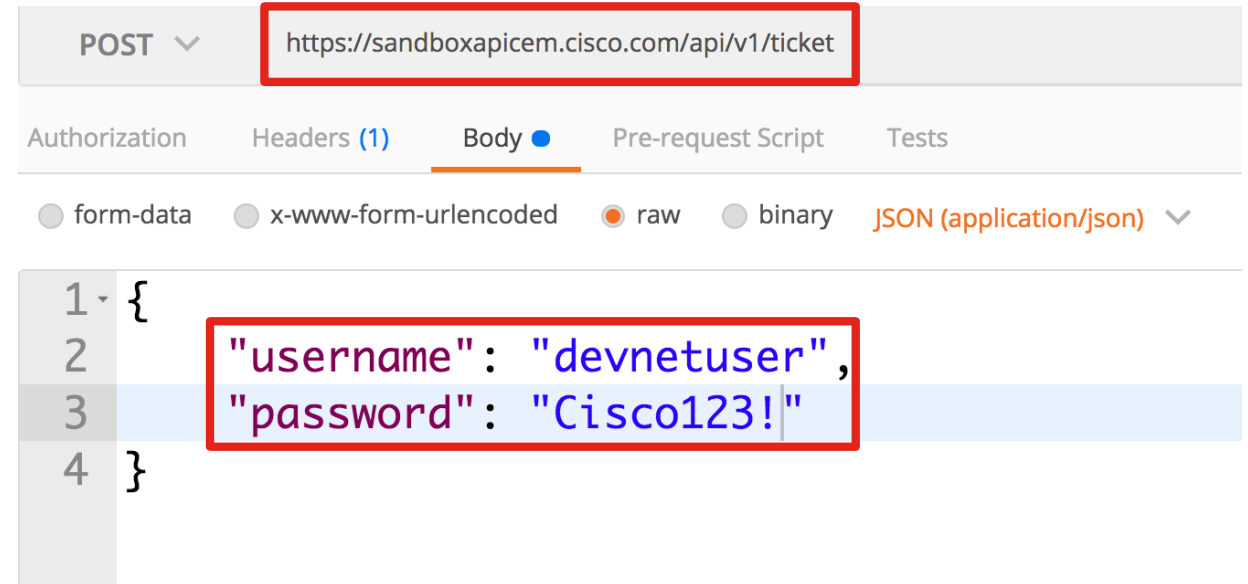
Or create new collection
Collection Name

Cancel Save

Using Environments

Variables Make Requests Reusable and Flexible

- Never good to hardcode details
- What if you want to connect to different host?
- What if credentials change?



