

Open NXOS Programmability

I Have APIs Now?! What Do I Do with Them?

Quinn Snyder, Developer Advocate and Evangelist

@qsnyder

DEVNET-DC



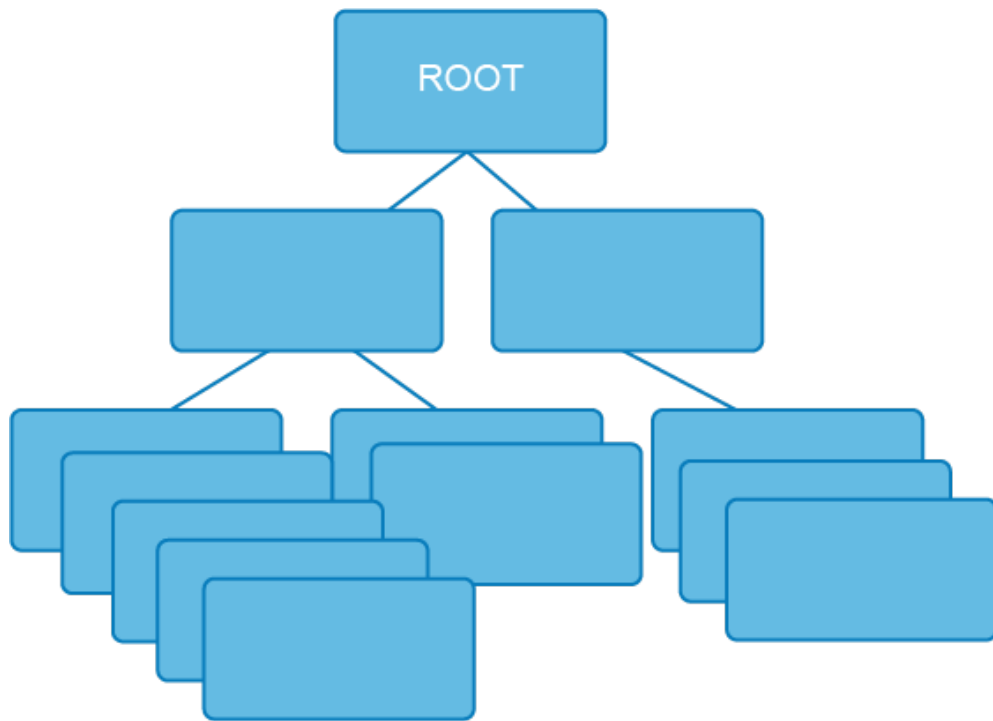


Agenda

- There's a Tree in My Switch
- RESTing with NXAPI
 - REST vs CLI
 - Sandboxes
- Just YANG It!
- Conclusion

There's a Tree in My Switch

NX-OS Management Information Tree (MIT)



An MIT exists within Open NX-OS

Similar concept of MOs, DNs, and classes to ACI

Always builds from root of universe (`sys`)

Used in NXAPI REST and MDT

MIT: There's a Browser for That

Visore exists here too (just in the old color schema)

Searches supported by class or DN; gather URI of call

Similar set of scoping filters supported as in ACI (surprised?)

Object Store Browser

Filter

Class or DN:

Property: Op: Val2:

Run Query

[Display URI of last query](#)

/api/node/mo/sys/bgp.xml?query-target=children

[Display last response](#)

bgpInst	
activateTs	2020-05-13T16:55:01.144+00:00
adminSt	enabled
affGrpActv	0
asPathDbSz	0
asn	65535
attribDbSz	124
childAction	
createTs	2020-05-13T16:54:59.080+00:00
ctrl	fastExtFallover
disPolBatch	disabled
disPolBatchv4PfxLst	

RESTing with NXAPI

NXAPI: Manage Nexus Switches via REST



REST API
GET PUT POST DELETE
(CRUD Operations)



