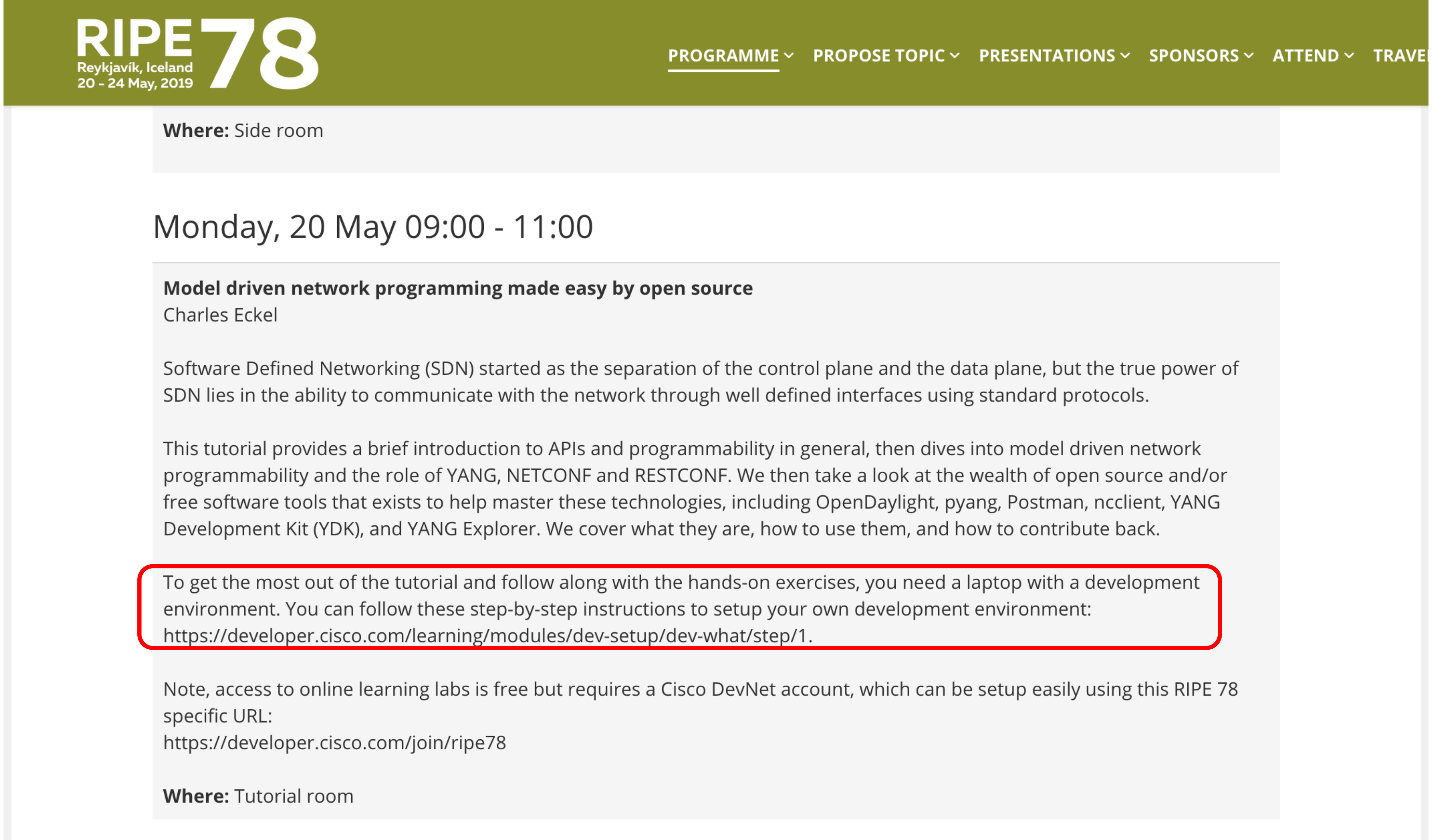
Model driven network programming made easy by open source

RIPE 78 Tutorial, 20 May 2019

Charles Eckel, Cisco DevNet
eckelcu@cisco.com, @eckelcu



Where: Side room

Monday, 20 May 09:00 - 11:00

Model driven network programming made easy by open source
Charles Eckel

Software Defined Networking (SDN) started as the separation of the control plane and the data plane, but the true power of SDN lies in the ability to communicate with the network through well defined interfaces using standard protocols.

This tutorial provides a brief introduction to APIs and programmability in general, then dives into model driven network programmability and the role of YANG, NETCONF and RESTCONF. We then take a look at the wealth of open source and/or free software tools that exists to help master these technologies, including OpenDaylight, pyang, Postman, ncclient, YANG Development Kit (YDK), and YANG Explorer. We cover what they are, how to use them, and how to contribute back.

To get the most out of the tutorial and follow along with the hands-on exercises, you need a laptop with a development environment. You can follow these step-by-step instructions to setup your own development environment: https://developer.cisco.com/learning/modules/dev-setup/dev-what/step/1.

Note, access to online learning labs is free but requires a Cisco DevNet account, which can be setup easily using this RIPE 78 specific URL:
https://developer.cisco.com/join/ripe78

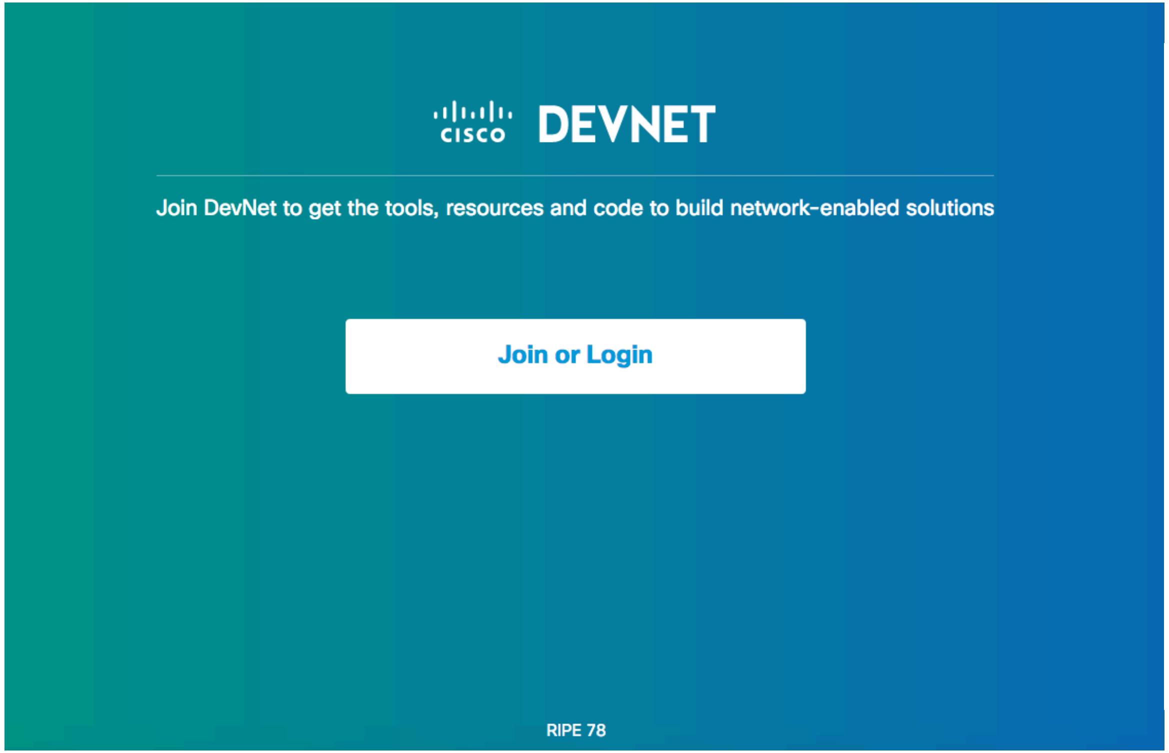
Where: Tutorial room

Agenda

- Setup
- Introduction to APIs
- REST APIs
- Network programmability
- Hands-on exercises

© 2018 Cisco and/or its affiliates. All rights reserved. Cisco Public

https://developer.cisco.com/join/ripe78

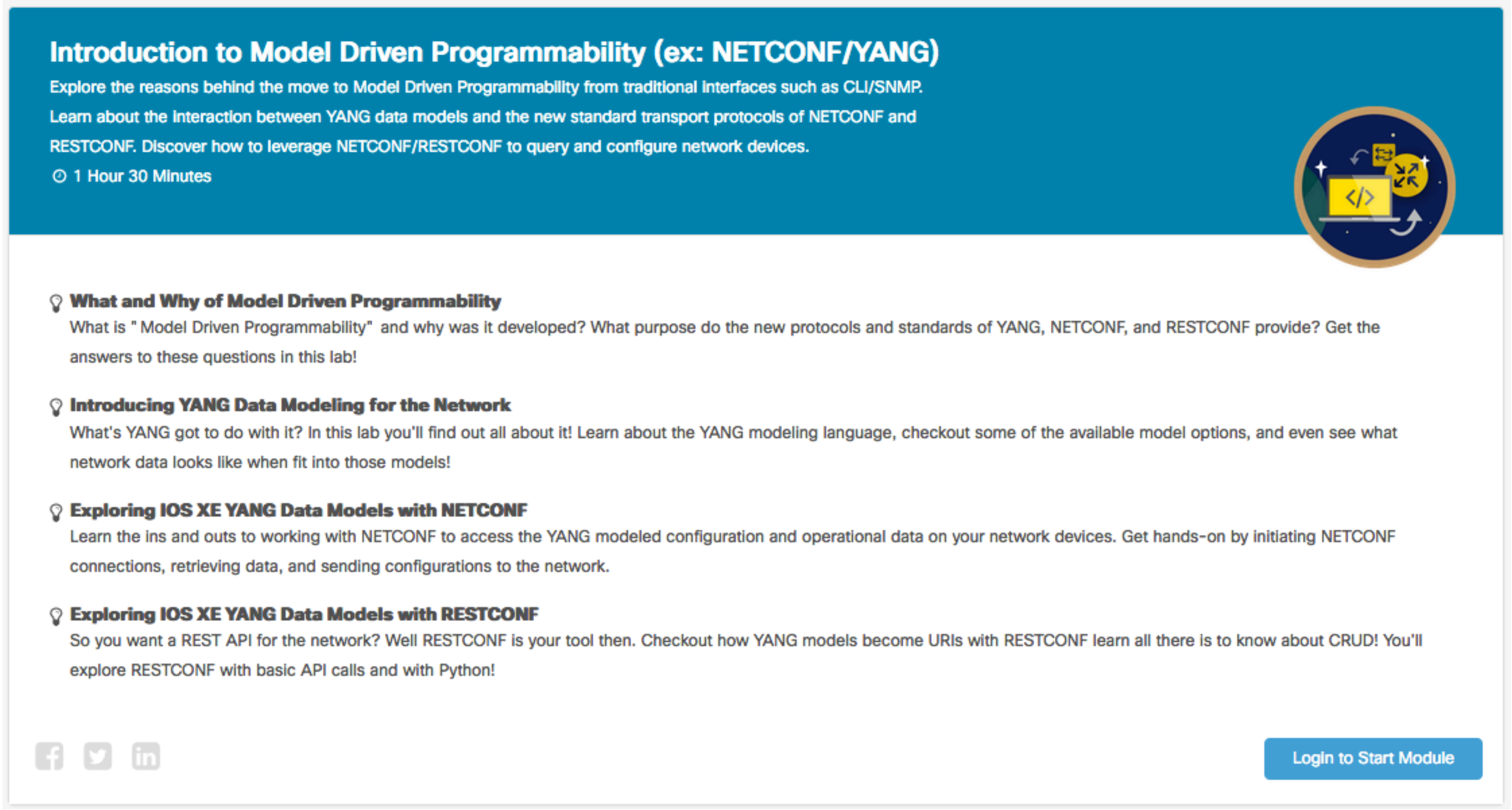


Join DevNet to get the tools, resources and code to build network-enabled solutions

RIPE 78

© 2018 Cisco and/or its affiliates. All rights reserved. Cisco Public

https://developer.cisco.com/learning/modules/intro-device-level-interfaces



Introduction to Model Driven Programmability (ex: NETCONF/YANG)

Explore the reasons behind the move to Model Driven Programmability from traditional interfaces such as CLI/SNMP. Learn about the interaction between YANG data models and the new standard transport protocols of NETCONF and RESTCONF. Discover how to leverage NETCONF/RESTCONF to query and configure network devices.

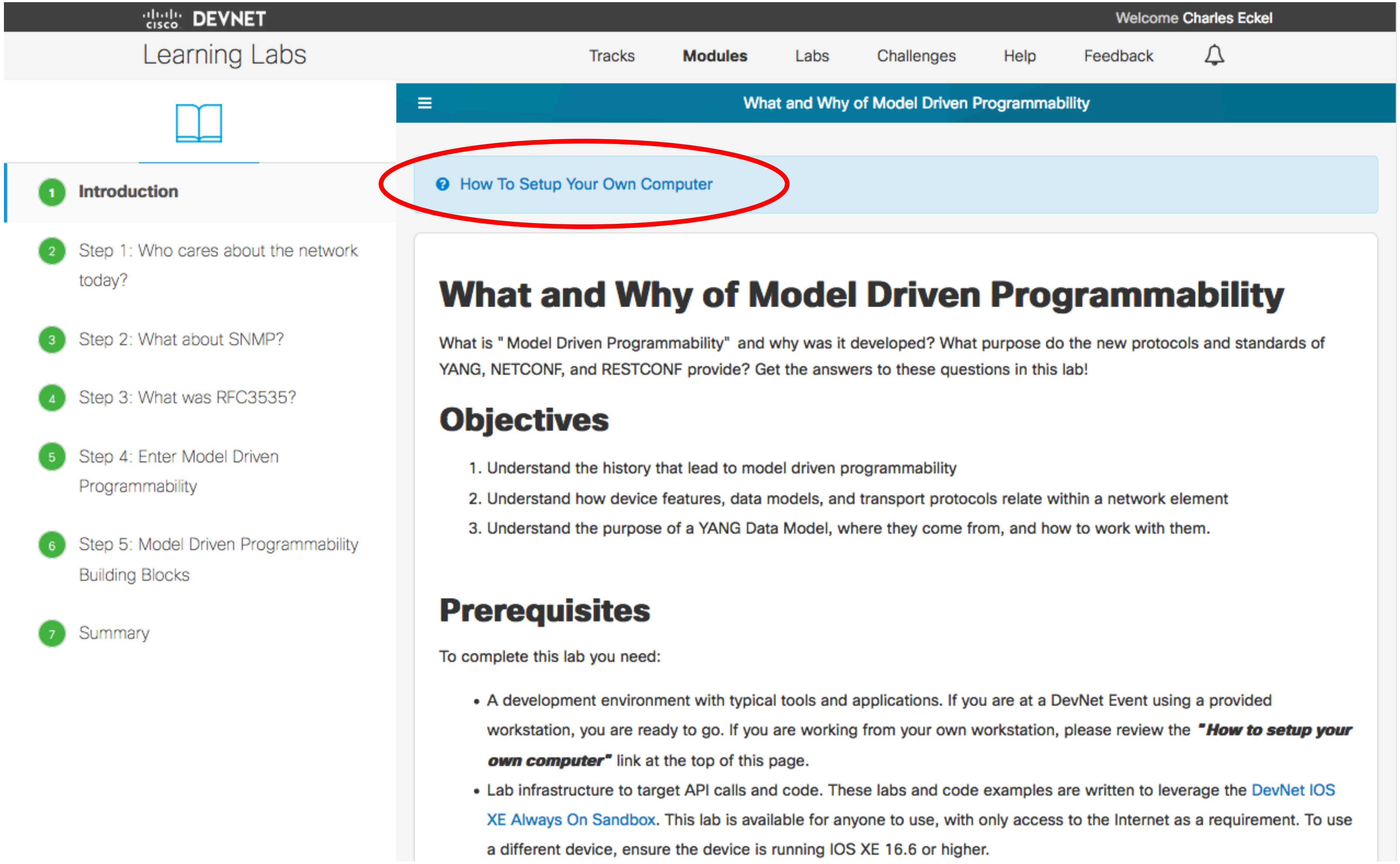
1 Hour 30 Minutes

What and Why of Model Driven Programmability
What is " Model Driven Programmability" and why was it developed? What purpose do the new protocols and standards of YANG, NETCONF, and RESTCONF provide? Get the answers to these questions in this lab!

Introducing YANG Data Modeling for the Network
What's YANG got to do with it? In this lab you'll find out all about it! Learn about the YANG modeling language, checkout some of the available model options, and even see what network data looks like when fit into those models!

Exploring IOS XE YANG Data Models with NETCONF
Learn the ins and outs to working with NETCONF to access the YANG modeled configuration and operational data on your network devices. Get hands-on by initiating NETCONF connections, retrieving data, and sending configurations to the network.

Exploring IOS XE YANG Data Models with RESTCONF
So you want a REST API for the network? Well RESTCONF is your tool then. Checkout how YANG models become URIs with RESTCONF learn all there is to know about CRUD! You'll explore RESTCONF with basic API calls and with Python!



1 Introduction

2 Step 1: Who cares about the network today?

3 Step 2: What about SNMP?

4 Step 3: What was RFC3535?

5 Step 4: Enter Model Driven Programmability

6 Step 5: Model Driven Programmability Building Blocks

7 Summary

What and Why of Model Driven Programmability

What is " Model Driven Programmability" and why was it developed? What purpose do the new protocols and standards of YANG, NETCONF, and RESTCONF provide? Get the answers to these questions in this lab!