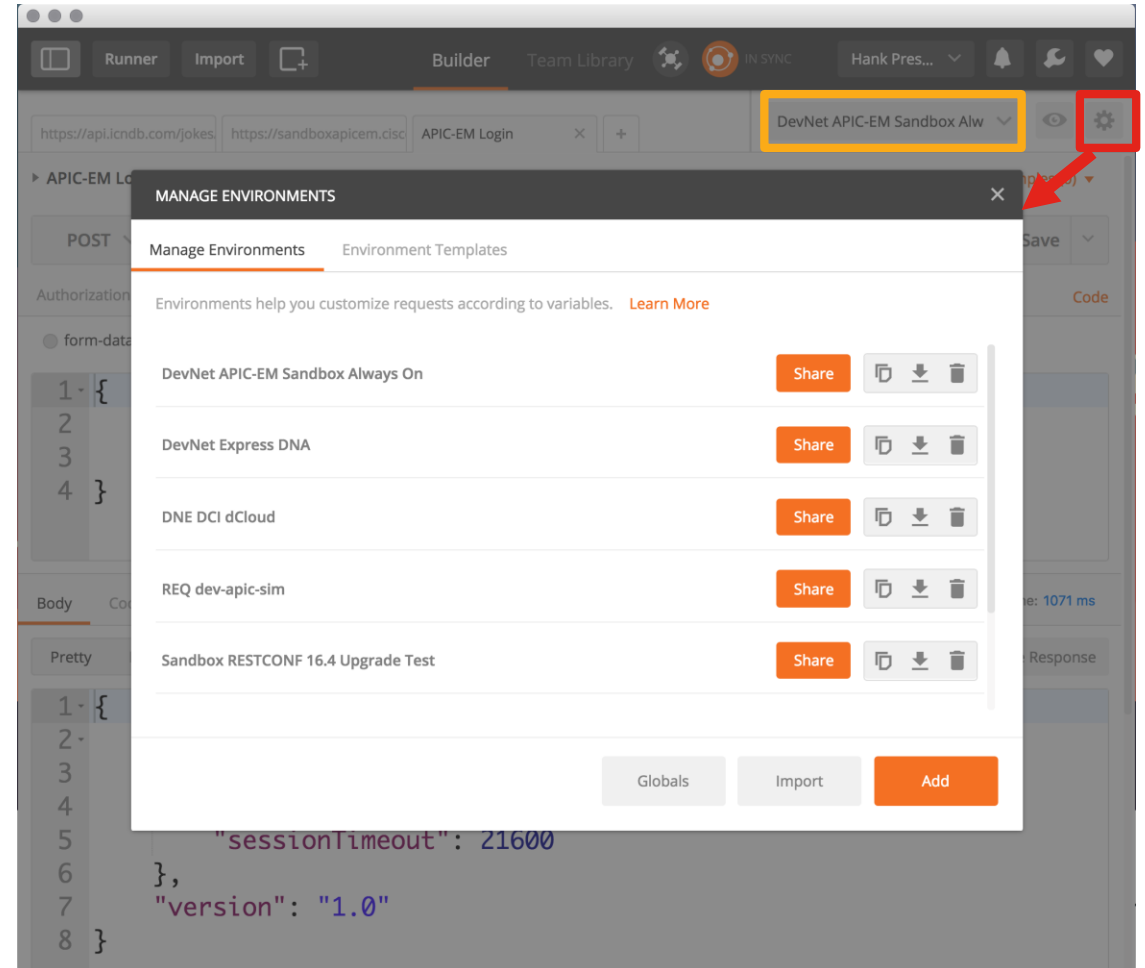
Variables Make Requests Reusable and Flexible

- Variables References
 - {{apic}}
 - {{username}}
 - {{password}}



Managing Environments

- Create any number of environments needed
- Change between environments with drop down list



Managing Environments

- Add as many variables as needed
- Reference anywhere with `{{variable name}}` syntax

POST ▼ `https://{{apic}}/api/v1/ticket`

Authorization Headers (1) Body Pre-request Script Tests

form-data x-www-form-urlencoded raw binary JSON (application/json)

```
1 {  
2   "username": "{{username}}",  
3   "password": "{{password}}"  
4 }
```