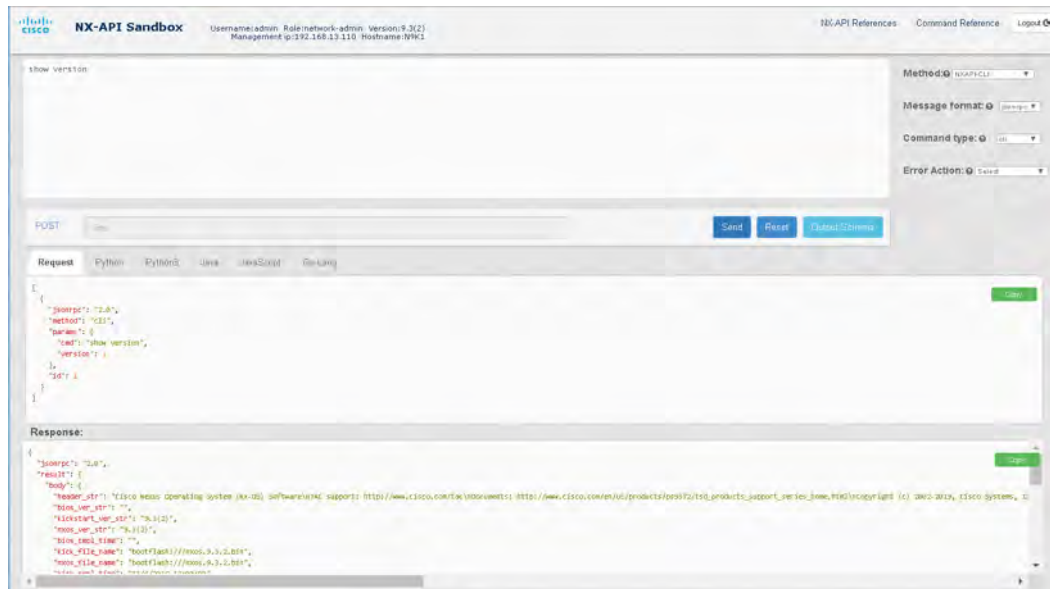
NXAPI: Sandbox for Exploration

- NXOS supports full RESTful API for config and operational data
- Accessible through in-band or mgmt0 interface
- Supported on N3K, N5K, N7K, and N9K
 - Code restrictions apply

Explore both modes:

- DME (NXAPI REST*)
- CLI (NXAPI CLI)

*DME on N3K/N9K only



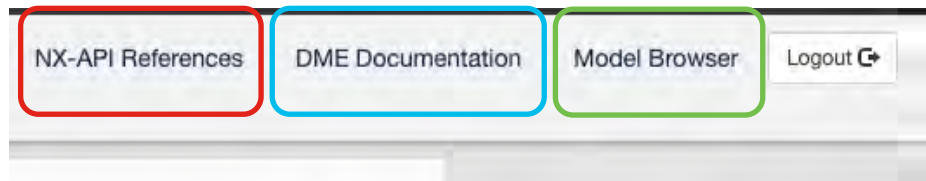
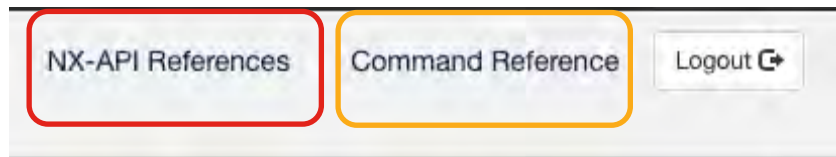
NXAPI: Handy Links

DevNet NX-API docs, aggregated

On-switch reference of output schemas for show commands

DevNet DME reference for MO paths

Launches the Visore browser



NXAPI CLI Method

NXAPI CLI Method

NX-API CLI



CLI Input

```
{  
  "jsonrpc": "2.0",  
  "method": "cli",  
  "params": {  
    "cmd": "show version",  
    "version": 1  
  },  
  "id": 1  
}
```

```
{  
  "jsonrpc": "2.0",  
  "result": {  
    "body": {  
      "kern_uptm_secs": 27,  
      "bios_cmpl_time": "03/02/2014",  
      "bootflash_size": 7906304,  
      "kickstart_ver_str": "7.0 (3) r1 (1) ",  
      "chassis_id": "Nexus9000 c9396PX Chassis",  
      ...  
    }  
  }  
}
```

Response



Nexus 9K

NXAPI CLI Method

CLI

```
show ip interface brief
```



NXAPI CLI

```
HTTP POST: http://sbx-nxos-mgmt.cisco.com/ins  
HEADERS: { 'Content-Type': 'application/json-rpc' }
```

```
[  
  {  
    "jsonrpc": "2.0",  
    "method": "cli",  
    "params": {  
      "cmd": "show ip interface brief",  
      "version": 1  
    },  
    "id": 1  
  }  
]
```


NXAPI CLI: Sandbox Exploration

Keep track of methods and formats; changes request format

All NXAPI CLI use HTTP POST verbs (post CLI into parser)