Objectives

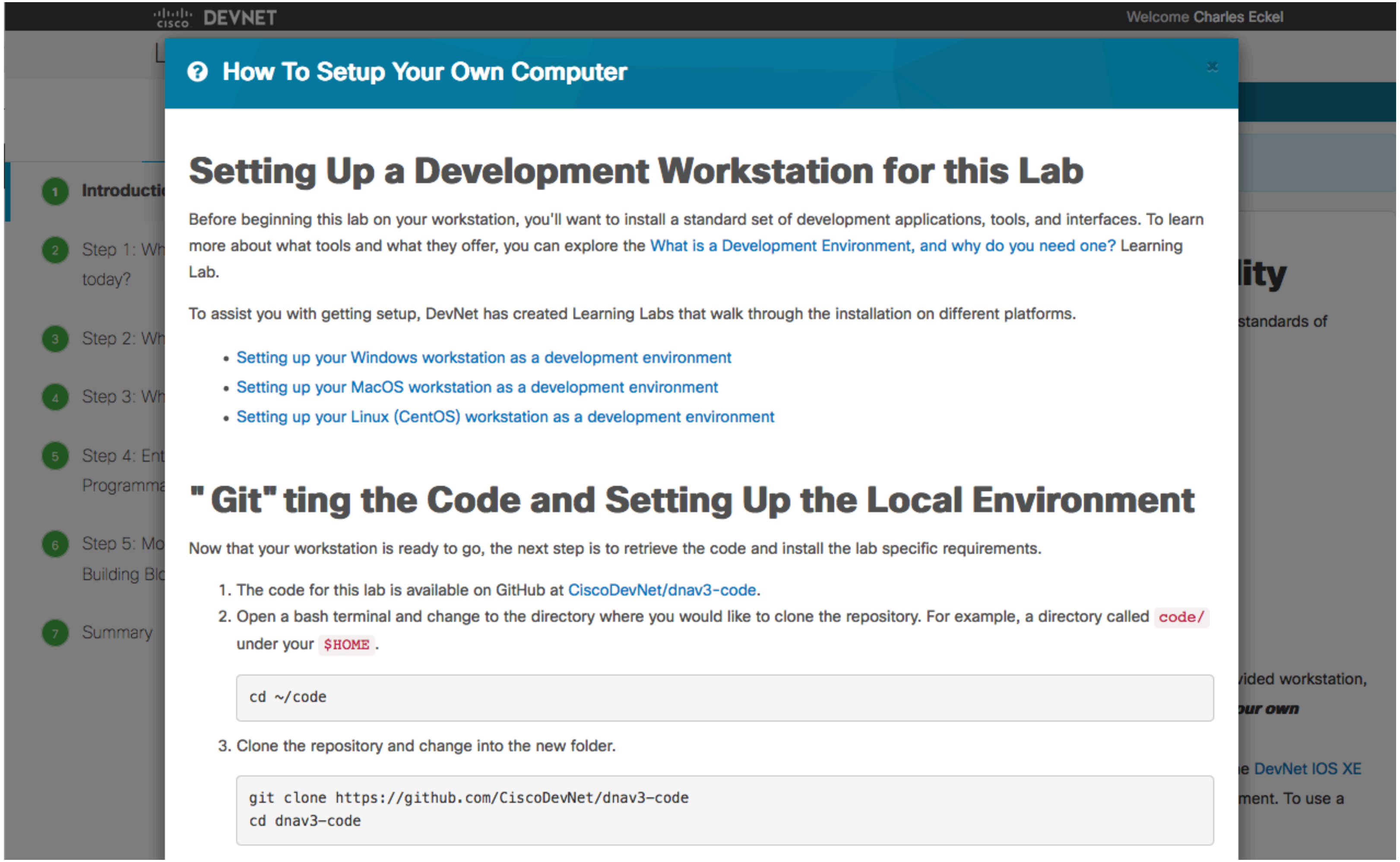
1. Understand the history that lead to model driven programmability
2. Understand how device features, data models, and transport protocols relate within a network element
3. Understand the purpose of a YANG Data Model, where they come from, and how to work with them.

Prerequisites

To complete this lab you need:

- A development environment with typical tools and applications. If you are at a DevNet Event using a provided workstation, you are ready to go. If you are working from your own workstation, please review the ***"How to setup your own computer"*** link at the top of this page.
- Lab infrastructure to target API calls and code. These labs and code examples are written to leverage the DevNet IOS XE Always On Sandbox. This lab is available for anyone to use, with only access to the Internet as a requirement. To use a different device, ensure the device is running IOS XE 16.6 or higher.

© 2018 Cisco and/or its affiliates. All rights reserved. Cisco Public



How To Setup Your Own Computer

Setting Up a Development Workstation for this Lab

Before beginning this lab on your workstation, you'll want to install a standard set of development applications, tools, and interfaces. To learn more about what tools and what they offer, you can explore the What is a Development Environment, and why do you need one? Learning Lab.

To assist you with getting setup, DevNet has created Learning Labs that walk through the installation on different platforms.

- Setting up your Windows workstation as a development environment
- Setting up your MacOS workstation as a development environment
- Setting up your Linux (CentOS) workstation as a development environment

"Git" ting the Code and Setting Up the Local Environment

Now that your workstation is ready to go, the next step is to retrieve the code and install the lab specific requirements.

1. The code for this lab is available on GitHub at CiscoDevNet/dnav3-code.
2. Open a bash terminal and change to the directory where you would like to clone the repository. For example, a directory called `code/` under your `$HOME`.

```
cd ~/code
```

3. Clone the repository and change into the new folder.

```
git clone https://github.com/CiscoDevNet/dnav3-code
cd dnav3-code
```

© 2018 Cisco and/or its affiliates. All rights reserved. Cisco Public

Introduction to APIs

© 2018 Cisco and/or its affiliates. All rights reserved. Cisco Public

Application Programming Interface

"It's a way for two pieces of software to talk to each other"

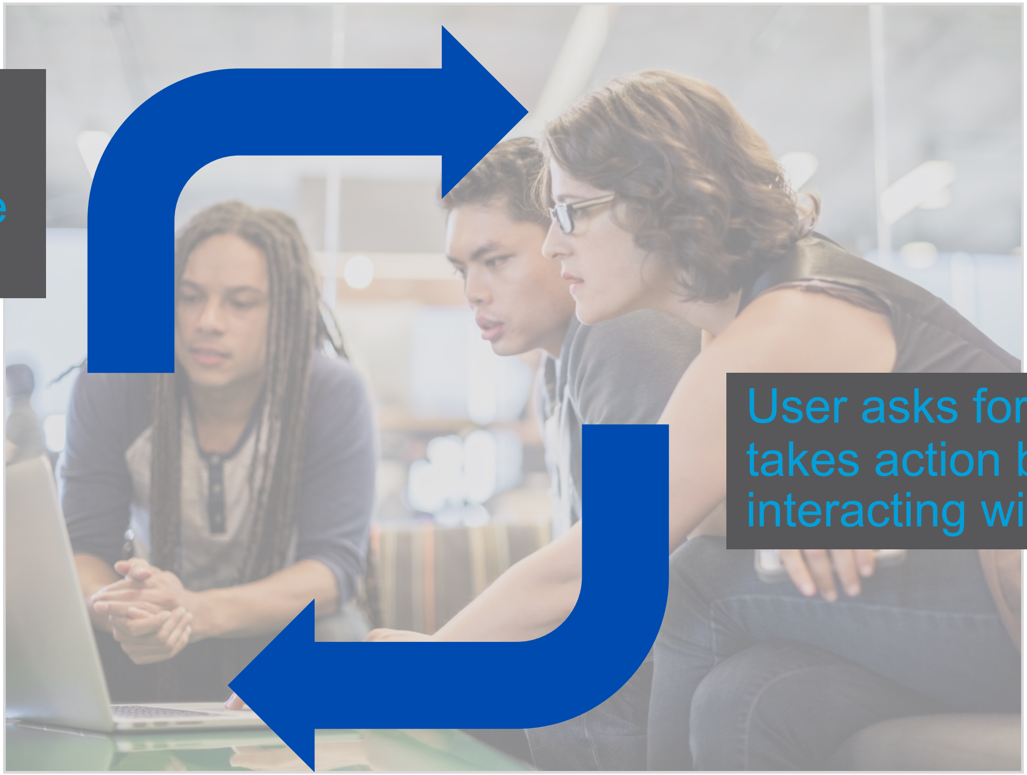
© 2018 Cisco and/or its affiliates. All rights reserved. Cisco Public

For a time.. Humans were the only users

© 2018 Cisco and/or its affiliates. All rights reserved. Cisco Public

For a time.. Humans were the only users

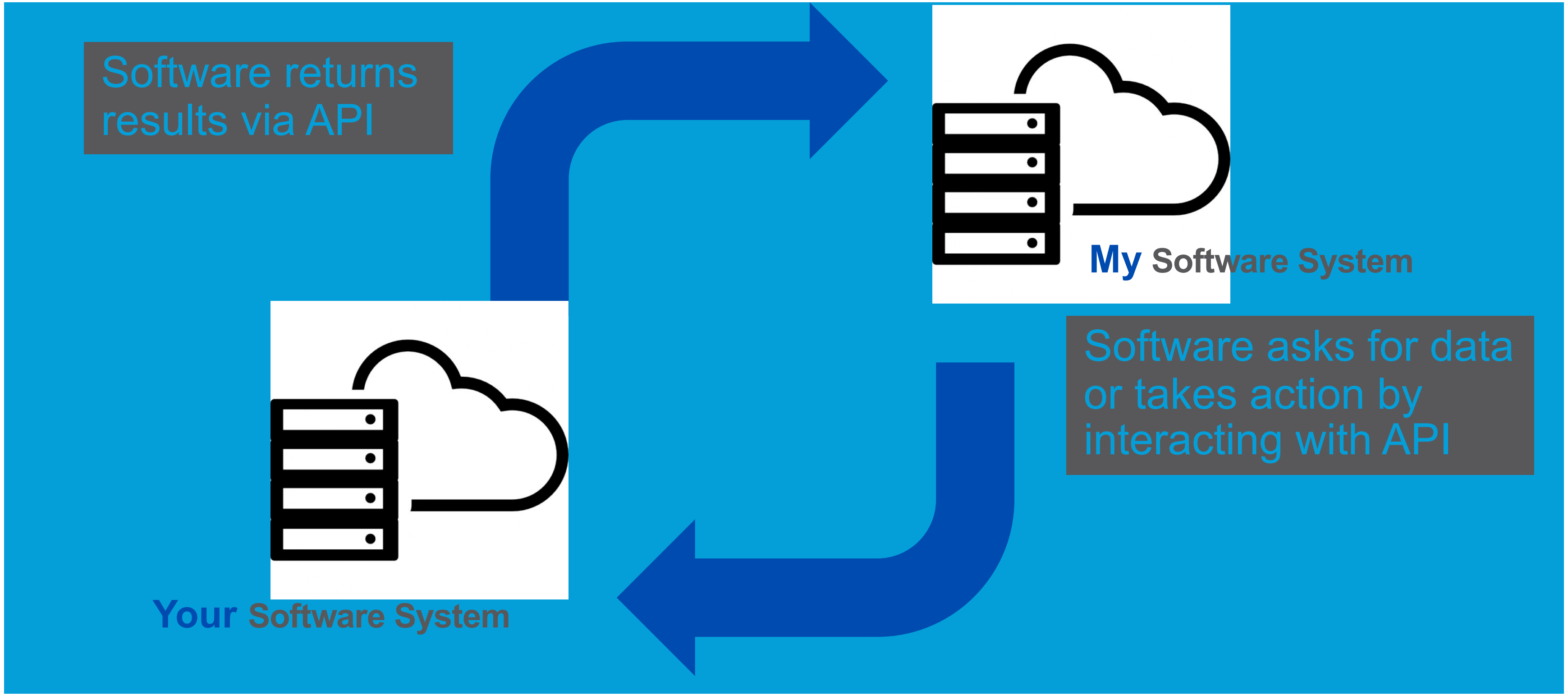
Software displays results in User Interface (UI)



User asks for data or takes action by interacting with UI

© 2018 Cisco and/or its affiliates. All rights reserved. Cisco Public

But what about when the user is another software system….


Software returns results via API
My Software System
Software asks for data or takes action by interacting with API
Your Software System

© 2018 Cisco and/or its affiliates. All rights reserved. Cisco Public

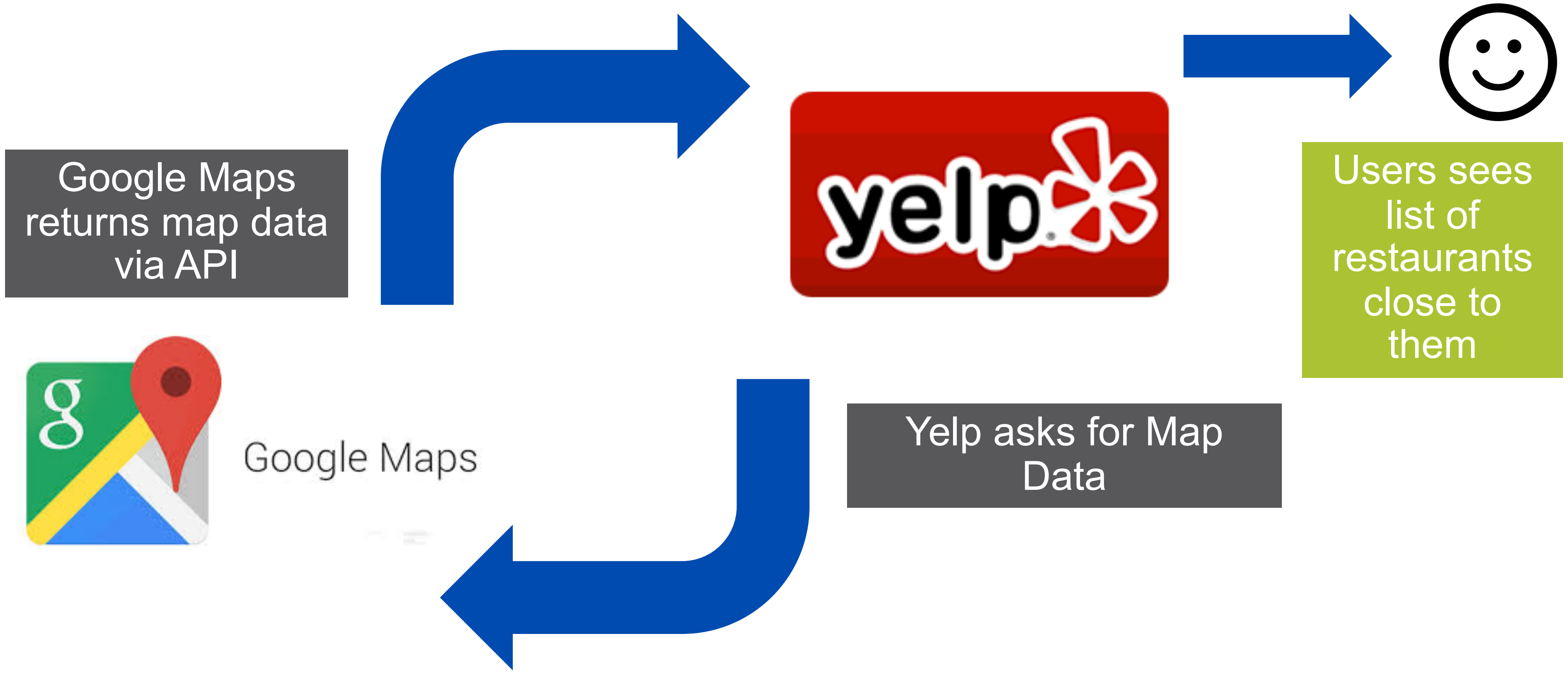
The API is the User Interface for software systems

© 2018 Cisco and/or its affiliates. All rights reserved. Cisco Public

APIs are sets of requirements that govern how one application can talk to another.

© 2018 Cisco and/or its affiliates. All rights reserved. Cisco Public

APIs help developers create apps that benefit the end user


Google Maps returns map data via API
yelp
Users sees list of restaurants close to them
Google Maps
Yelp asks for Map Data

© 2018 Cisco and/or its affiliates. All rights reserved. Cisco Public

“APIs are often referred to as “an engine of innovation.”