MANAGE ENVIRONMENTS

Manage Environments Environment Templates

Edit Environment

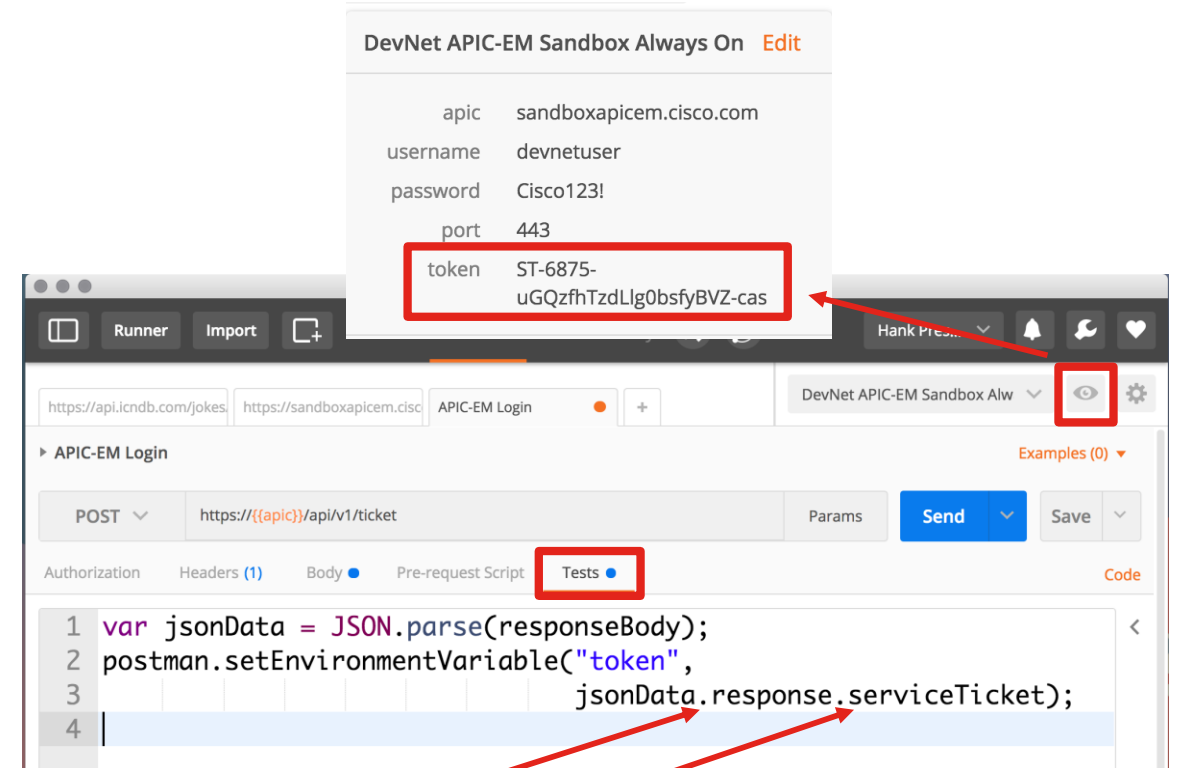
DevNet APIC-EM Sandbox Always On

	Key	Value	Bulk Edit
<input checked="" type="checkbox"/>	apic	sandboxapicem.cisco.com	
<input checked="" type="checkbox"/>	username	devnetuser	
<input checked="" type="checkbox"/>	password	Cisco123!	
<input checked="" type="checkbox"/>	port	443	
	New key	Value	

Cancel Update

“Tests” Enable Dynamic Environment Variables

- Each API Request offers both pre and post actions
 - Pre -> Pre-request Script
 - Post -> Tests
- Written in JavaScript



Response Body

```
{
  "response": {
    "serviceTicket": "ST-6862-5DmKF5FrP0S4bSjy9rDM-cas",
    "idleTimeout": 1800,
    "sessionTimeout": 21600
  },
  "version": "1.0"
}
```

Red arrows point from the 'jsonData.response.serviceTicket' in the test script to the 'serviceTicket' field in the response body.