The screenshot displays the NX-API Sandbox web interface. At the top, it shows the user 'admin' with role 'network-admin' and version '9.3(3)'. The main area is titled 'show ip interface brief'. On the right, a red box highlights the configuration options: 'Method' set to 'NXAPI-CLI', 'Message format' set to 'json-api', 'Command type' set to 'cli', and 'Error Action' set to 'Select'. Below these, a 'POST' request is shown with the path '/ins'. The 'Request' tab is active, displaying a JSON payload. The 'Response' tab is also active, showing a JSON response with a 'result' object containing a 'body' with a 'TABLE_intf' array. The 'Send', 'Reset', and 'Output Schemas' buttons are visible. The 'Request' and 'Response' tabs have 'Copy' buttons next to them.

show ip interface brief

Method: Message format: Command type: Error Action:

POST /ins

Send Reset Output Schemas

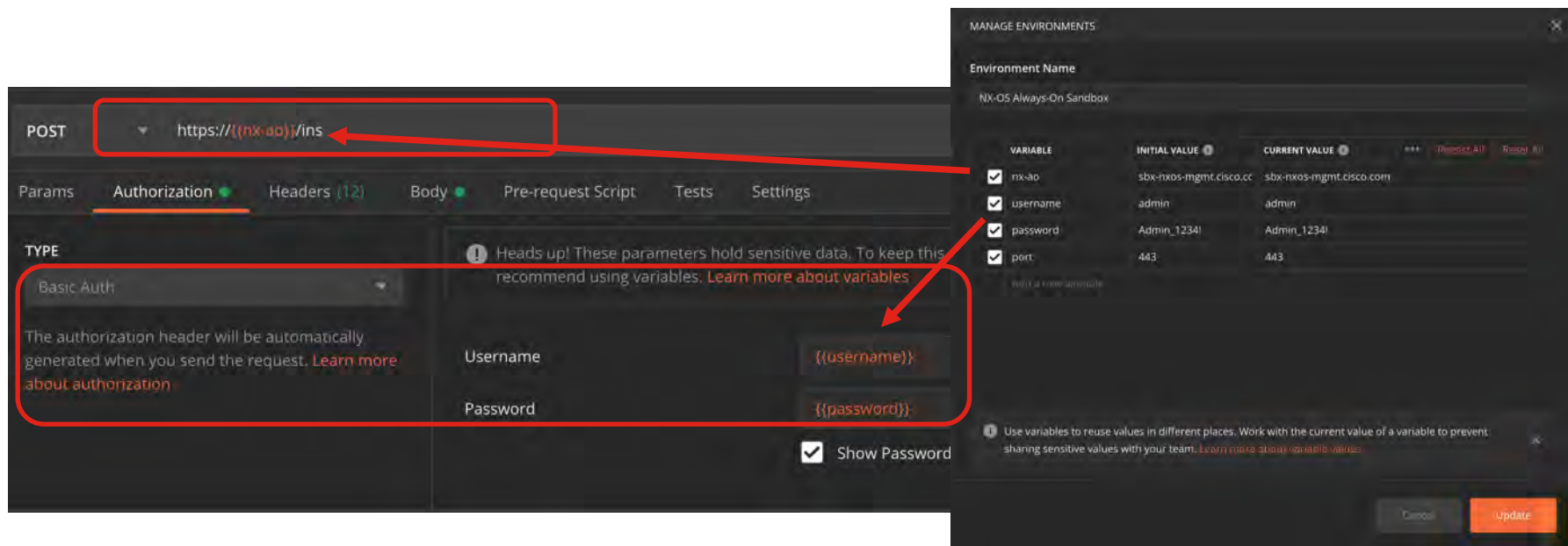
Request Python Python3 Java JavaScript Go-Lang

```
{
  "jsonrpc": "2.0",
  "method": "cli",
  "params": {
    "cmd": "show ip interface brief",
    "version": 1
  },
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "body": {
      "TABLE_intf": {
        "ROW_intf": [
          {
            "vrf-name-out": "default",
            "intf-name": "Vlan100",
            "proto-state": "down",
            "link-state": "down",
            "admin-state": "down",
            "l2d": 1
          }
        ]
      }
    }
  }
}
```