-- Programmable Web

REST APIs

REST Web service

- **What is REST?**
  - **RE**presentational **S**tate **T**ransfer (REST)
  - API framework built on HTTP
- **What is a REST Web Service?**
  - REST is *an architecture style* for designing networked applications**.**
  - Popular due to performance, scale, simplicity, and reliability

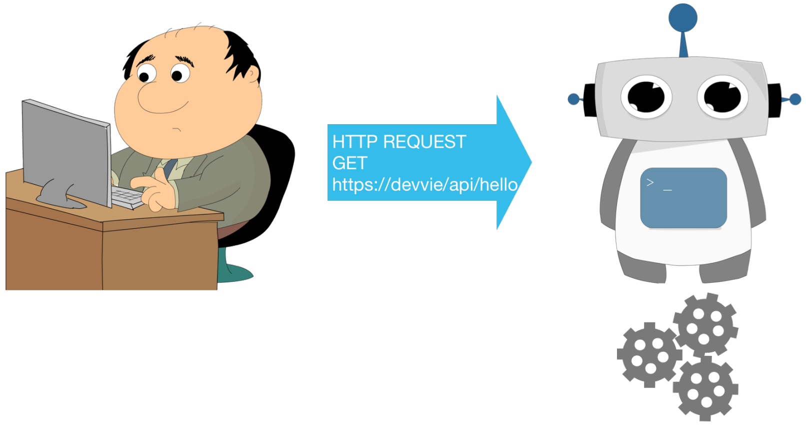
GET
POST
PUT
DELETE
{REST}

Request and Response, the REST API Flow

HTTP REQUEST
GET
https://devvie/api/hello
> _

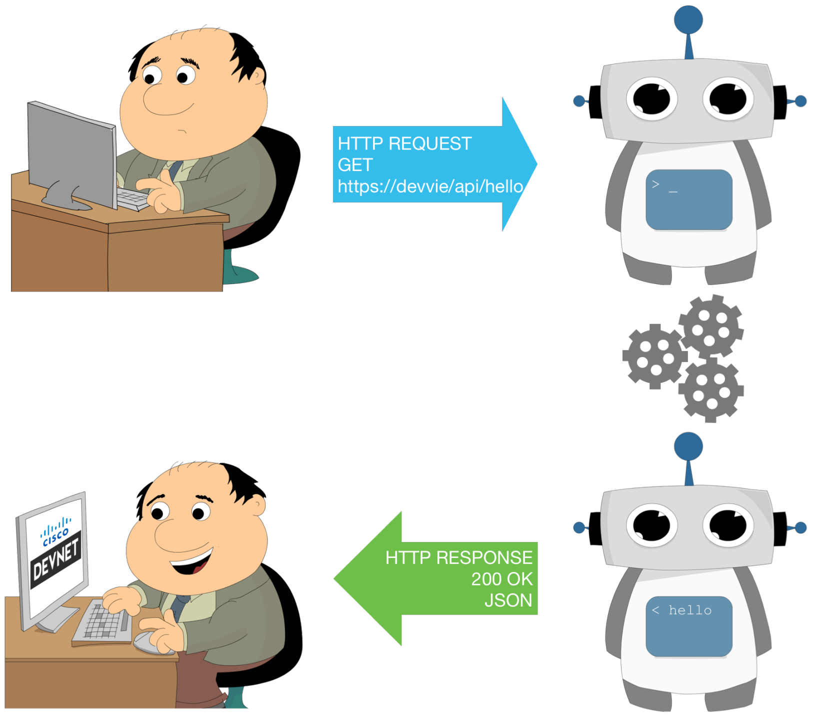
© 2018 Cisco and/or its affiliates. All rights reserved. Cisco Public

Request and Response, the REST API Flow



© 2018 Cisco and/or its affiliates. All rights reserved. Cisco Public

Request and Response, the REST API Flow



© 2018 Cisco and/or its affiliates. All rights reserved. Cisco Public

HTTP Methods: What to do?

HTTP Verb	Typical Purpose (CRUD)	Description
POST	Create	Used to create a new object, or resource. Example: Add new book to library
GET	Read	Retrieve resource details from the system. Example: Get list of books from the library
PUT	Update	Typically used to replace or update a resource. Can be used to modify or create. Example: Update the borrower details for a book
PATCH	Update	Used to modify some details about a resource. Example: Change the author of a book
DELETE	Delete	Remove a resource from the system. Example: Delete a book from the library.

© 2018 Cisco and/or its affiliates. All rights reserved. Cisco Public

Response Status Codes: Did it work?

	Status Code	Status Message	Meaning
2xx	200	OK	All looks good
	201	Created	New resource created
	202	Accepted	Accepted for processing, but processing not completed
	204	No Content	Request succeeded, but no message body returned
4xx	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
5xx	500	Internal Server Error	Something wrong with the server
	503	Service Unavailable	Server is unable to complete request

© 2018 Cisco and/or its affiliates. All rights reserved. Cisco Public

The URI: What are you Requesting?

$$\text{https://}\underbrace{\text{deckofcardsapi.com}}_{\text{Server or Host}}/\underbrace{\text{api/deck/new/shuffle}}_{\text{Resource}}/\underbrace{\text{?deck\_count=1}}_{\text{Parameters}}$$

- **http:// or https://**
  - Protocol over which data is sent between client and server
  - ‘s’ in https stands for secure
- **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.

© 2018 Cisco and/or its affiliates. All rights reserved. Cisco Public

Data: Sending and Receiving

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

```
{
    "success": true,
    "deck_id": "3p40paa87x90",
    "shuffled": true,
    "remaining": 52
}
```

© 2018 Cisco and/or its affiliates. All rights reserved. Cisco Public

Headers:

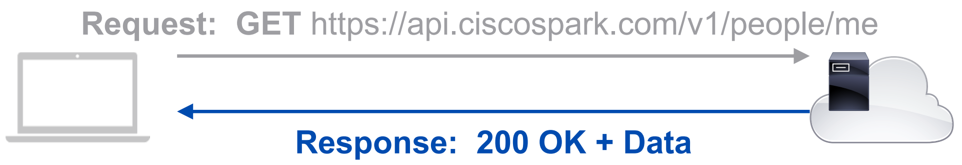
What additional details and metadata can I use?

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
Authorization	Basic dmFncmFudDp2YWdyYW50	Provide credentials to authorize a request
Date	Tue, 25 Jul 2017 19:26:00 GMT	Date and time of the message

- Used to pass information between client and server
- Included in both REQUEST and RESPONSE
- Some APIs use custom headers for authentication or other purpose

© 2018 Cisco and/or its affiliates. All rights reserved. Cisco Public

Review: Request/Response



HTTPS Request

Request Headers

```
GET /v1/people/me HTTPS/1.1
Host: api.ciscospark.com
Authorization: Bearer <redacted>
Accept: */*
Accept-Encoding: gzip, deflate, sdch
Connection: keep-alive
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_4) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/49.0.2623.112 Safari/537.36
```

HTTPS Response

Response Headers

<blank line>

Response Payload

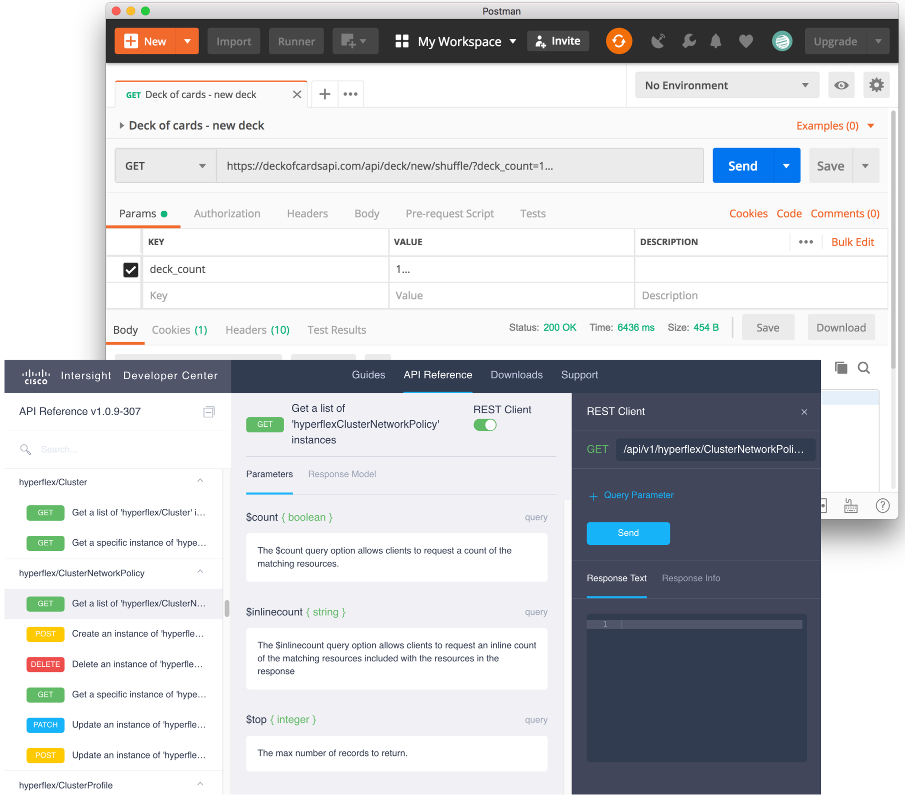
```
HTTPS/1.1 200 OK
Date: Wed, 23 Jan 2019 23:12:11 GMT
Content-Type: application/json;charset=UTF-8
Content-Encoding: gzip
Content-Length: 323
Trackingid: ROUTER_5C48F4B1-9789-01BB-4148-xxxxxxxxx
Vary: Accept-Encoding
Strict-Transport-Security: max-age=63072000; includeSubDomains; preload

{
"id":
"Y2lzY29zcGFyazovL3VzL1BFT1BMRS9iODBjM2NmOC01ZGIwLTQyNzAtOThiZS1mYzFhYjA3MzE1YWE",
"emails": ["eckelcu@cisco.com"],
"displayName": "Charles Eckel",
"nickName": "Charles",
"firstName": "Charles",
"lastName": "Eckel",
:
"status": "active",
"type": "person"
}
```

Note: This is all exchanged as simple text over a TCP/TLS connection.

© 2018 Cisco and/or its affiliates. All rights reserved. Cisco Public

Many Options for Working with REST APIs

- Web browser
  - Chrome, Firefox, etc.
- curl
  - Linux command line application
- Postman
  - API testing application and framework
- Requests
  - Python library for scripting
- OpenAPI/Swagger
  - Dynamic API Documentation



Postman
New
Import
Runner
My Workspace
Invite
Upgrade
GET Deck of cards - new deck
No Environment
Deck of cards - new deck
Examples (0)
GET
https://deckofcardsapi.com/api/deck/new/shuffle/?deck_count=1...
Send
Save
Params
Authorization
Headers
Body
Pre-request Script
Tests
Cookies
Code
Comments (0)
KEY
VALUE
DESCRIPTION
Bulk Edit
deck_count
1...
Key
Value
Description
Body
Cookies (1)
Headers (10)
Test Results
Status: 200 OK
Time: 6436 ms
Size: 454 B
Save
Download
Intersight Developer Center
Guides
API Reference
Downloads
Support
API Reference v1.0.9-307
Search...
hyperflex/Cluster
GET Get a list of 'hyperflex/Cluster' i...
GET Get a specific instance of 'hype...
hyperflex/ClusterNetworkPolicy
GET Get a list of 'hyperflex/ClusterN...
POST Create an instance of 'hyperfle...
DELETE Delete an instance of 'hyperfle...
GET Get a specific instance of 'hype...
PATCH Update an instance of 'hyperfle...
POST Update an instance of 'hyperfle...
hyperflex/ClusterProfile
GET Get a list of 'hyperflexClusterNetworkPolicy' instances
REST Client
Parameters
Response Model
$count { boolean } query
$inlinecount { string } query
$stop { integer } query
REST Client
GET /api/v1/hyperflex/ClusterNetworkPoli...
+ Query Parameter
Send
Response Text
Response Info

The $count query option allows clients to request a count of the matching resources.

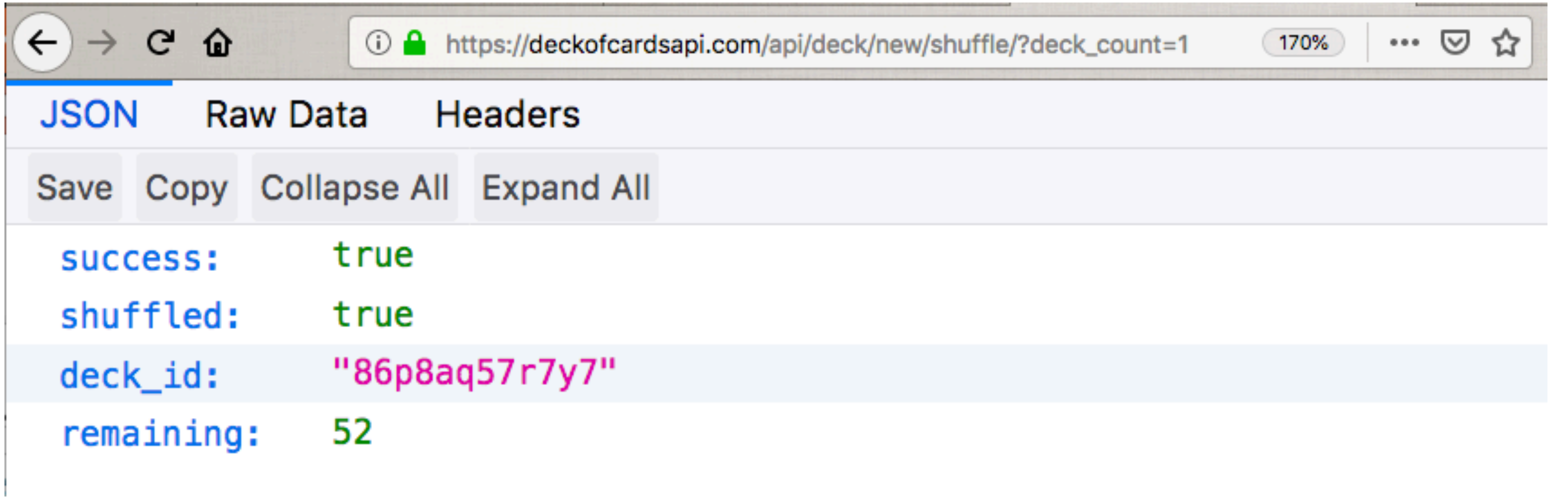
The $inlinecount query option allows clients to request an inline count of the matching resources included with the resources in the response

The max number of records to return.

© 2018 Cisco and/or its affiliates. All rights reserved. Cisco Public

Web Browser

https://deckofcardsapi.com/api/deck/new/shuffle/?deck_count=1

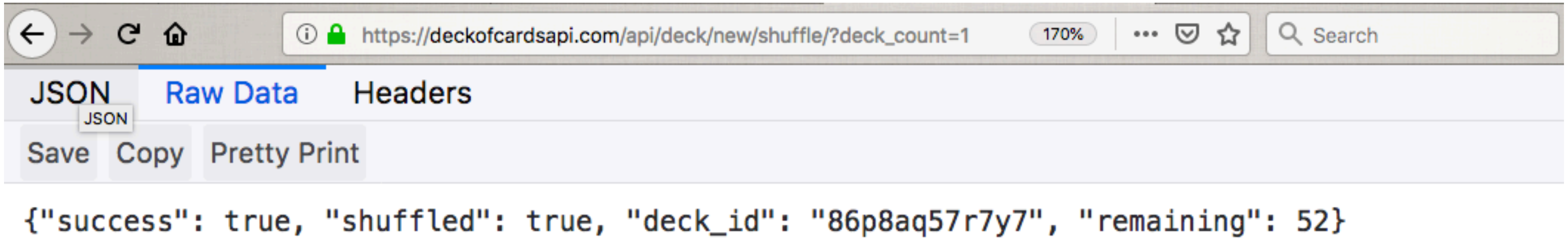

https://deckofcardsapi.com/api/deck/new/shuffle/?deck_count=1
170%
JSON
Raw Data
Headers
Save
Copy
Collapse All
Expand All

```
success:      true
shuffled:     true
deck_id:      "86p8aq57r7y7"
remaining:    52
```

© 2018 Cisco and/or its affiliates. All rights reserved. Cisco Public

Web Browser

https://deckofcardsapi.com/api/deck/new/shuffle/?deck_count=1

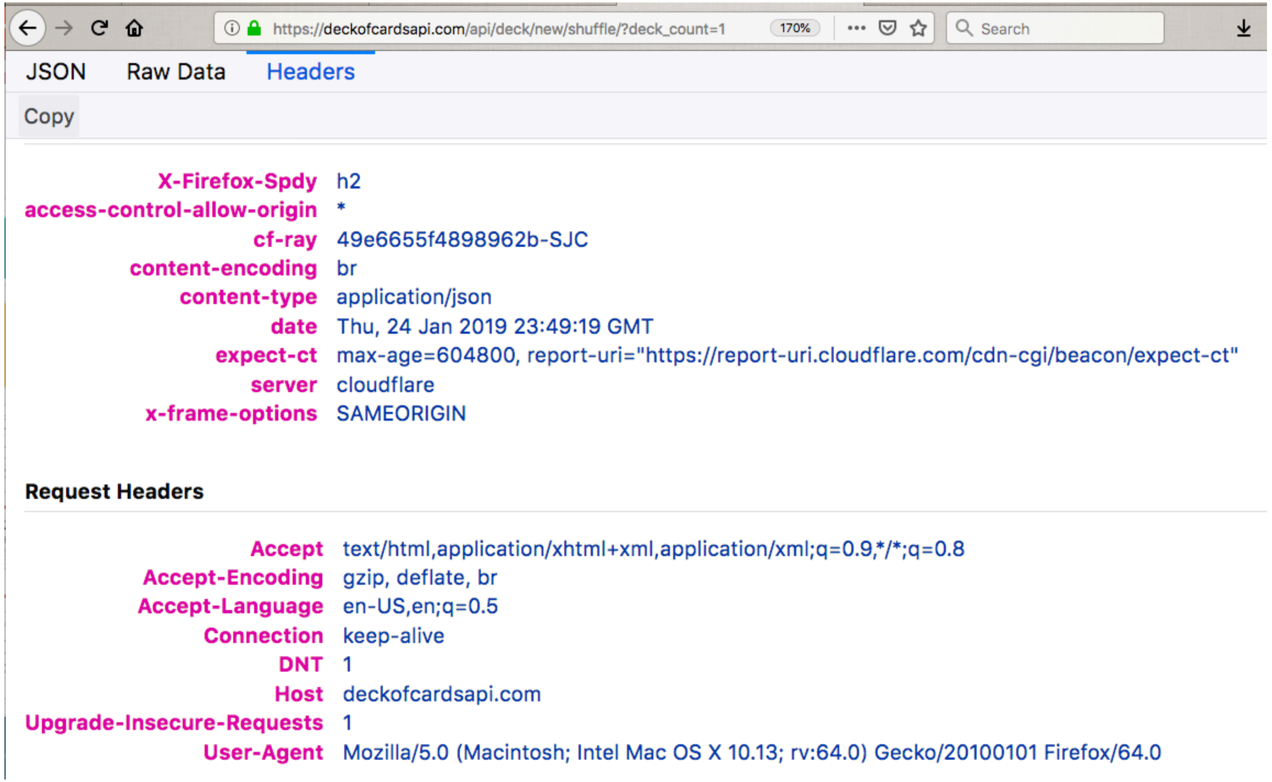

https://deckofcardsapi.com/api/deck/new/shuffle/?deck_count=1
170%
Search
JSON
Raw Data
Headers
JSON
Save
Copy
Pretty Print

```
{"success": true, "shuffled": true, "deck_id": "86p8aq57r7y7", "remaining": 52}
```