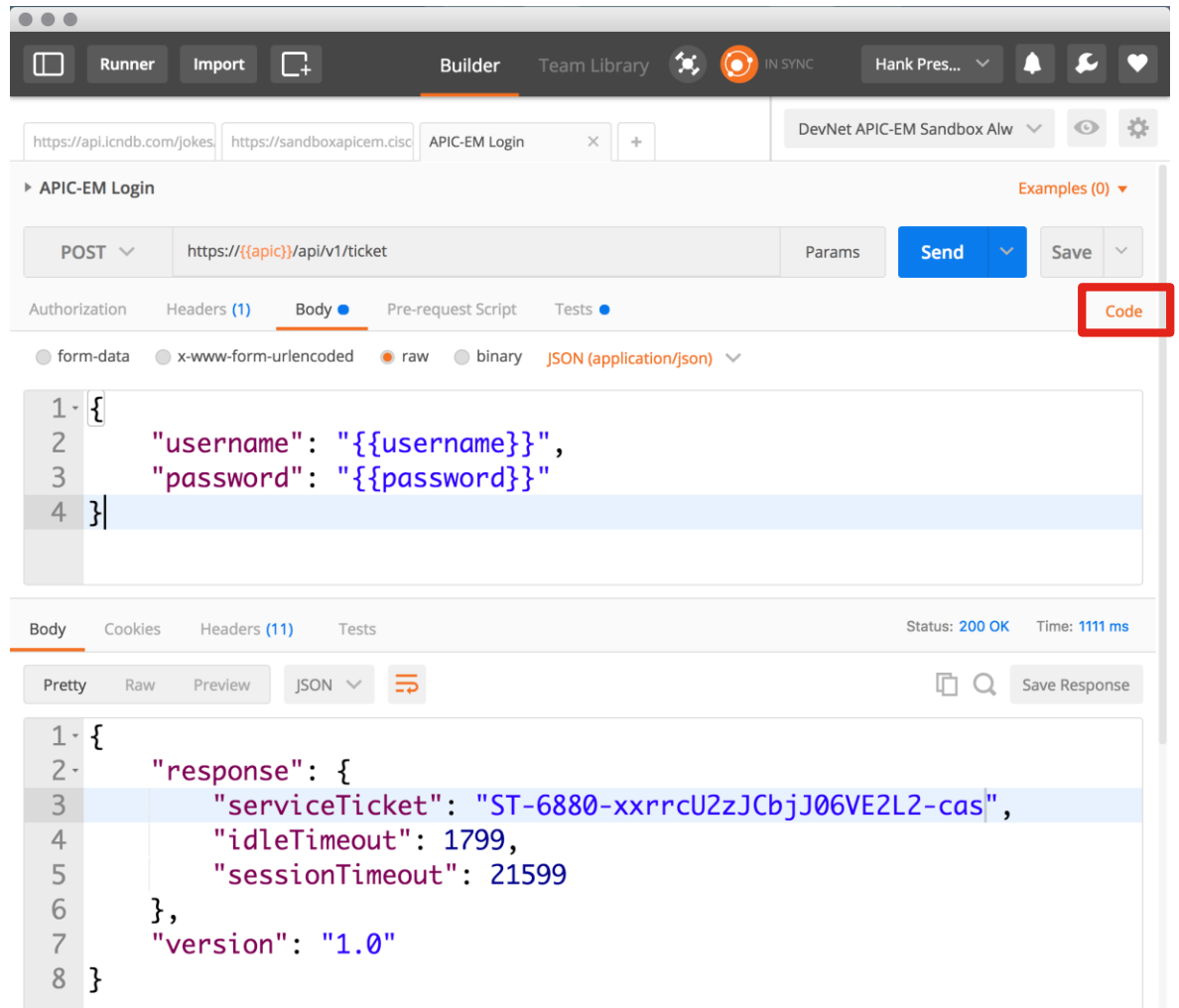
Demo Time!



Postman to Code!

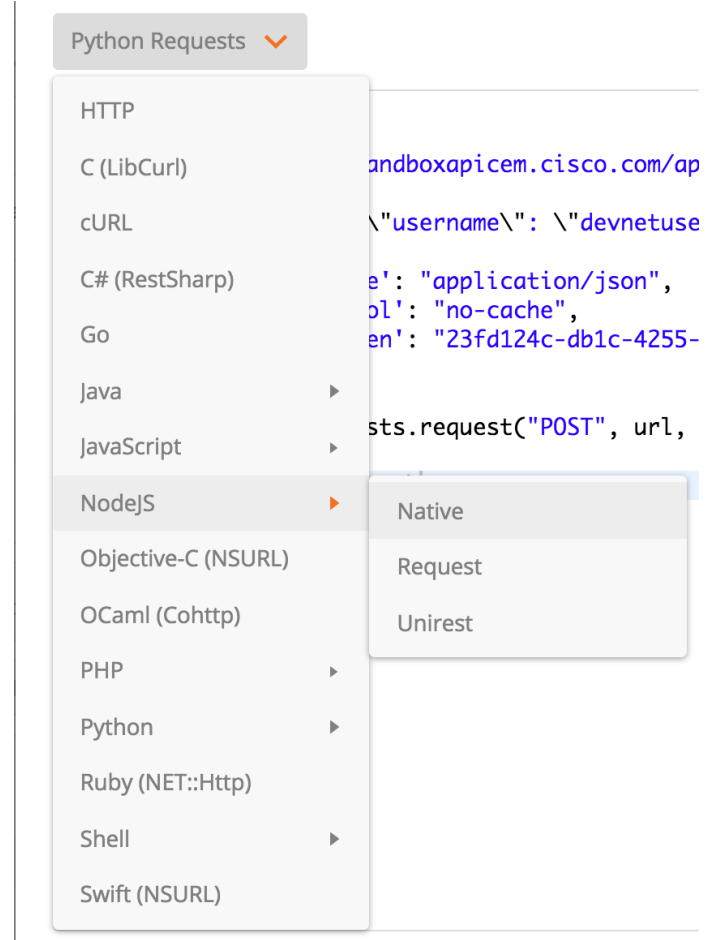
You'll eventually want to write some code...