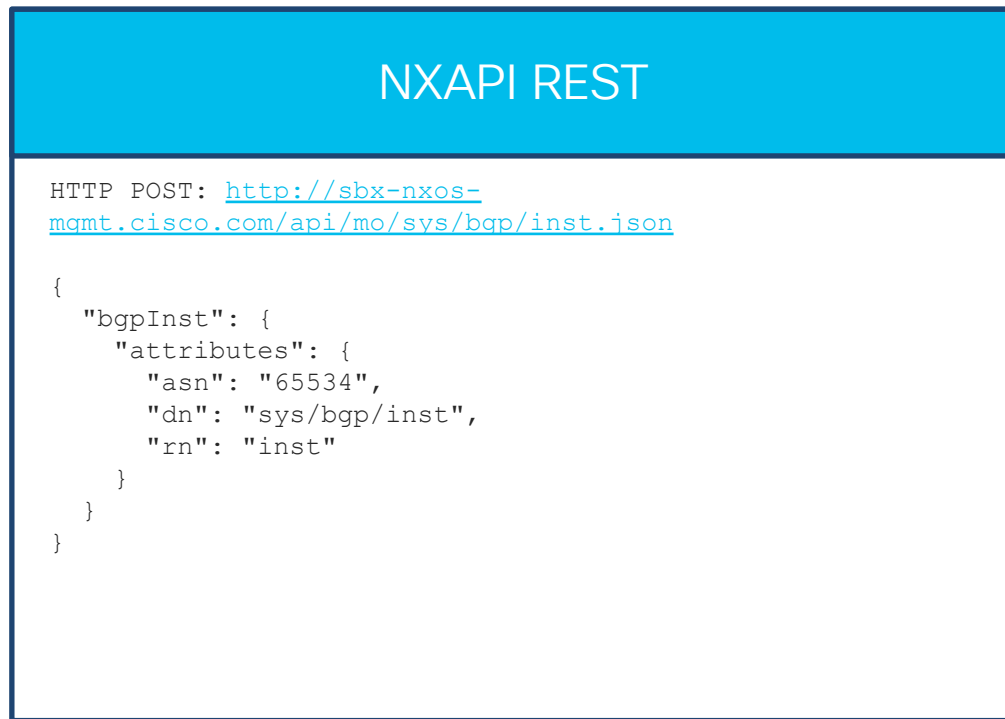
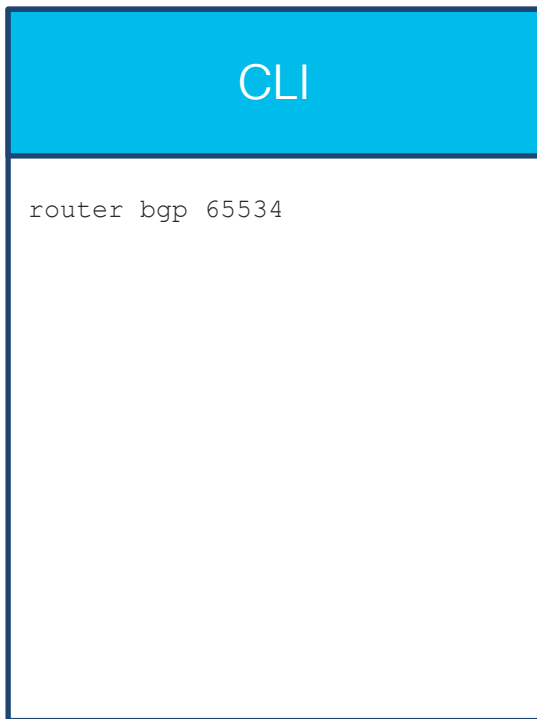
NXAPI CLI: Postman Usage



A cookie is still generated, but all RESTful calls can use “basic auth”

NXAPI REST Method

NXAPI REST Method



NXAPI REST: Convert from CLI to DME

The screenshot shows the NX-API Sandbox interface. At the top, it displays the Cisco logo, 'NX-API Sandbox', and user information: 'Username: admin Role: network-admin Version: 9.3(3) Management ip: 10.10.20.95 Hostname: sbx-n9ky-ao'. Navigation links include 'NX-API References', 'DME Documentation', 'Model Browser', and 'Logout'. The main area shows a CLI command 'router bgp 65534' in the input field. To the right, a red box highlights the 'Method' dropdown set to 'NX-API REST (DME)' and the 'Input type' dropdown set to 'cli'. Below the input field, another red box highlights the 'POST' method and the endpoint '/api/mo/sys.json'. To the right of the input field are 'Send', 'Reset', and 'Convert' buttons. Below the input field, there are tabs for 'Request', 'Python', 'Python3', 'Java', 'JavaScript', and 'Go-Lang'. The 'Request' tab is selected, showing a JSON payload for a BGP configuration. A 'Copy' button is visible next to the JSON output.

```
{
  "topSystem": {
    "children": [
      {
        "bgpEntity": {
          "children": [
            {
              "bgpInst": {
                "attributes": {
                  "asn": "65534"
                }
              }
            }
          ]
        }
      }
    ]
  }
}
```

DME model exploration may not be intuitive.