© 2018 Cisco and/or its affiliates. All rights reserved. Cisco Public

Web Browser

https://deckofcardsapi.com/api/deck/new/shuffle/?deck_count=1


https://deckofcardsapi.com/api/deck/new/shuffle/?deck_count=1
170%
Search
JSON
Raw Data
Headers
Copy

```
           X-Firefox-Spdy  h2
access-control-allow-origin  *
                   cf-ray  49e6655f4898962b-SJC
         content-encoding  br
             content-type  application/json
                     date  Thu, 24 Jan 2019 23:49:19 GMT
                expect-ct  max-age=604800, report-uri="https://report-uri.cloudflare.com/cdn-cgi/beacon/expect-ct"
                   server  cloudflare
          x-frame-options  SAMEORIGIN
```

Request Headers

```
                   Accept  text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
          Accept-Encoding  gzip, deflate, br
          Accept-Language  en-US,en;q=0.5
               Connection  keep-alive
                      DNT  1
                     Host  deckofcardsapi.com
Upgrade-Insecure-Requests  1
               User-Agent  Mozilla/5.0 (Macintosh; Intel Mac OS X 10.13; rv:64.0) Gecko/20100101 Firefox/64.0
```

© 2018 Cisco and/or its affiliates. All rights reserved. Cisco Public

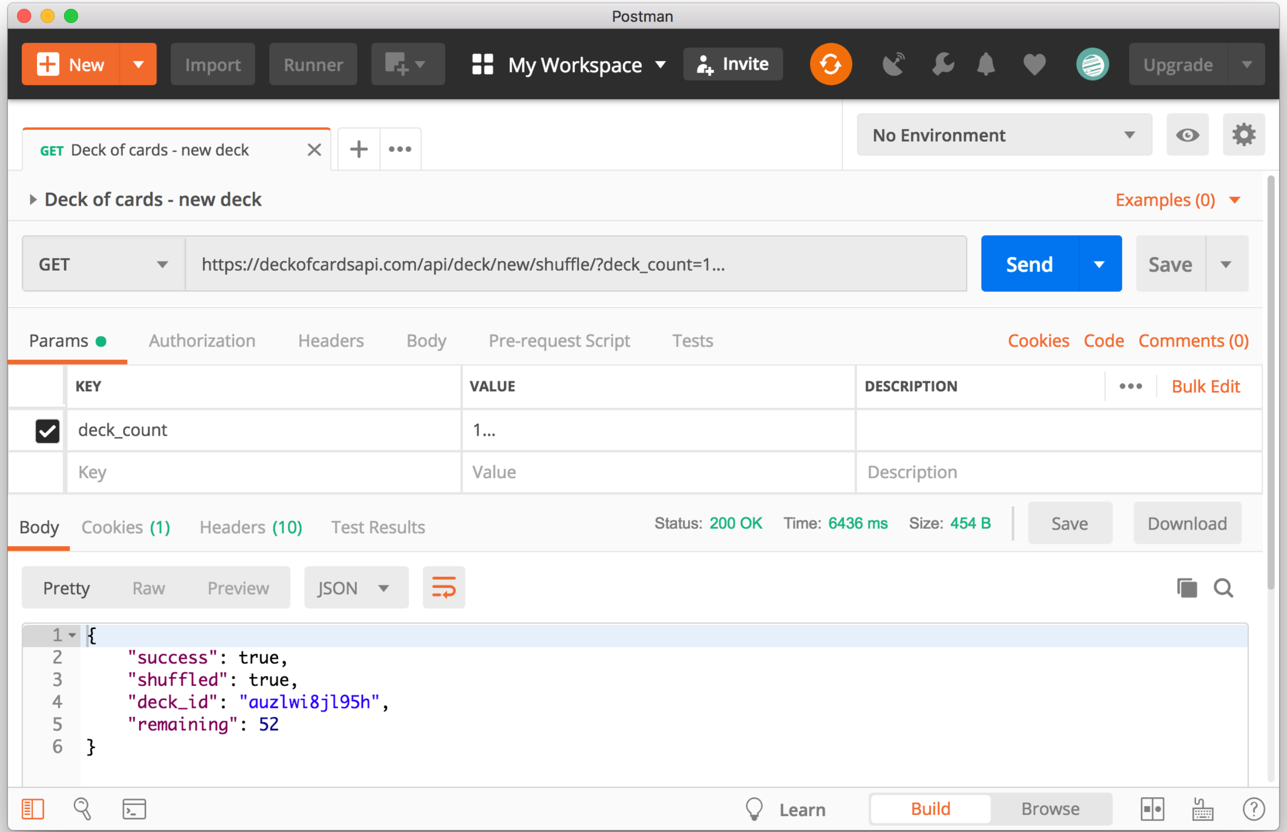
curl

```
$ curl
https://deckofcardsapi.com/api/deck/new/shuffle/?deck_count=1
```

```
{"success": true, "shuffled": true, "deck_id": "sr405eihisjl",
"remaining": 52}
```

© 2018 Cisco and/or its affiliates. All rights reserved. Cisco Public

Postman



GET Deck of cards - new deck

Deck of cards - new deck

GET https://deckofcardsapi.com/api/deck/new/shuffle/?deck_count=1...

Params ● | Authorization | Headers | Body | Pre-request Script | Tests

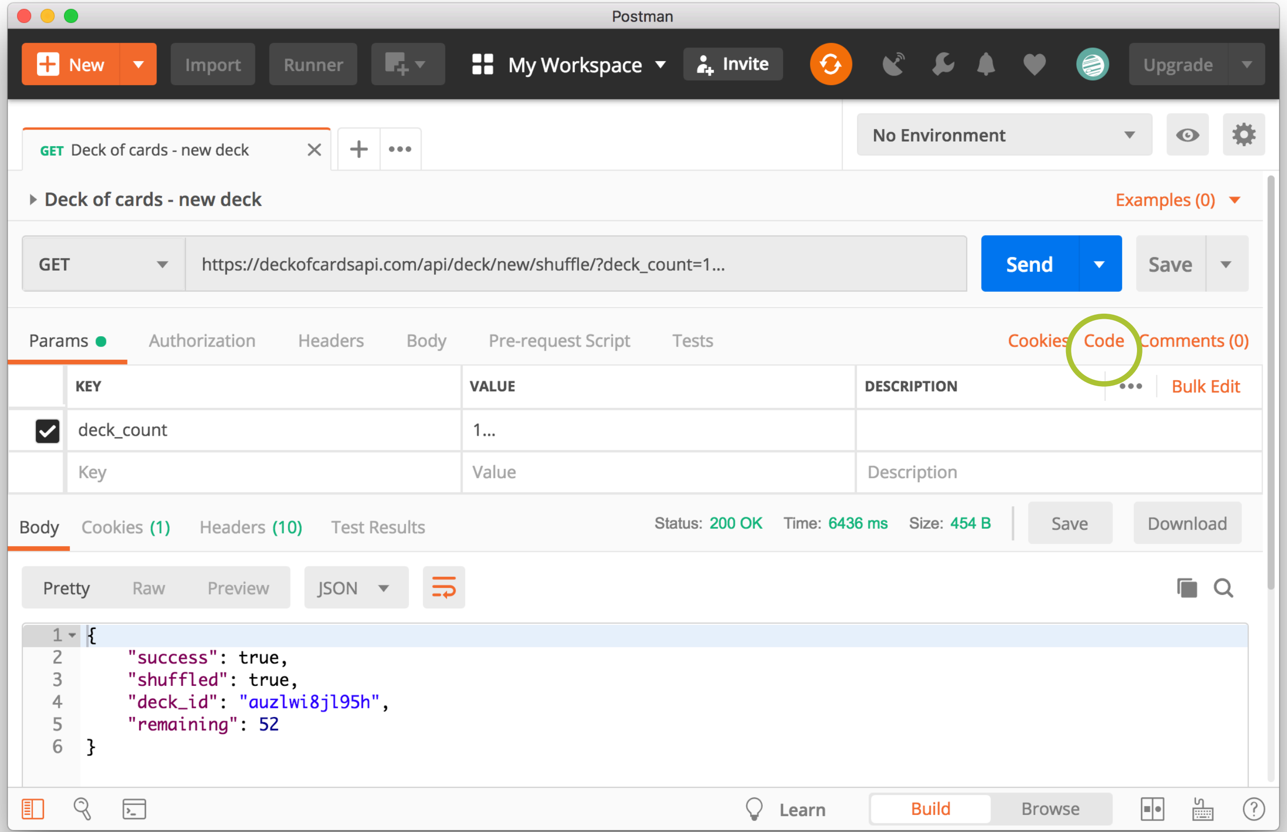
	KEY	VALUE	DESCRIPTION
✓	deck_count	1...	
	Key	Value	Description

Body | Cookies (1) | Headers (10) | Test Results

Status: 200 OK Time: 6436 ms Size: 454 B

```
{
    "success": true,
    "shuffled": true,
    "deck_id": "auzlwi8jl95h",
    "remaining": 52
}
```

Postman



https://deckofcardsapi.com/api/deck/new/shuffle/?deck_count=1...

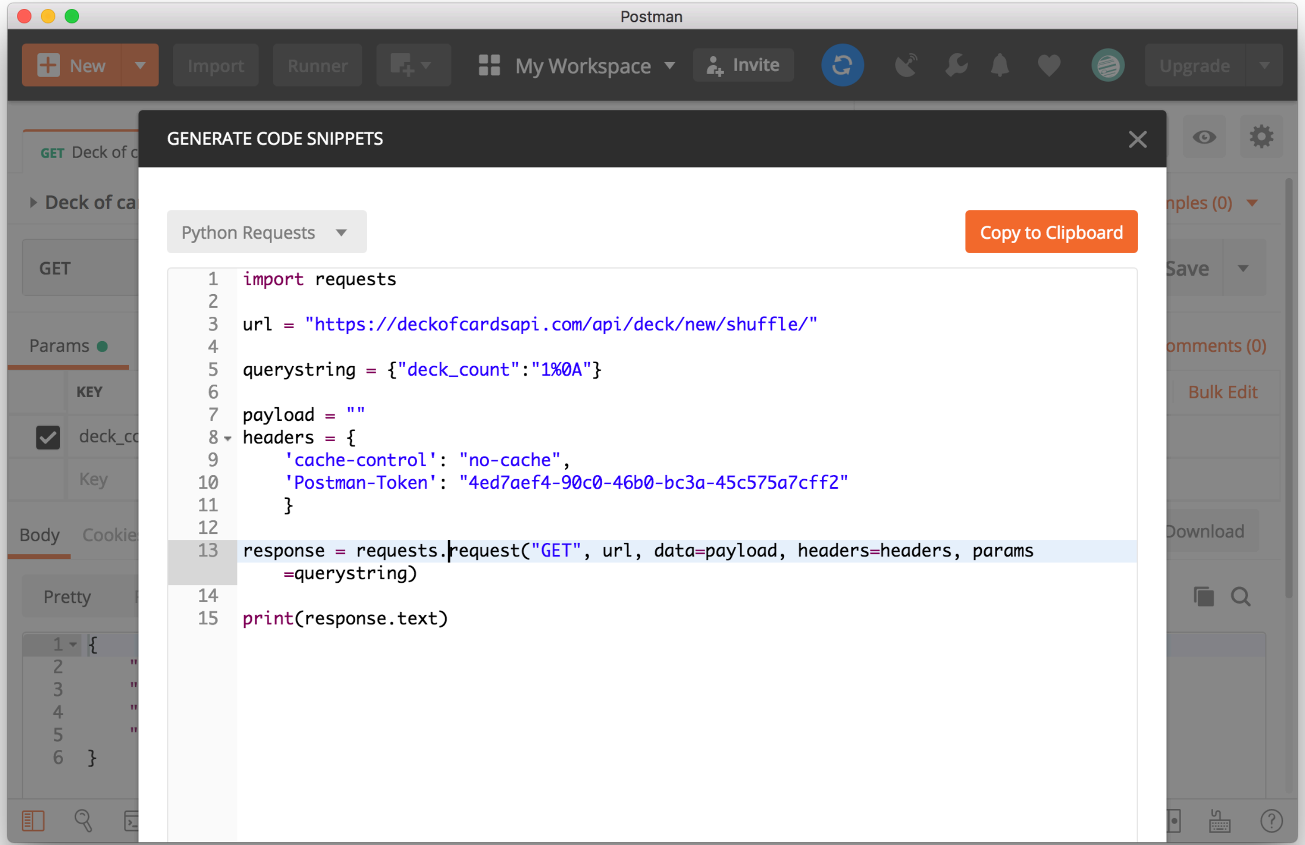
KEY	VALUE	DESCRIPTION
✓ deck_count	1...	
Key	Value	Description

Status: 200 OK Time: 6436 ms Size: 454 B

```
{
    "success": true,
    "shuffled": true,
    "deck_id": "auzlwi8jl95h",
    "remaining": 52
}
```

© 2018 Cisco and/or its affiliates. All rights reserved. Cisco Public

Python



```python
import requests

url = "https://deckofcardsapi.com/api/deck/new/shuffle/"

querystring = {"deck_count":"1%0A"}

payload = ""
headers = {
    'cache-control': "no-cache",
    'Postman-Token': "4ed7aef4-90c0-46b0-bc3a-45c575a7cff2"
    }

response = requests.request("GET", url, data=payload, headers=headers, params
    =querystring)

print(response.text)
```

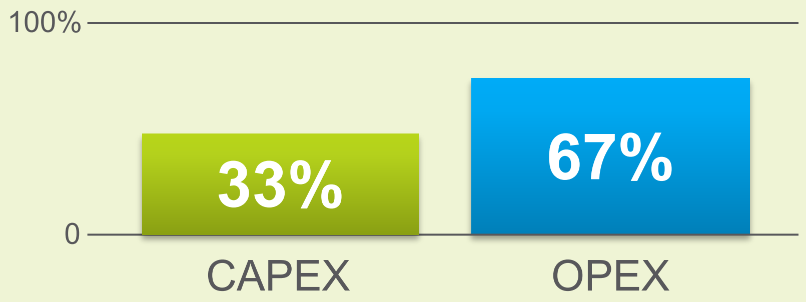
© 2018 Cisco and/or its affiliates. All rights reserved. Cisco Public