- Postman great for testing and validating APIs
- But it's about atomic actions
- Business Logic, stringing APIs together, etc all need code
- Jumpstart with auto-generated code by Postman



You'll eventually want to write some code...

- Many, many options for languages available



Full API Request to Code!

- Headers, payload data, and URI all included
- Environment variables are translated
- Great starting point, but expect to edit and update

GENERATE CODE SNIPPETS ×

Python Requests ▼ Copy to Clipboard

```
1 import requests
2
3 url = "https://sandboxapicem.cisco.com/api/v1/ticket"
4
5 payload = "{\n\t\"username\": \"devnetuser\", \n\t\"password\": \"Cisco123!\"\n}"
6 headers = {
7     'content-type': "application/json",
8     'cache-control': "no-cache",
9     'postman-token': "23fd124c-db1c-4255-83fe-b97d57a59b29"
10 }
11
12 response = requests.request("POST", url, data=payload, headers=headers)
13
14 print(response.text)
```

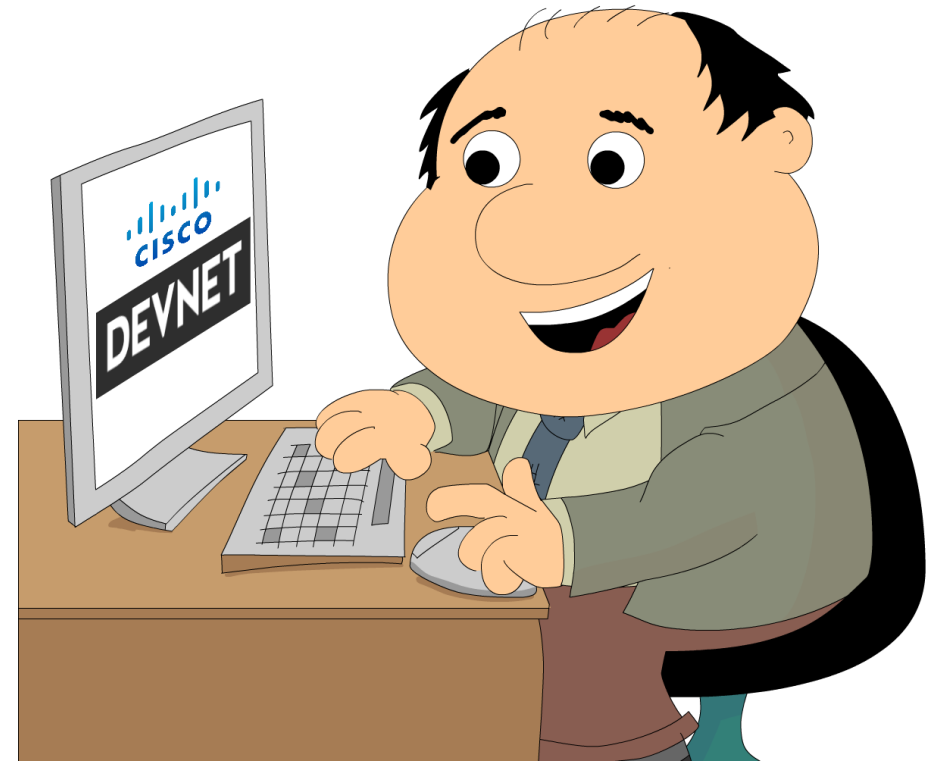

Summing up

Review

- Understand what Postman offers for API developers
- Learned how to send basic API requests
- Save API requests into Collections to re-use later
- Make APIs more modular and powerful with Environments
- Auto-generate code in many languages from an API call

Call to Action!

- Complete the full **Network Programmability Basics** Course
- Run the examples and exercises yourself!
 - Bonus Examples!
- Join [DevNet](#) for so much more!
 - [Learning Labs](#)
 - [Development Sandboxes](#)
 - Code Samples and API Guides



Got more questions? Come find me!

 hapresto@cisco.com

 [@hfpreston](https://twitter.com/hfpreston)

 <http://github.com/hpreston>

 [@CiscoDevNet](https://twitter.com/CiscoDevNet)

 facebook.com/ciscodevnet/

 <http://github.com/CiscoDevNet>





DEVNET