Sandbox can convert from CLI commands to DME*

*not all CLI exposed by DME

NXAPI REST: Convert from DME to CLI

The screenshot shows the NX-API Sandbox interface. At the top, the Cisco logo is on the left, followed by "NX-API Sandbox". To the right, user information is displayed: "Username:admin Role:network-admin Version:9.3(3) Management ip:10.10.20.95 Hostname:sbx-n9kv-ao". Further right are links for "NX-API References", "DME Documentation", and "Model Browser", along with a "Logout" button.

The main area is divided into two sections. The top section contains a JSON model for a BGP entity, with some fields like "bgpEntity" and "bgpInst" highlighted in red. The bottom section contains a text input field with the path "/api/mo/sys.json", a "Reset" button, and a "Convert" button.

On the right side, there is a control panel with two dropdown menus: "Method:" set to "NXAPI-REST (DME)" and "Input type:" set to "model". This panel is highlighted with a red rectangle.

Below the "Convert" button, there is a "Request" section showing the CLI output: "router bgp 65534". A green "Copy" button is located to the right of the output.

```
{
  "topSystem": {
    "children": [
      {
        "bgpEntity": {
          "children": [
            {
              "bgpInst": {
                "attributes": {
                  "asn": "65534"
                }
              }
            }
          ]
        }
      }
    ]
  }
}
```

/api/mo/sys.json

Reset Convert

Method: NXAPI-REST (DME)

Input type: model

Request

router bgp 65534

Copy

NXAPI REST: Postman Works (like ACI!)

The screenshot displays a Postman REST client interface. The top section shows a POST request to the URL `https://(nx-ao)/api/aaaLogin.json`. The request body is a JSON object:

```
{  "aaaUser": {    "attributes": {      "name": "{{username}}",      "pwd": "{{password}}"    }  }}
```

The bottom section shows the response, which is a 200 OK status with a JSON body:

```
{  "imdata": [    {      "aaaLogin": {        "attributes": {          "token": "aRxDLhSiYCi9madwfSM2MzofkTEZaejJnV63zaEc6JczlK0WuSmjfxwcBytyvesQ86sLX98xBEEuLTLAhZWL+0Q03iPGbCsPFdSCDCD0tazeVHKmn4v9p1wXkFs/rV/M0Q657No0tpnKPaeGDgpp1KpdYDm0n+3tJKYiTvPFXNwc",          "refreshTimeoutSeconds": "600",          "guidIdleTimeoutSeconds": "1200",          "restTimeoutSeconds": "300"        }      }    }  ]}
```

A 'MANAGE ENVIRONMENTS' dialog is open on the right, showing the following variables:

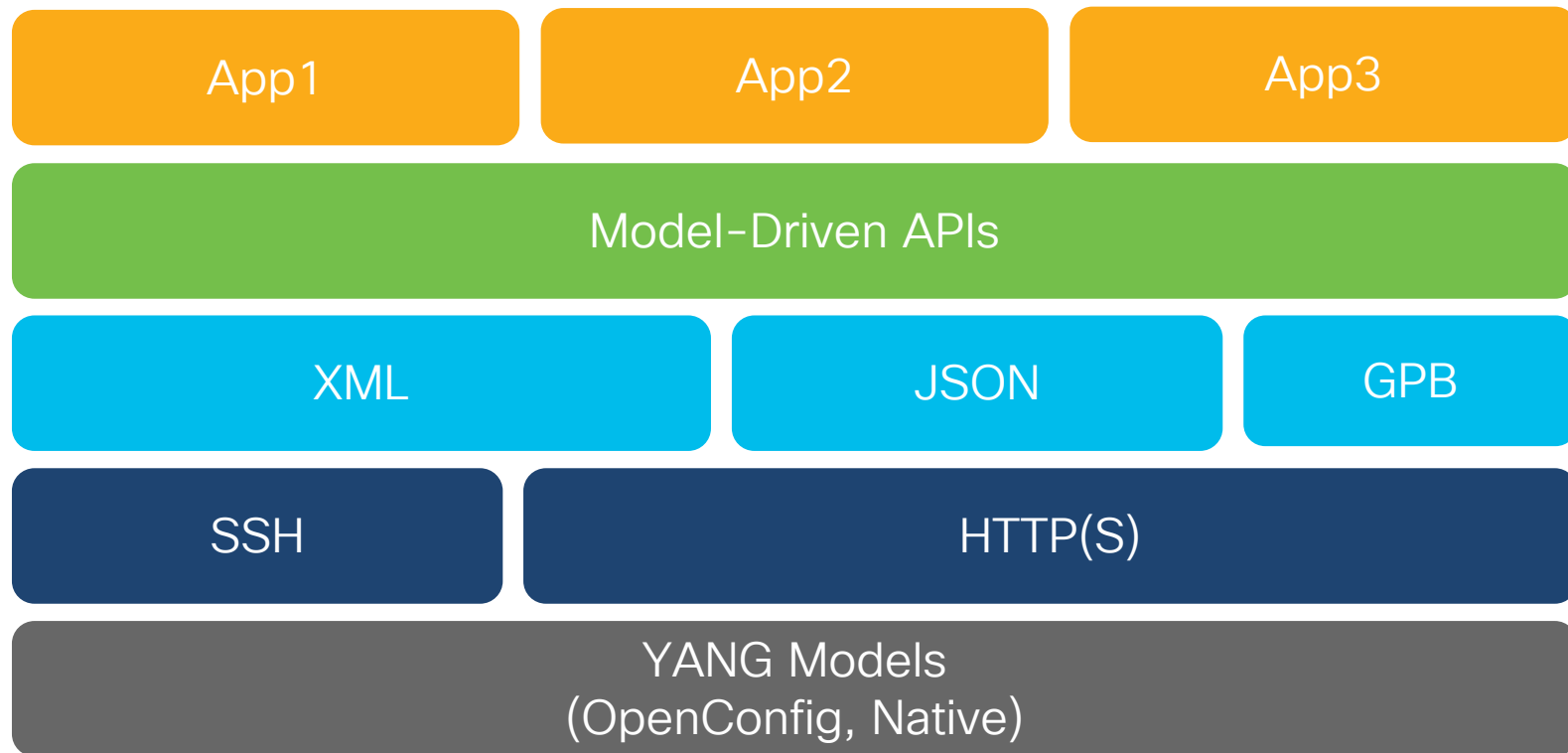
VARIABLE	INITIAL VALUE	CURRENT VALUE
<input checked="" type="checkbox"/> nx-ao	sbx-nxos-mgmt.cisco.cc	sbx-nxos-mgmt.cisco.com
<input checked="" type="checkbox"/> username	admin	admin
<input checked="" type="checkbox"/> password	Admin_1234!	Admin_1234!
<input checked="" type="checkbox"/> port	443	443

Red arrows indicate the mapping of these variables into the request body and the response. The 'Update' button is visible at the bottom right of the dialog.