Network programmability

© 2018 Cisco and/or its affiliates. All rights reserved. Cisco Public

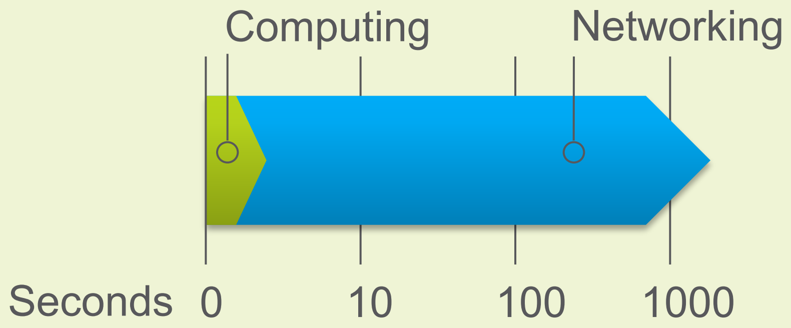
Why Network Programmability Matters

Network Expenses



Source: Forrester

Deployment Speed

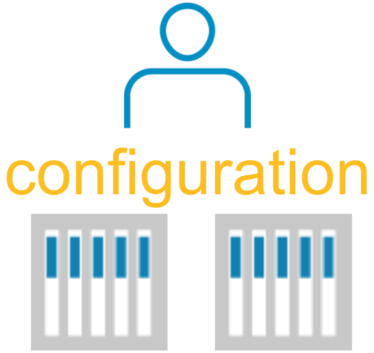


Source: Open Compute Project

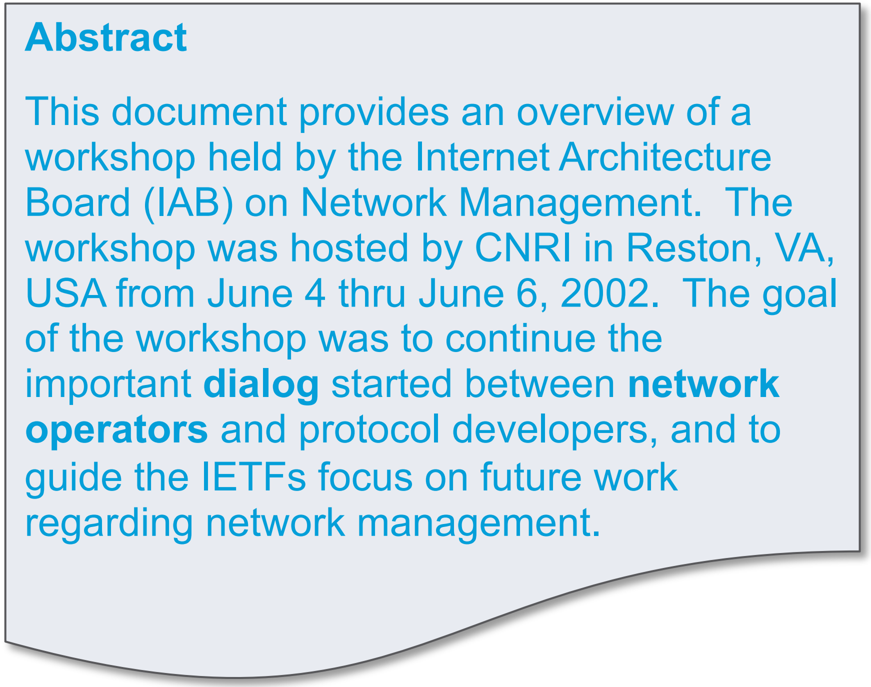
© 2018 Cisco and/or its affiliates. All rights reserved. Cisco Public

The Need for Something Better

- **SNMP had failed**
  - For configuration, that is
  - Extensive use in fault handling and monitoring
- CLI scripting
  - “Market share” 70%+



RFC 3535

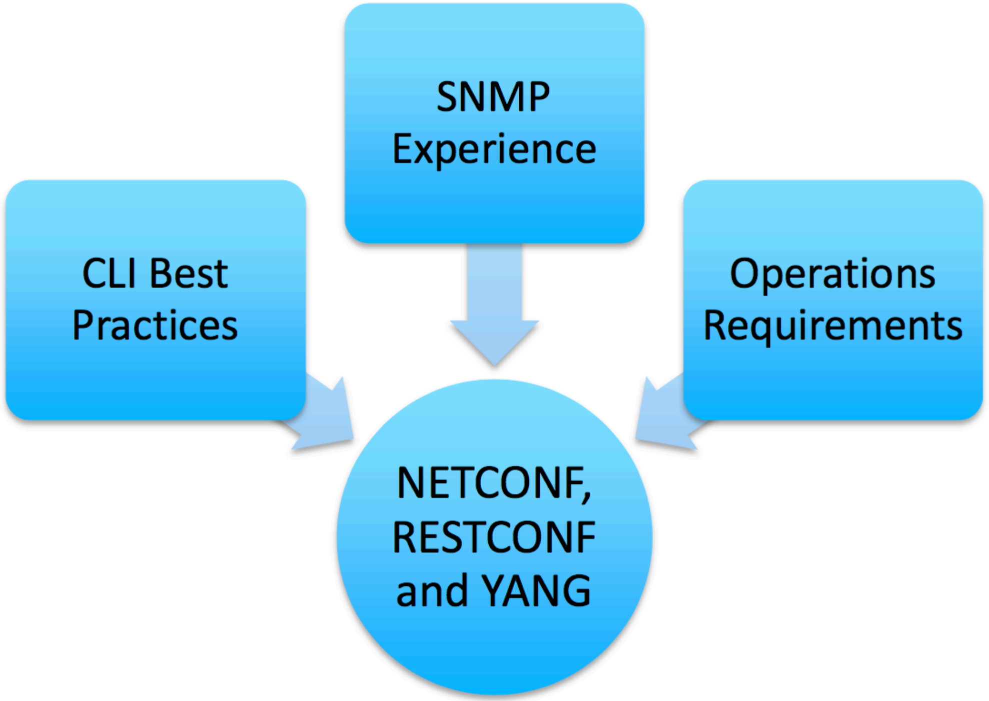


Abstract

This document provides an overview of a workshop held by the Internet Architecture Board (IAB) on Network Management. The workshop was hosted by CNRI in Reston, VA, USA from June 4 thru June 6, 2002. The goal of the workshop was to continue the important **dialog** started between **network operators** and protocol developers, and to guide the IETFs focus on future work regarding network management.

© 2018 Cisco and/or its affiliates. All rights reserved. Cisco Public

Best Practices Coming Together



© 2018 Cisco and/or its affiliates. All rights reserved. Cisco Public

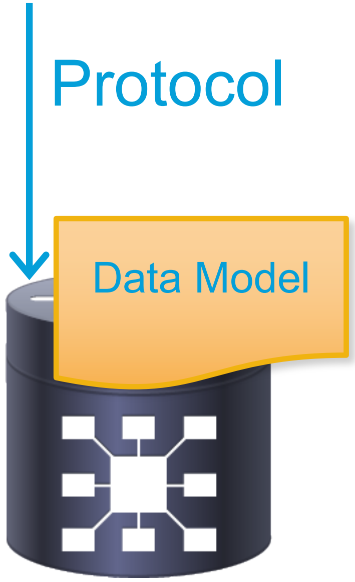
YANG

© 2018 Cisco and/or its affiliates. All rights reserved. Cisco Public

YANG

Data Modeling Language for Networking

- Modeling language, YANG version 1 [RFC6020], YANG version 1.1 [RFC7950]
- Models configuration and state data, RPCs, and notifications
- Defines semantics
  - Constraints (i.e. “MUSTs”)
  - Reusable structures
  - Built-in and derived types

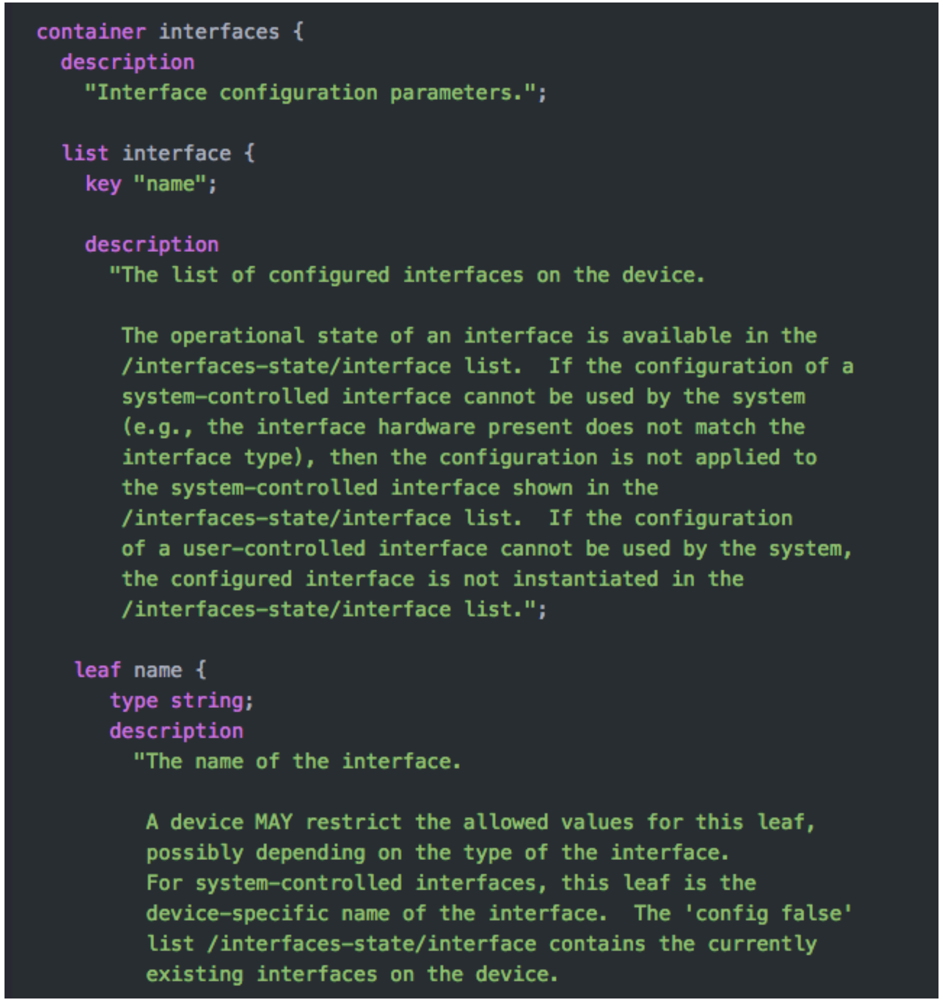


YANG is a full, formal contract language with rich syntax and semantics for network data

© 2018 Cisco and/or its affiliates. All rights reserved. Cisco Public

YANG Model Example

- Screenshot from ietf-interfaces.yang
- Container 'interfaces' with list of interface' items
- List items (leafs) have a 'name' which is also the key for the list

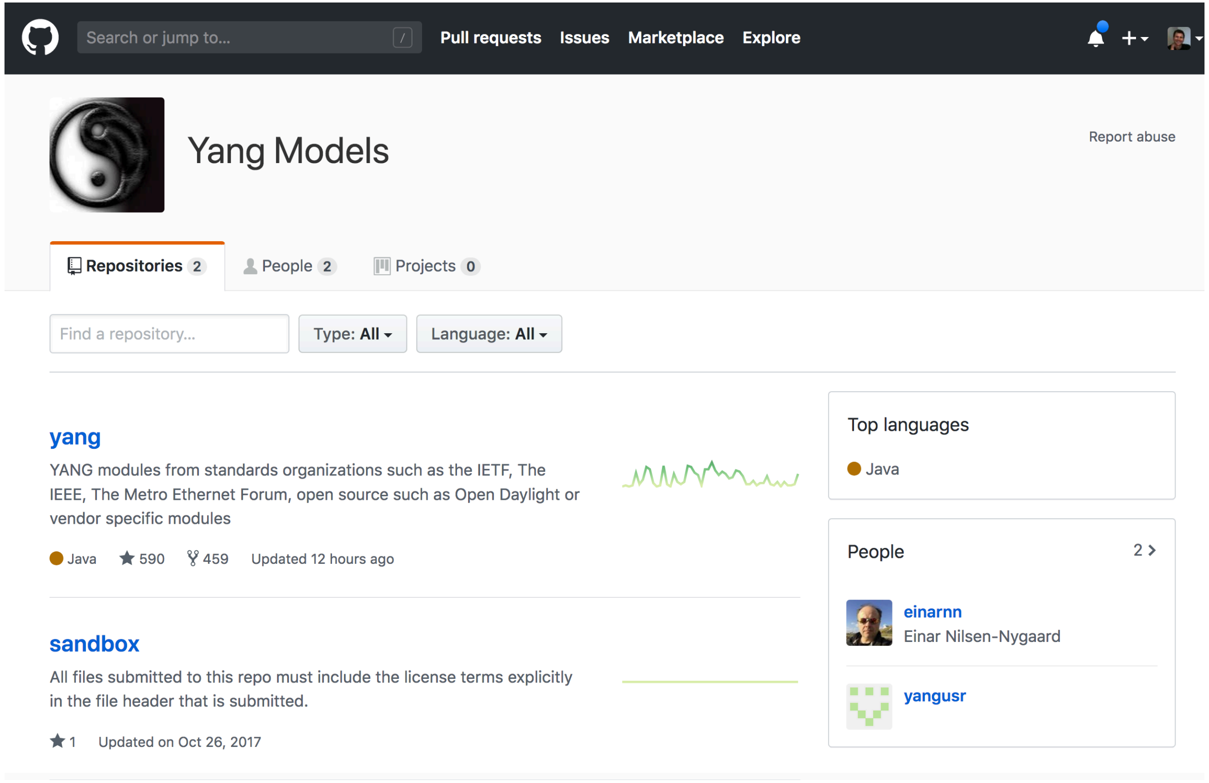


```
container interfaces {
  description
    "Interface configuration parameters.";

  list interface {
    key "name";

    description
      "The list of configured interfaces on the device.

       The operational state of an interface is available in the
       /interfaces-state/interface list.  If the configuration of a
       system-controlled interface cannot be used by the system
       (e.g., the interface hardware present does not match the
       interface type), then the configuration is not applied to
       the system-controlled interface shown in the
       /interfaces-state/interface list.  If the configuration
       of a user-controlled interface cannot be used by the system,
       the configured interface is not instantiated in the
       /interfaces-state/interface list.";

    leaf name {
      type string;
      description
        "The name of the interface.

         A device MAY restrict the allowed values for this leaf,
         possibly depending on the type of the interface.
         For system-controlled interfaces, this leaf is the
         device-specific name of the interface.  The 'config false'
         list /interfaces-state/interface contains the currently
         existing interfaces on the device.
```

© 2018 Cisco and/or its affiliates. All rights reserved. Cisco Public

Finding YANG Models

https://github.com/YangModels/



Yang Models

yang

YANG modules from standards organizations such as the IETF, The IEEE, The Metro Ethernet Forum, open source such as Open Daylight or vendor specific modules

● Java ★ 590 459 Updated 12 hours ago

sandbox

All files submitted to this repo must include the license terms explicitly in the file header that is submitted.

★ 1 Updated on Oct 26, 2017

Top languages

● Java

People 2

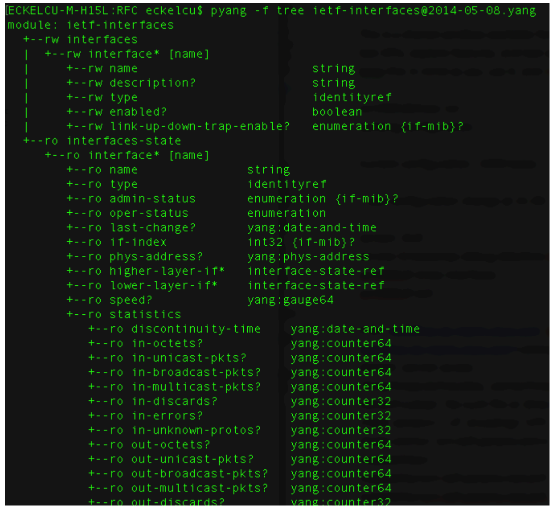
einarnn Einar Nilsen-Nygaard

yangusr

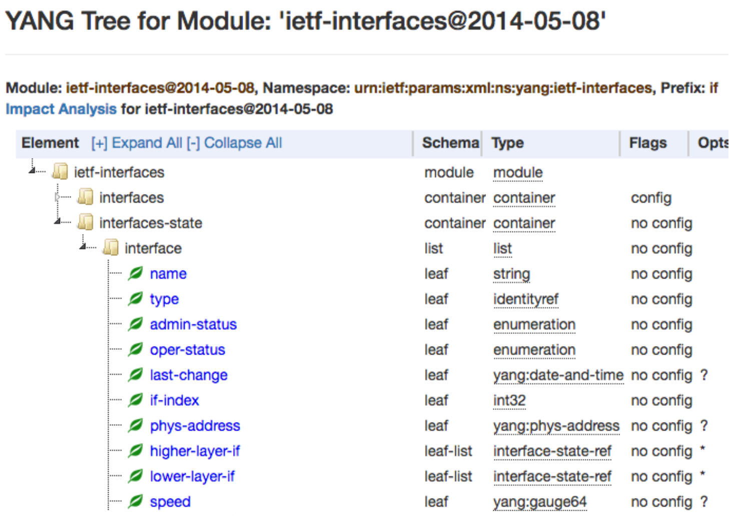
© 2018 Cisco and/or its affiliates. All rights reserved. Cisco Public

Tools to work with YANG Models

- pyang - An extensible YANG validator and converter
  - Command line tool
  - Source Code - https://github.com/mbj4668/pyang
  - Python Package - https://pypi.python.org/pypi/pyang
- YANG Catalog - YANG validator, search, and impact tools
  - Web Based
  - https://yangcatalog.org/
- OpenDaylight YANG Tools
  - Tools supporting NETCONF and YANG
  - Code generation from YANG models
  - https://wiki.opendaylight.org/view/YANG_Tools:Main



```
ECKELCU-M-H15L:RFC eckelcu$ pyang -f tree ietf-interfaces@2014-05-08.yang
module: ietf-interfaces
  +--rw interfaces
  |  +--rw interface* [name]
  |     +--rw name                        string
  |     +--rw description?                string
  |     +--rw type                        identityref
  |     +--rw enabled?                    boolean
  |     +--rw link-up-down-trap-enable?   enumeration {if-mib}?
  +--ro interfaces-state
     +--ro interface* [name]
        +--ro name               string
        +--ro type               identityref
        +--ro admin-status       enumeration {if-mib}?
        +--ro oper-status        enumeration
        +--ro last-change?       yang:date-and-time
        +--ro if-index           int32 {if-mib}?
        +--ro phys-address?      yang:phys-address
        +--ro higher-layer-if*   interface-state-ref
        +--ro lower-layer-if*    interface-state-ref
        +--ro speed?             yang:gauge64
        +--ro statistics
           +--ro discontinuity-time    yang:date-and-time
           +--ro in-octets?            yang:counter64
           +--ro in-unicast-pkts?      yang:counter64
           +--ro in-broadcast-pkts?    yang:counter64
           +--ro in-multicast-pkts?    yang:counter64
           +--ro in-discards?          yang:counter32
           +--ro in-errors?            yang:counter32
           +--ro in-unknown-protos?    yang:counter32
           +--ro out-octets?           yang:counter64
           +--ro out-unicast-pkts?     yang:counter64
           +--ro out-broadcast-pkts?   yang:counter64
           +--ro out-multicast-pkts?   yang:counter64
           +--ro out-discards?         yang:counter32
```



YANG Tree for Module: 'ietf-interfaces@2014-05-08'

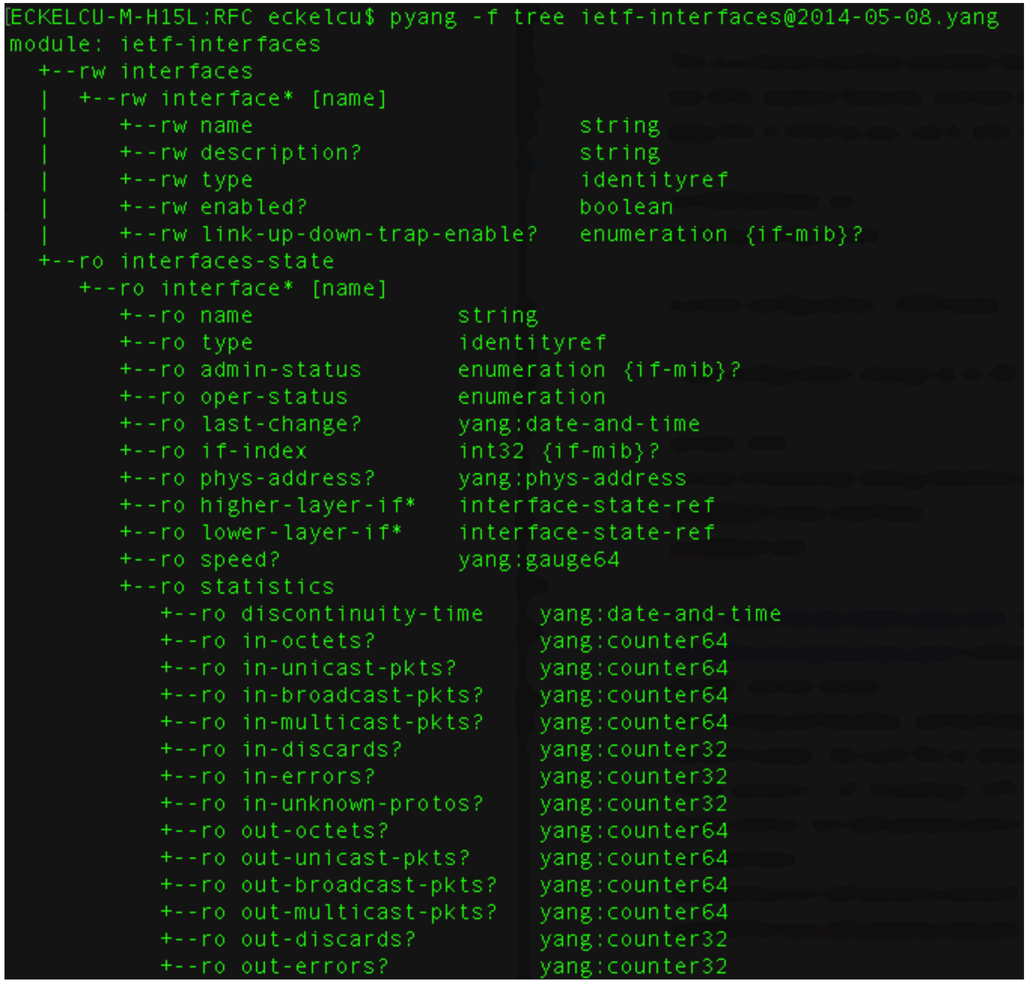
Module: ietf-interfaces@2014-05-08, Namespace: urn:ietf:params:xml:ns:yang:ietf-interfaces, Prefix: if
Impact Analysis for ietf-interfaces@2014-05-08

Element [+] Expand All [-] Collapse All	Schema	Type	Flags	Opts
ietf-interfaces	module	module		
interfaces	container	container	config	
interfaces-state	container	container	no config	
interface	list	list	no config	
name	leaf	string	no config	
type	leaf	identityref	no config	
admin-status	leaf	enumeration	no config	
oper-status	leaf	enumeration	no config	
last-change	leaf	yang:date-and-time	no config	?
if-index	leaf	int32	no config	
phys-address	leaf	yang:phys-address	no config	?
higher-layer-if	leaf-list	interface-state-ref	no config	*
lower-layer-if	leaf-list	interface-state-ref	no config	*
speed	leaf	yang:gauge64	no config	?

DEVNET

pyang

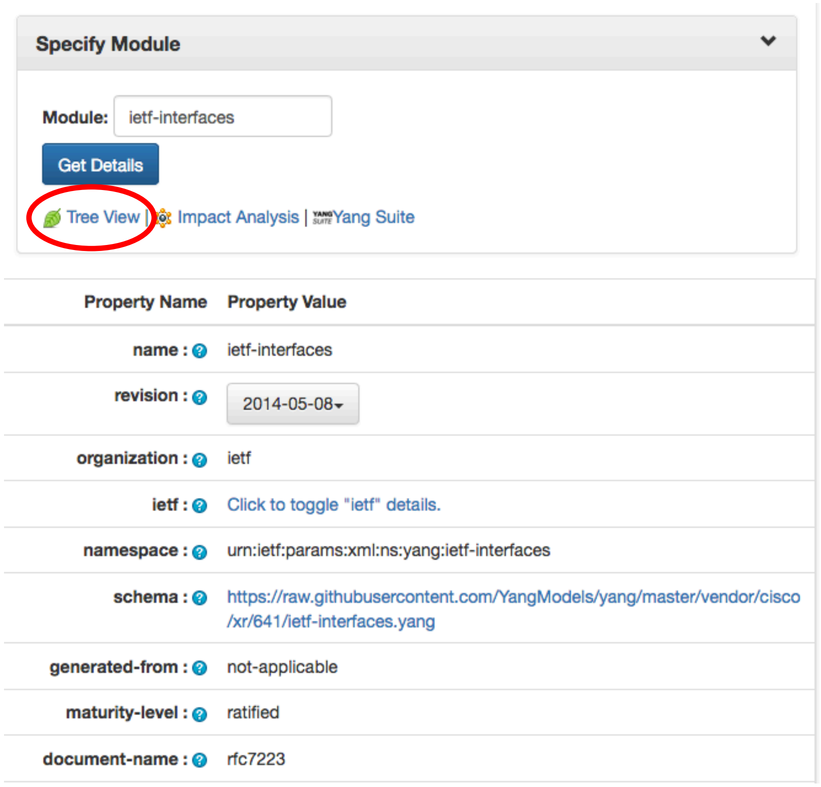
```
$ pyang -f tree
<yang-file>
```



```
[ECKELCU-M-H15L:RFC eckelcu$ pyang -f tree ietf-interfaces@2014-05-08.yang
module: ietf-interfaces
   +--rw interfaces
   |  +--rw interface* [name]
   |     +--rw name                       string
   |     +--rw description?               string
   |     +--rw type                       identityref
   |     +--rw enabled?                   boolean
   |     +--rw link-up-down-trap-enable?  enumeration {if-mib}?
   +--ro interfaces-state
      +--ro interface* [name]
         +--ro name               string
         +--ro type               identityref
         +--ro admin-status       enumeration {if-mib}?
         +--ro oper-status        enumeration
         +--ro last-change?       yang:date-and-time
         +--ro if-index           int32 {if-mib}?
         +--ro phys-address?      yang:phys-address
         +--ro higher-layer-if*   interface-state-ref
         +--ro lower-layer-if*    interface-state-ref
         +--ro speed?             yang:gauge64
         +--ro statistics
            +--ro discontinuity-time    yang:date-and-time
            +--ro in-octets?            yang:counter64
            +--ro in-unicast-pkts?      yang:counter64
            +--ro in-broadcast-pkts?    yang:counter64
            +--ro in-multicast-pkts?    yang:counter64
            +--ro in-discards?          yang:counter32
            +--ro in-errors?            yang:counter32
            +--ro in-unknown-protos?    yang:counter32
            +--ro out-octets?           yang:counter64
            +--ro out-unicast-pkts?     yang:counter64
            +--ro out-broadcast-pkts?   yang:counter64
            +--ro out-multicast-pkts?   yang:counter64
            +--ro out-discards?         yang:counter32
            +--ro out-errors?           yang:counter32
```

Yang Catalog

https://yangcatalog.org/yang-search/



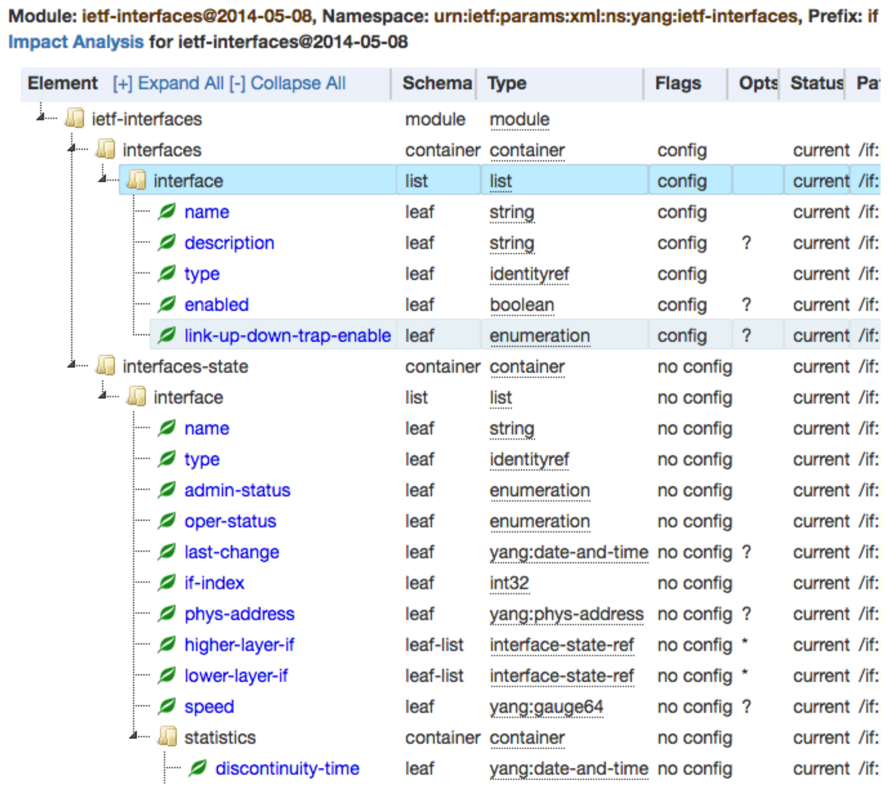
Specify Module

Module: ietf-interfaces

Get Details

Tree View | Impact Analysis | Yang Suite

Property Name	Property Value
name :	ietf-interfaces
revision :	2014-05-08
organization :	ietf
ietf :	Click to toggle "ietf" details.
namespace :	urn:ietf:params:xml:ns:yang:ietf-interfaces
schema :	https://raw.githubusercontent.com/YangModels/yang/master/vendor/cisco/xr/641/ietf-interfaces.yang
generated-from :	not-applicable
maturity-level :	ratified
document-name :	rfc7223



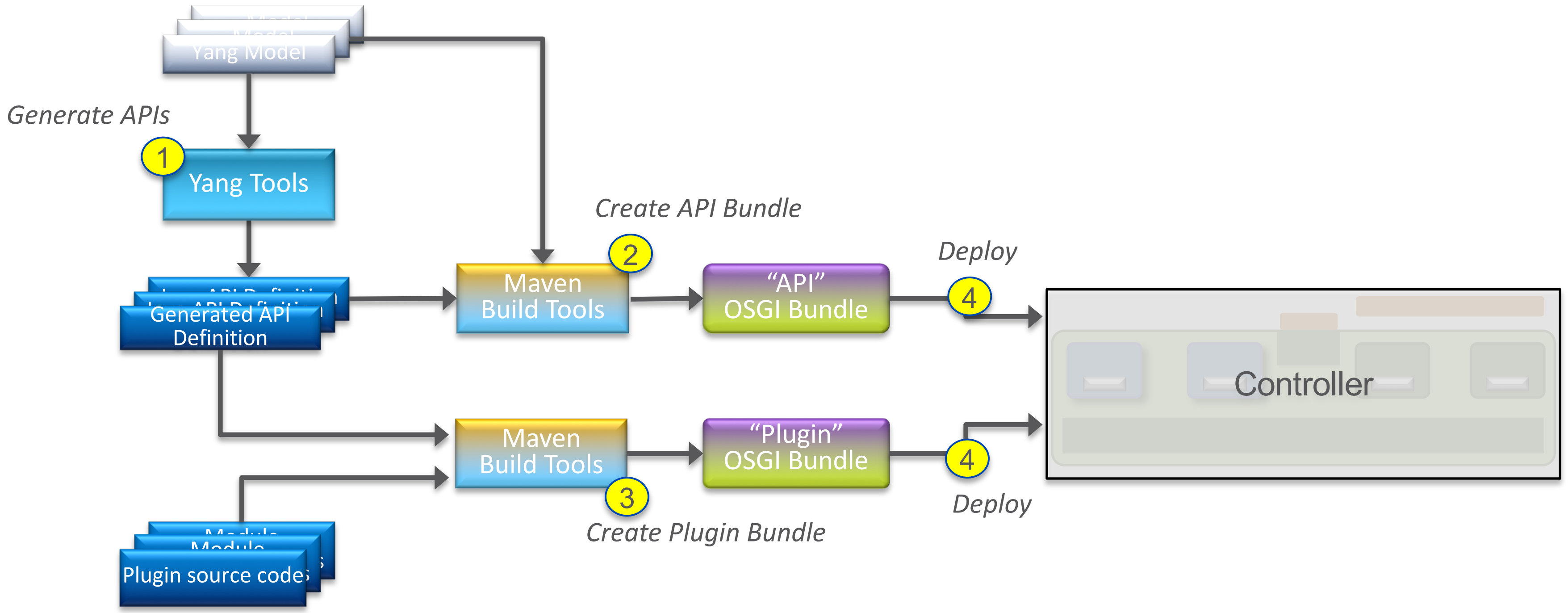
Module: ietf-interfaces@2014-05-08, Namespace: urn:ietf:params:xml:ns:yang:ietf-interfaces, Prefix: if

Impact Analysis for ietf-interfaces@2014-05-08

Element [+] Expand All [-] Collapse All	Schema	Type	Flags	Opts	Status	Pa
ietf-interfaces	module	module				
interfaces	container	container	config		current	/if:
interface	list	list	config		current	/if:
name	leaf	string	config		current	/if:
description	leaf	string	config	?	current	/if:
type	leaf	identityref	config		current	/if:
enabled	leaf	boolean	config	?	current	/if:
link-up-down-trap-enable	leaf	enumeration	config	?	current	/if:
interfaces-state	container	container	no config		current	/if:
interface	list	list	no config		current	/if:
name	leaf	string	no config		current	/if:
type	leaf	identityref	no config		current	/if:
admin-status	leaf	enumeration	no config		current	/if:
oper-status	leaf	enumeration	no config		current	/if:
last-change	leaf	yang:date-and-time	no config	?	current	/if:
if-index	leaf	int32	no config		current	/if:
phys-address	leaf	yang:phys-address	no config	?	current	/if:
higher-layer-if	leaf-list	interface-state-ref	no config	*	current	/if:
lower-layer-if	leaf-list	interface-state-ref	no config	*	current	/if:
speed	leaf	yang:gauge64	no config	?	current	/if:
statistics	container	container	no config		current	/if:
discontinuity-time	leaf	yang:date-and-time	no config		current	/if:

DEVNET

Building a Plugin/Application with OpenDaylight YANG tools



© 2018 Cisco and/or its affiliates. All rights reserved. Cisco Public

NETCONF

© 2018 Cisco and/or its affiliates. All rights reserved. Cisco Public

NETCONF

IETF network management protocol

- Defined in RFC 4741 (2006), updated by RFC 6241 (2011)
- Connection oriented, with transport via SSH/TSL
- Data defined by YANG models, encoded in XML
- Distinguishes between configuration and state data
- Multiple configuration datastores (candidate, running, startup)
- Change validation, transactions, filtering, and notifications

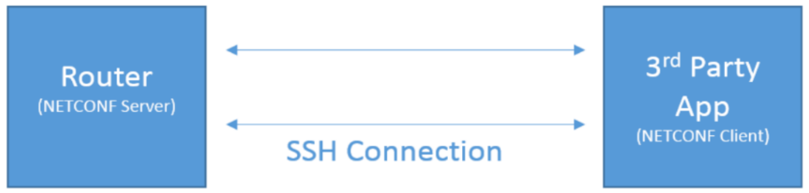
NETCONF provides fundamental programming features for convenient and robust automation of network services

© 2018 Cisco and/or its affiliates. All rights reserved. Cisco Public

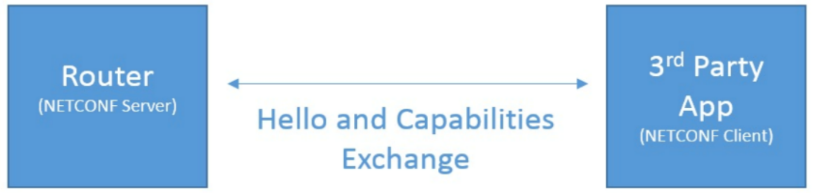
NETCONF Sessions

- NETCONF is connection-oriented
  - SSH, TLS as underlying transport
  - XML for payload
- NETCONF client establishes session with server
- Session establishment: <hello> exchange
  - Announce capabilities, modules, features
- Session termination
  - <close-session>, <kill-session>

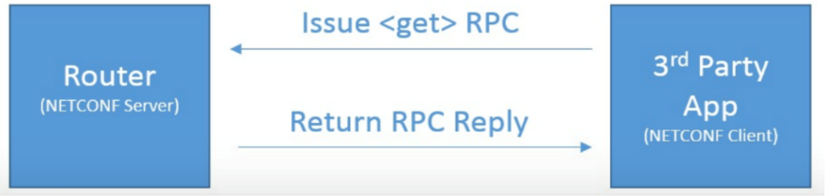
1. The NETCONF client establishes an SSH session to the NETCONF server.