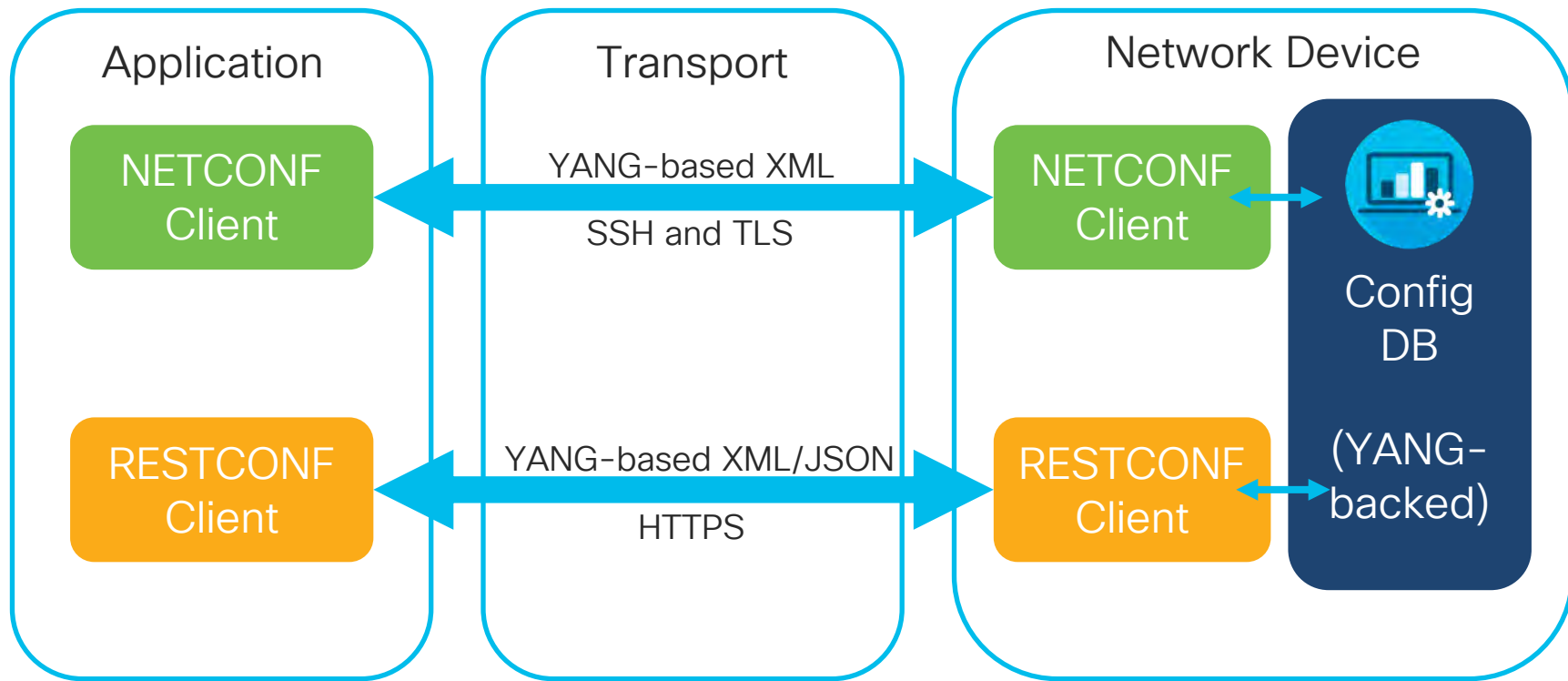


Just YANG It!

YANG: Enabling Model-Driven Programmability



YANG: NETCONF and RESTCONF

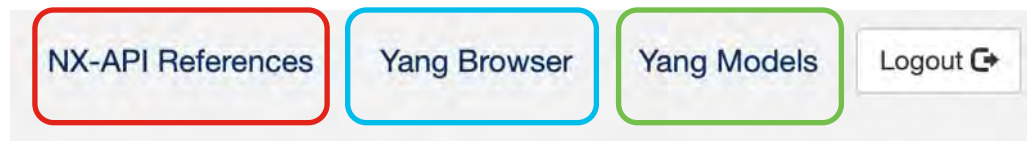


YANG: Useful Links in the Sandbox Too!

DevNet NX-API docs, aggregated

On-box YANG model browser
(rather than YangExplorer, etc)

GitHub repository of all YANG
models for all vendors; links to
Cisco Nexus



YANG: Accessing Models

```

module: Cisco-NX-OS-device
+--rw System
  +--rw bgp-items
    +--rw name?          naming_Name256
    +--rw adminSt?       nw_AdminSt
    +--ro operSt?        nw_EntOperSt
    +--rw inst-items
      +--rw asn?          bgp_AsnNum
      +--rw disPolBatch?  bgp_AdminSt
      +--rw disPolBatchv4PfxLst? string
      +--rw disPolBatchv6PfxLst? string
      +--ro createTs?     union
      +--ro activateTs?   union
      +--ro waitDoneTs?   union
      +--ro memAlert?     nw_MemAlertLevel
      +--ro numRtAttrib?   cap_Quant
      +--ro attribDbSz?    bgp_AttribDbSz
      +--ro numAsPath?     cap_Quant
      +--ro asPathDbSz?    bgp_AsPathDbSz
      +--rw isolate?       bgp_AdminSt
      +--rw medDampIntvl?  bgp_MedIntvl
      +--rw fabricSoo?     mtx_array_community
      +--rw flushRoutes?   bgp_AdminSt
      +--rw affGrpActv?    bgp_AffGrpActv
      +--ro srgbMinLbl?    bgp_SRGBRange
      +--ro srgbMaxLbl?    bgp_SRGBRange
      +--ro epeConfiguredPeers? bgp_NumPeers
      +--ro epeActivePeers? bgp_NumPeers
      +--ro lnkStSrvr?     bgp_LsAdminSt
      +--ro lnkStClnt?     bgp_LsAdminSt
      +--rw nhSupprDefRes? bgp_AdminSt
      +--rw enhancedErr?   boolean
      +--rw name?          naming_Name256
      +--rw adminSt?       nw_AdminSt
      +--rw ctrl?          nw_InstCtrl_Inst_ctrl
      +--ro operErr?       nw_OperErrQual
      +--rw dom-items

```

Module: Cisco-NX-OS-device, Namespace: <http://cisco.com/ns/yang/cisco-nx-os-device>, Prefix: top