2. The NETCONF client and server exchange NETCONF **hello** messages to exchange capabilities.



3. Now that the NETCONF client and server have exchanged **hello** messages, the client may issue an RPC. In this scenario, the client sends a **get** operation and the server responds with operational data. Note that the **get** operational should be filtered for specific data. Filters are built using XML.



© 2018 Cisco and/or its affiliates. All rights reserved. Cisco Public

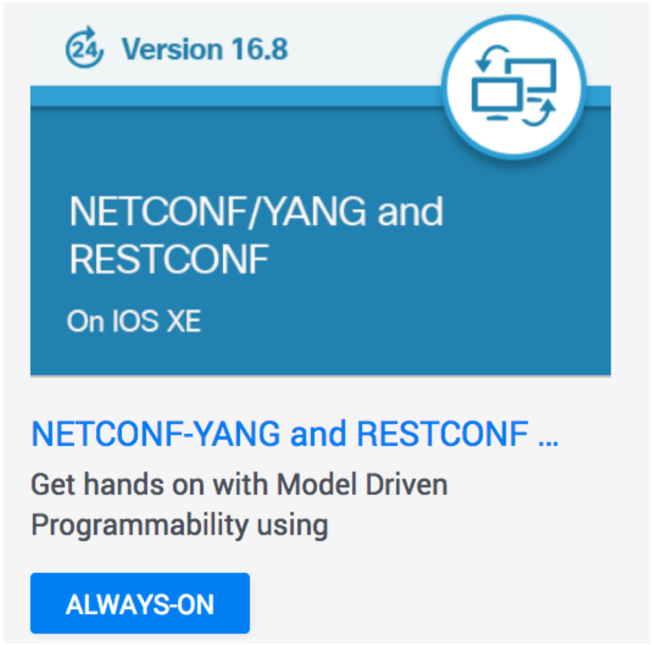
NETCONF Commands

- get : to retrieve operational data
- get-config : to retrieve configuration data
- edit-config : to edit a device configuration
- copy-config : to copy a configuration to another data store (e.g. non-volatile memory)
- delete-config : to delete a configuration in a data store

© 2018 Cisco and/or its affiliates. All rights reserved. Cisco Public

DevNet Always On Sandbox

- CSR1000V Host : ios-xe-mgmt.cisco.com
  - SSH Port: 8181
  - NETCONF Port: 10000
  - RESTCONF Port : 9443 (HTTPS)
- Credentials:
  - Username: **root**
  - Password: **D_Vay!_10&**


Version 16.8
NETCONF/YANG and RESTCONF
On IOS XE
NETCONF-YANG and RESTCONF ...
Get hands on with Model Driven Programmability using
ALWAYS-ON

© 2018 Cisco and/or its affiliates. All rights reserved. Cisco Public

Connect to DevNet Always on Sandbox

ssh root@ios-xe-mgmt.cisco.com -p 8181
ssh -oHostKeyAlgorithms=+ssh-dss root@ios-xe-mgmt.cisco.com -p 10000 -s netconf

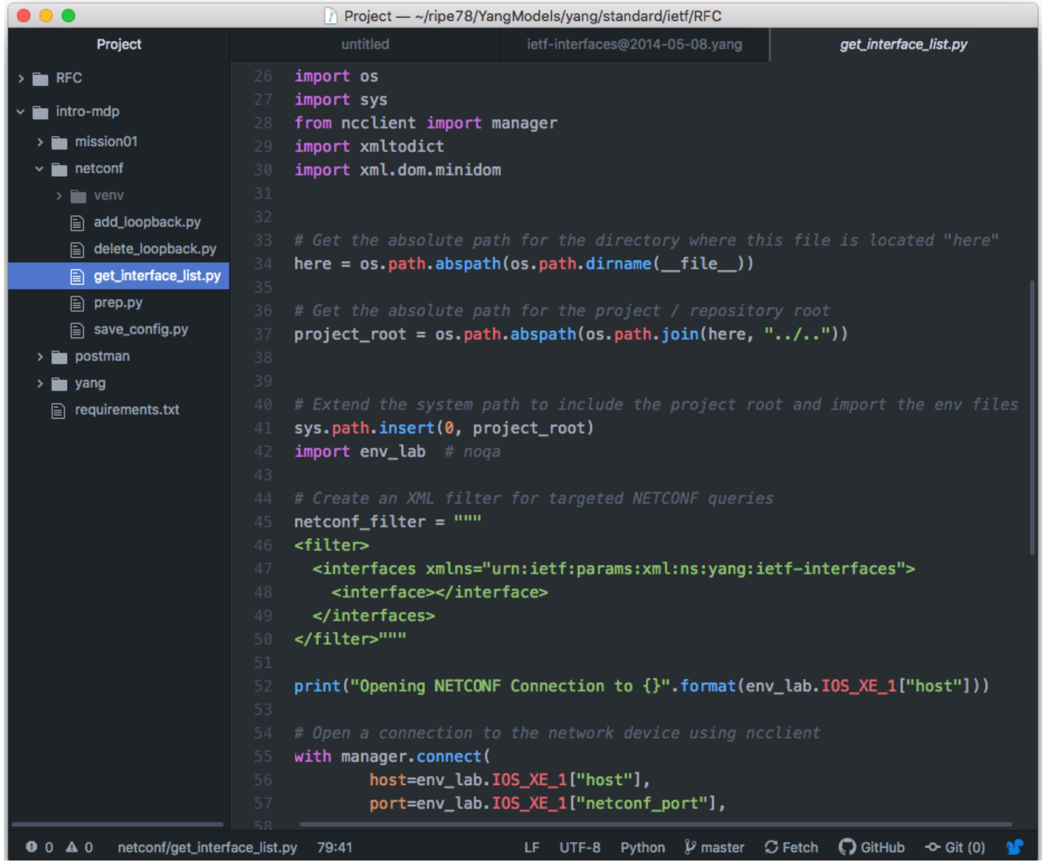
```
ECKELCU-M-H15L:ripe78 eckelcu$ ssh root@ios-xe-mgmt.cisco.com -p 8181
Password:

Welcome to the DevNet Always On Sandbox for IOS XE

This is a shared sandbox available for anyone to use to
test APIs, explore features, and test scripts.  Please
keep this in mind as you use it, and respect others use.

The following programmability features are already enabled:
  - NETCONF
  - RESTCONF

Thanks for stopping by.


csr1000v#show run
Building configuration...

Current configuration : 5332 bytes
!
! Last configuration change at 16:55:51 UTC Fri May 17 2019 by root
!
version 16.8
service timestamps debug datetime msec
service timestamps log datetime msec
platform qfp utilization monitor load 80
no platform punt-keepalive disable-kernel-core
platform console virtual
!
hostname csr1000v
```

```
ECKELCU-M-H15L:ripe78 eckelcu$ ssh -oHostKeyAlgorithms=+ssh-dss
root@ios-xe-mgmt.cisco.com -p 10000 -s netconf
root@ios-xe-mgmt.cisco.com's password:
<?xml version="1.0" encoding="UTF-8"?>
<hello xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
<capabilities>
<capability>urn:ietf:params:netconf:base:1.0</capability>
<capability>urn:ietf:params:netconf:base:1.1</capability>
<capability>urn:ietf:params:netconf:capability:writable-
running:1.0</capability>
<capability>urn:ietf:params:netconf:capability:xpath:1.0</capabil
ity>
<capability>urn:ietf:params:netconf:capability:validate:1.0</capa
bility>
<capability>urn:ietf:params:netconf:capability:validate:1.1</capa
bility>
<capability>urn:ietf:params:netconf:capability:rollback-on-
error:1.0</capability>
<capability>urn:ietf:params:netconf:capability:notification:1.0</
capability>
<capability>urn:ietf:params:netconf:capability:interleave:1.0</ca
pability>
<capability>urn:ietf:params:netconf:capability:with-
defaults:1.0?basic-mode=explicit&amp;also-supported=report-all-
tagged</capability>
<capability>urn:ietf:params:netconf:capability:yang-
library:1.0?revision=2016-06-21&amp;module-set-
id=88c694c75e847aba17e8ab19254ad090</capability>
<capability>http://tail-f.com/ns/netconf/actions/1.0</capability>
<capability>http://tail-f.com/ns/netconf/extensions</capability>
<capability>http://cisco.com/ns/cisco-xe-ietf-ip-deviation?module
```

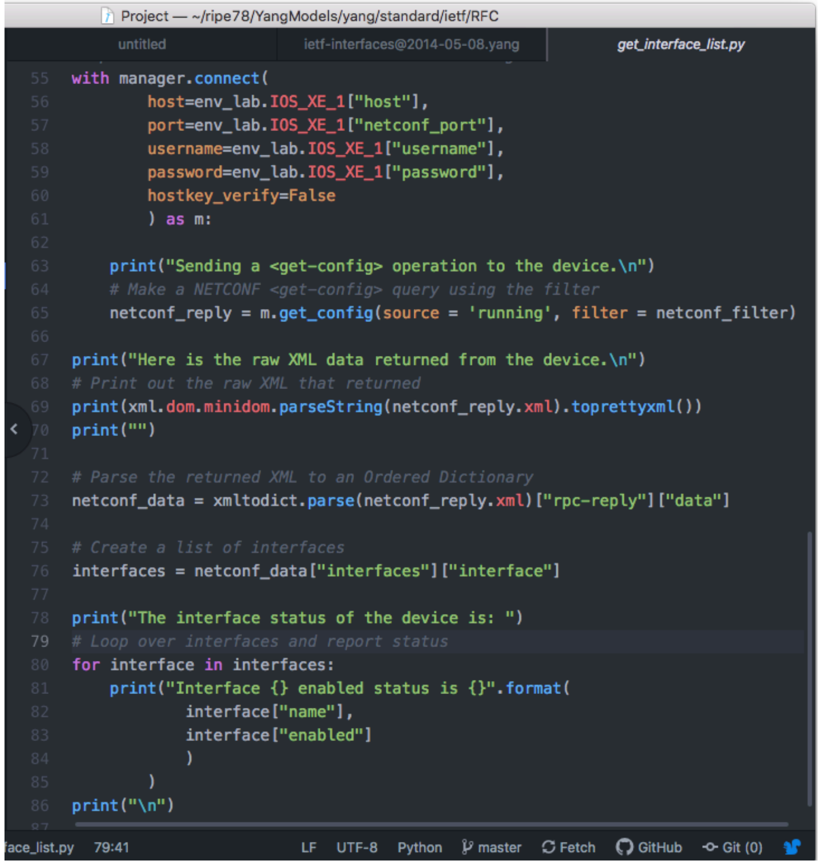
© 2018 Cisco and/or its affiliates. All rights reserved. Cisco Public

NETCONF using ncclient – Python code

https://developer.cisco.com/learning/modules/intro-device-level-interfaces/intro-netconf/step/1



```python
import os
import sys
from ncclient import manager
import xmltodict
import xml.dom.minidom


# Get the absolute path for the directory where this file is located "here"
here = os.path.abspath(os.path.dirname(__file__))

# Get the absolute path for the project / repository root
project_root = os.path.abspath(os.path.join(here, "../.."))


# Extend the system path to include the project root and import the env files
sys.path.insert(0, project_root)
import env_lab  # noqa

# Create an XML filter for targeted NETCONF queries
netconf_filter = """
<filter>
  <interfaces xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces">
    <interface></interface>
  </interfaces>
</filter>"""

print("Opening NETCONF Connection to {}".format(env_lab.IOS_XE_1["host"]))

# Open a connection to the network device using ncclient
with manager.connect(
        host=env_lab.IOS_XE_1["host"],
        port=env_lab.IOS_XE_1["netconf_port"],
```



Project — ~/ripe78/YangModels/yang/standard/ietf/RFC
untitled
ietf-interfaces@2014-05-08.yang
get_interface_list.py
face_list.py 79:41 LF UTF-8 Python master Fetch GitHub Git (0)

```python
with manager.connect(
        host=env_lab.IOS_XE_1["host"],
        port=env_lab.IOS_XE_1["netconf_port"],
        username=env_lab.IOS_XE_1["username"],
        password=env_lab.IOS_XE_1["password"],
        hostkey_verify=False
        ) as m:

    print("Sending a <get-config> operation to the device.\n")
    # Make a NETCONF <get-config> query using the filter
    netconf_reply = m.get_config(source = 'running', filter = netconf_filter)

print("Here is the raw XML data returned from the device.\n")
# Print out the raw XML that returned
print(xml.dom.minidom.parseString(netconf_reply.xml).toprettyxml())
print("")

# Parse the returned XML to an Ordered Dictionary
netconf_data = xmltodict.parse(netconf_reply.xml)["rpc-reply"]["data"]

# Create a list of interfaces
interfaces = netconf_data["interfaces"]["interface"]

print("The interface status of the device is: ")
# Loop over interfaces and report status
for interface in interfaces:
    print("Interface {} enabled status is {}".format(
            interface["name"],
            interface["enabled"]
            )
        )
print("\n")
```

NETCONF using ncclient - Output

```
(venv) ECKELCU-M-H15L:netconf eckelcu$ python get_interface_list.py
Opening NETCONF Connection to ios-xe-mgmt.cisco.com

Sending a <get-config> operation to the device.

Here is the raw XML data returned from the device.

<?xml version="1.0" ?>
<rpc-reply message-id="urn:uuid:10be2e92-4093-4307-8e80-e13c55b896ed" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0">
          <data>
                    <interfaces xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces">
                              <interface>
                                        <name>GigabitEthernet1</name>
                                        <description>DON'T TOUCH ME</description>
--- snip ---
                              <interface>
                                        <name>Tunnel2</name>
                                        <enabled>true</enabled>
                              </interface>
                    </interfaces>
          </data>
</rpc-reply>

The interface status of the device is:
Interface GigabitEthernet1 enabled status is true
Interface GigabitEthernet2 enabled status is true
Interface GigabitEthernet3 enabled status is false
Interface Loopback0 enabled status is true
Interface Tunnel0 enabled status is true
Interface Tunnel1 enabled status is true
Interface Tunnel2 enabled status is true
```

© 2018 Cisco and/or its affiliates. All rights reserved. Cisco Public

RESTCONF

© 2018 Cisco and/or its affiliates. All rights reserved. Cisco Public

RESTCONF

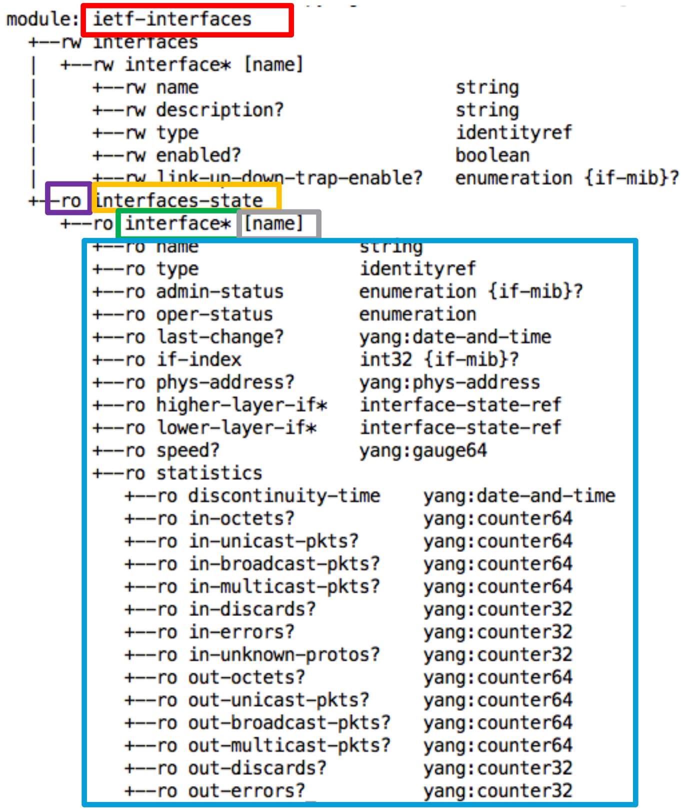
Restful API for YANG data models

- IETF RFC 8040
- Configuration and state data exposed as resources
- Access data using REST verbs (GET / PUT / POST …)
- Construct URIs, based on structure of YANG model, to access data
- HTTP instead of SSH for transport
- JSON in addition to XML for data encoding

RESTCONF provides light weight interface to network datastores leveraging well known combination of REST and JSON

© 2018 Cisco and/or its affiliates. All rights reserved. Cisco Public

RESTCONF URI & JSON Example



```
module: ietf-interfaces
  +--rw interfaces
  |  +--rw interface* [name]
  |     +--rw name                        string
  |     +--rw description?                string
  |     +--rw type                        identityref
  |     +--rw enabled?                    boolean
  |     +--rw link-up-down-trap-enable?   enumeration {if-mib}?
  +--ro interfaces-state
     +--ro interface* [name]
        +--ro name               string
        +--ro type               identityref
        +--ro admin-status       enumeration {if-mib}?
        +--ro oper-status        enumeration
        +--ro last-change?       yang:date-and-time
        +--ro if-index           int32 {if-mib}?
        +--ro phys-address?      yang:phys-address
        +--ro higher-layer-if*   interface-state-ref
        +--ro lower-layer-if*    interface-state-ref
        +--ro speed?             yang:gauge64
        +--ro statistics
           +--ro discontinuity-time    yang:date-and-time
           +--ro in-octets?            yang:counter64
           +--ro in-unicast-pkts?      yang:counter64
           +--ro in-broadcast-pkts?    yang:counter64
           +--ro in-multicast-pkts?    yang:counter64
           +--ro in-discards?          yang:counter32
           +--ro in-errors?            yang:counter32
           +--ro in-unknown-protos?    yang:counter32
           +--ro out-octets?           yang:counter64
           +--ro out-unicast-pkts?     yang:counter64
           +--ro out-broadcast-pkts?   yang:counter64
           +--ro out-multicast-pkts?   yang:counter64
           +--ro out-discards?         yang:counter32
           +--ro out-errors?           yang:counter32
```

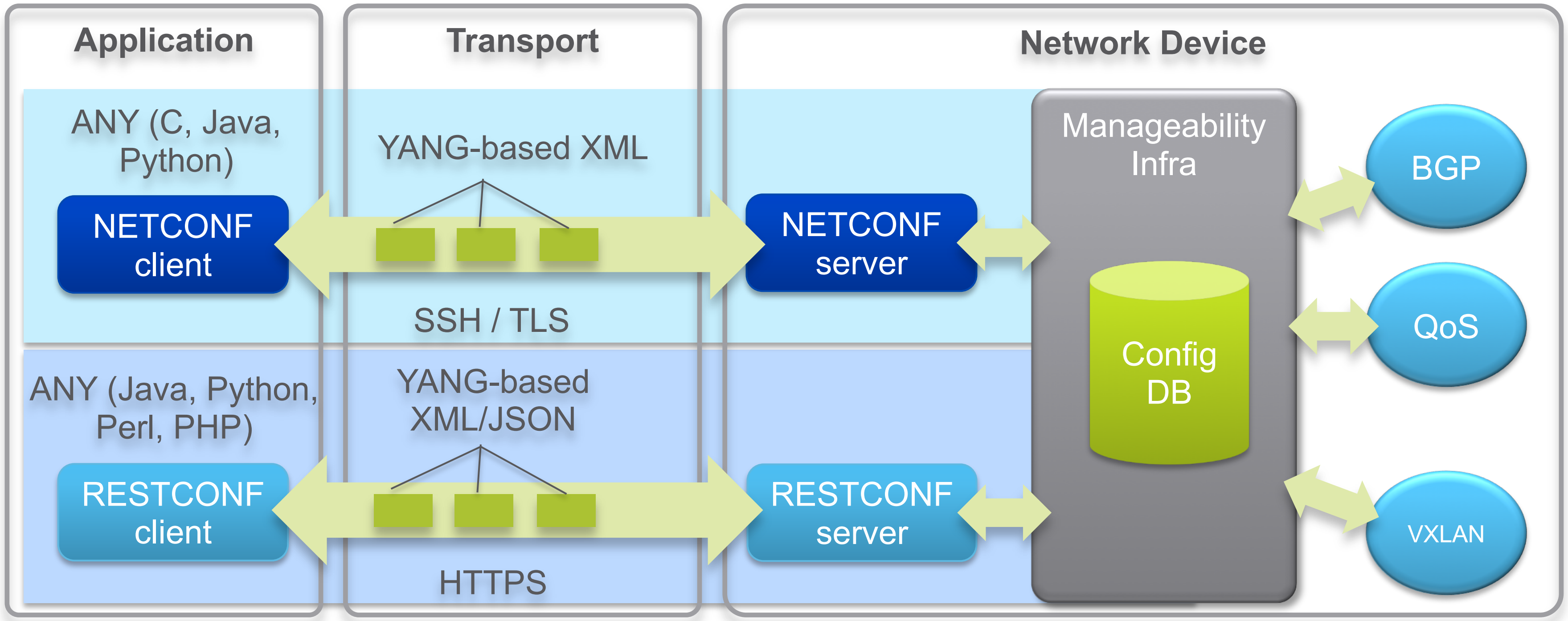
GET

https://{{host}}:{{port}}/restconf/data/ietf-interfaces:interfaces-state/interface=GigabitEthernet1

```
{
  "ietf-interfaces:interface": {
    "name": "GigabitEthernet1",
    "type": "iana-if-type:ethernetCsmacd",
    "admin-status": "up",
    "oper-status": "up",
    "last-change": "2019-05-16T19:40:02.000393+00:00",
    "if-index": 1,
    "phys-address": "00:50:56:bb:18:c4",
    "speed": "1024000000",
    "statistics": {
      "discontinuity-time": "2019-05-16T19:38:03.000573+00:00",
      "in-octets": "5339802",
      "in-unicast-pkts": "48925",
      "in-broadcast-pkts": "0",
      "in-multicast-pkts": "0",
      "in-discards": 0,
      "in-errors": 0,
      "in-unknown-protos": 0,
      "out-octets": "9405098",
      "out-unicast-pkts": "17451",
      "out-broadcast-pkts": "0",
      "out-multicast-pkts": "0",
      "out-discards": 0,
      "out-errors": 0
    }
  }
}
```

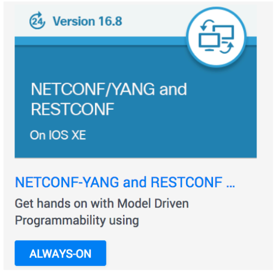
© 2018 Cisco and/or its affiliates. All rights reserved. Cisco Public

High Level Manageability Architecture



© 2018 Cisco and/or its affiliates. All rights reserved. Cisco Public

RESTCONF with curl


Version 16.8

NETCONF/YANG and RESTCONF

On IOS XE

NETCONF-YANG and RESTCONF ...

Get hands on with Model Driven Programmability using

ALWAYS-ON

The Request

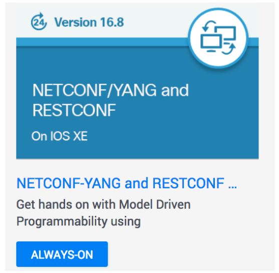
```
$ curl -vk \
  -u root:D_Vay\!_10\& \
  -H 'accept: application/yang-data+json' \
https://ios-xe-mgmt.cisco.com:9443/restconf/data/ietf-interfaces:interfaces/interface=GigabitEthernet1

> GET /restconf/data/ietf-interfaces:interfaces/interface=GigabitEthernet1 HTTP/1.1
> Host: ios-xe-mgmt.cisco.com:9443
> Authorization: Basic cm9vdDpEX1ZheSFfMTAm
> User-Agent: curl/7.54.0
> accept: application/yang-data+json
>
```

- -u provides `user:password` for Basic Authentication
- -H to set headers
- Lines beginning with “>” indicate Request elements
- Lines beginning with “<” indicate Response elements (next slide)

DEVNET

© 2018 Cisco and/or its affiliates. All rights reserved. Cisco Public

RESTCONF with curl


Version 16.8

NETCONF/YANG and RESTCONF

On IOS XE

NETCONF-YANG and RESTCONF ...

Get hands on with Model Driven Programmability using

ALWAYS-ON

The Response - Headers

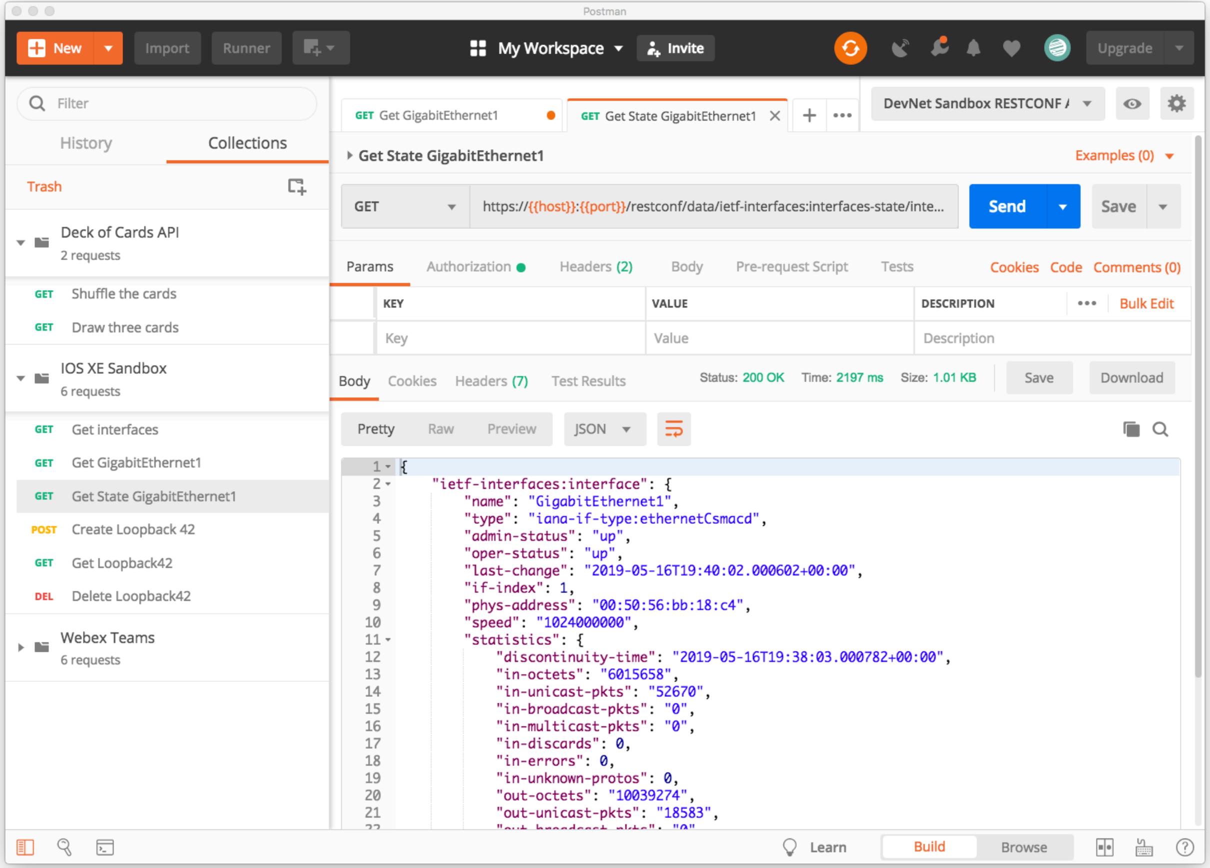
```
< HTTP/1.1 200 OK
< Server: nginx
< Date: Fri, 25 Jan 2019 17:37:43 GMT
< Content-Type: application/yang-data+json
< Transfer-Encoding: chunked
< Connection: close
< Cache-Control: private, no-cache, …
< Pragma: no-cache
<
```

The Response - Body

```
{
  "ietf-interfaces:interface": {
    "name": "GigabitEthernet1",
    "description": "DON'T TOUCH ME",
    "type": "iana-if-type:ethernetCsmacd",
    "enabled": true,
    "ietf-ip:ipv4": {
      "address": [
        {
          "ip": "10.10.20.48",
          "netmask": "255.255.255.0"
        }
      ]
    },
    "ietf-ip:ipv6": {
    }
  }
}
```

© 2018 Cisco and/or its affiliates. All rights reserved. Cisco Public

RESTCONF with Postman



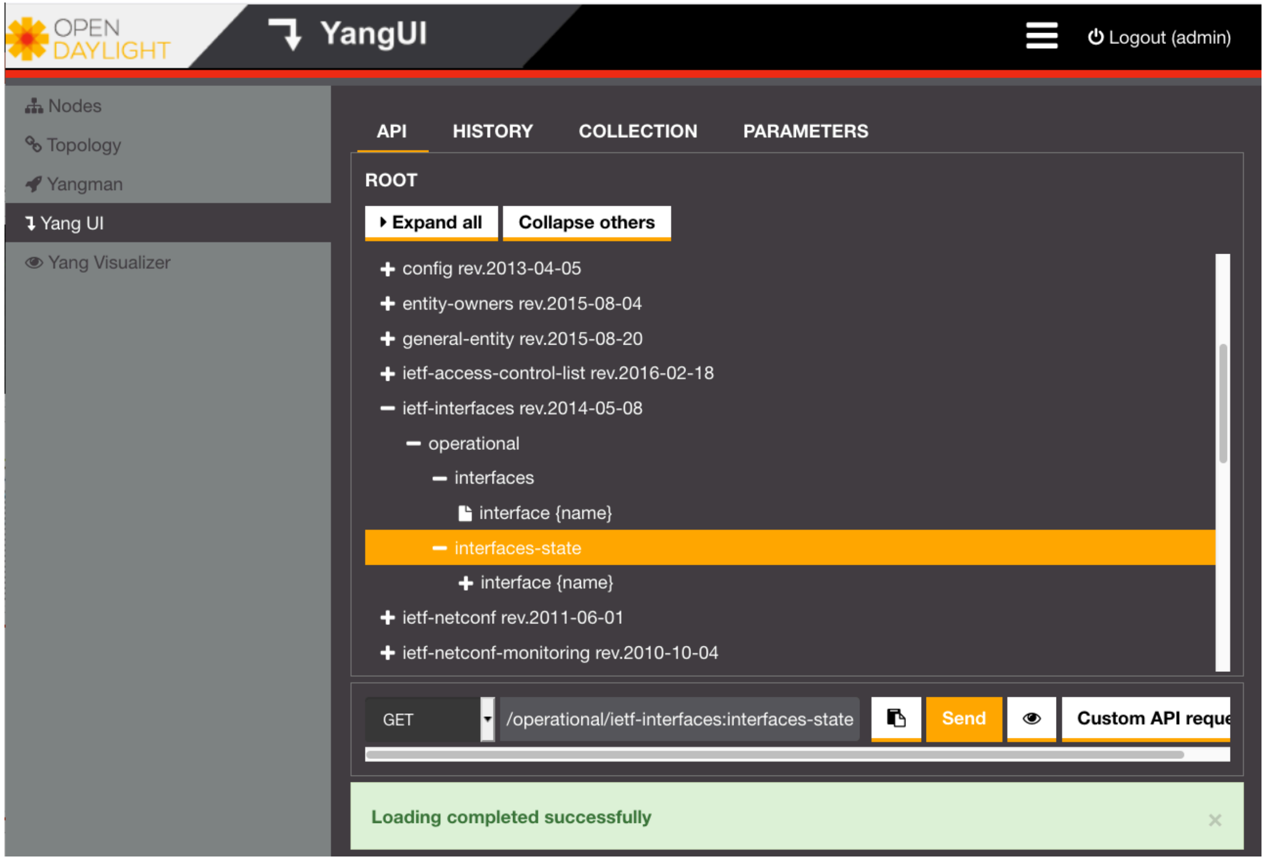
Get State GigabitEthernet1

GET https://{{host}}:{{port}}/restconf/data/ietf-interfaces:interfaces-state/inte...

Status: 200 OK Time: 2197 ms Size: 1.01 KB

```
{
    "ietf-interfaces:interface": {
        "name": "GigabitEthernet1",
        "type": "iana-if-type:ethernetCsmacd",
        "admin-status": "up",
        "oper-status": "up",
        "last-change": "2019-05-16T19:40:02.000602+00:00",
        "if-index": 1,
        "phys-address": "00:50:56:bb:18:c4",
        "speed": "1024000000",
        "statistics": {
            "discontinuity-time": "2019-05-16T19:38:03.000782+00:00",
            "in-octets": "6015658",
            "in-unicast-pkts": "52670",
            "in-broadcast-pkts": "0",
            "in-multicast-pkts": "0",
            "in-discards": 0,
            "in-errors": 0,
            "in-unknown-protos": 0,
            "out-octets": "10039274",
            "out-unicast-pkts": "18583",
            "out-broadcast-pkts": "0",
```

OpenDaylight YANG UI



+ config rev.2013-04-05
+ entity-owners rev.2015-08-04
+ general-entity rev.2015-08-20
+ ietf-access-control-list rev.2016-02-18
− ietf-interfaces rev.2014-05-08
  − operational
    − interfaces
      interface {name}
    − interfaces-state
      + interface {name}
+ ietf-netconf rev.2011-06-01
+ ietf-netconf-monitoring rev.2010-10-04

/operational/ietf-interfaces:interfaces-state

Loading completed successfully

© 2018 Cisco and/or its affiliates. All rights reserved. Cisco Public

Questions?

© 2018 Cisco and/or its affiliates. All rights reserved. Cisco Public

Thank you!