Element	Schema	Type	Flags	Path
System	module			/top:System
name	leaf	string	config	/top:System/top:name
currentTime	leaf	union	no config	/top:System/top:currentTime
systemUpTime	leaf	union	no config	/top:System/top:systemUpTime
serial	leaf	eapi_Serial	no config	/top:System/top:serial
configModTs	leaf	mo_TStamp	no config	/top:System/top:configModTs
configDirty	leaf	boolean	no config	/top:System/top:configDirty
configModTsBackup	leaf	mo_TStamp	no config	/top:System/top:configModTsBackup
configDirtyBackup	leaf	boolean	no config	/top:System/top:configDirtyBackup
bgp-items	container		config	/top:System/top:bgp-items
bgptrap-items	container		config	/top:System/top:bgptrap-items
cbgp2trap-items	container		config	/top:System/top:cbgp2trap-items
callhome-items	container		config	/top:System/top:callhome-items
caploglevel-items	container		config	/top:System/top:caploglevel-items
clock-items	container		config	/top:System/top:clock-items
time-items	container		config	/top:System/top:time-items
ntp-items	container		config	/top:System/top:ntp-items
span-items	container		config	/top:System/top:span-items
originip-items	container		config	/top:System/top:originip-items
dhcp-items	container		config	/top:System/top:dhcp-items

YangModels / yang

<> Code 35 Issues 35 Pull requests 2

Branch: master yang / vendor / cisco / nx / 9.3-3 /

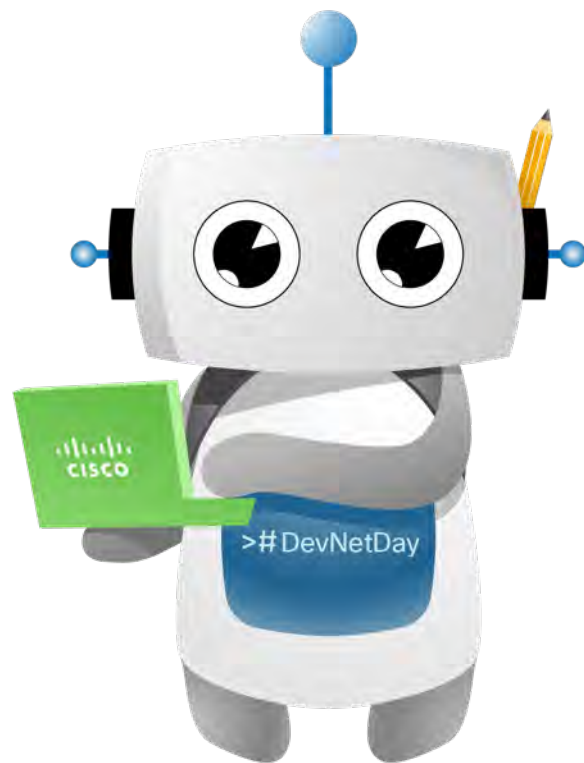
Create new file Find file History

kyoungyun and einarnn added models for 9.3-3 (#736) ✓ Latest commit 2839eb3 on Dec 25, 2019

extensions	added models for 9.3-3 (#736)	5 months ago
Cisco-NX-OS-device.yang	added models for 9.3-3 (#736)	5 months ago
README.md	added models for 9.3-3 (#736)	5 months ago
check-models.sh	added models for 9.3-3 (#736)	5 months ago
cisco-nx-openconfig-acl-deviations.yang	added models for 9.3-3 (#736)	5 months ago
cisco-nx-openconfig-bgp-policy-deviations.yang	added models for 9.3-3 (#736)	5 months ago
cisco-nx-openconfig-if-aggregate-deviations.yang	added models for 9.3-3 (#736)	5 months ago
cisco-nx-openconfig-if-ethernet-deviations.yang	added models for 9.3-3 (#736)	5 months ago
cisco-nx-openconfig-if-ip-deviations.yang	added models for 9.3-3 (#736)	5 months ago

Explore More

- NX-OS REST API Documentation
 - http://cs.co/NX-API_Docs
- YANG Model GitHub
 - <https://github.com/YangModels/yang>
- _NX-OS Programmability Learning Labs
 - http://cs.co/DevNet_NXOS
- Always-On NX-OS Sandbox
 - http://cs.co/NXOS_SBX
- Open NX-OS on DevNet
 - <https://developer.cisco.com/site/nx-os/>



Thank you



Possibilities

#CiscoLive | #DevNetDay