WELCOME TO DAY TWO!

© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

What is an example of a Service in your Network? (not one we have used as an example)

© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

What does NCS Stand For?

© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Agenda

- Programmatic Interfaces
- YANG
- XML
- REST

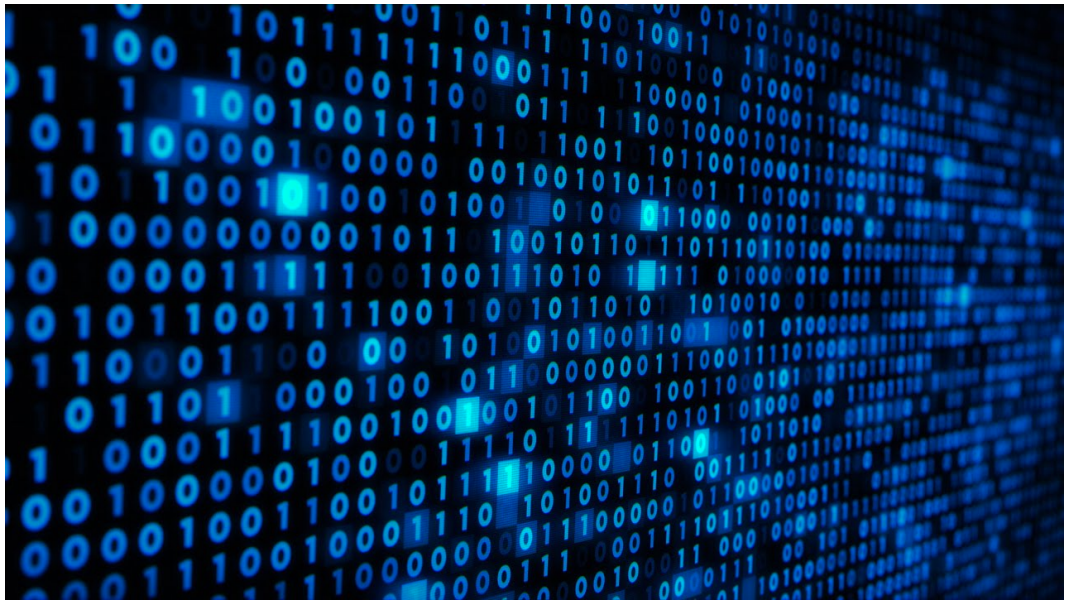
© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Interfaces that are good for humans…

Aren’t very good for machines

© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

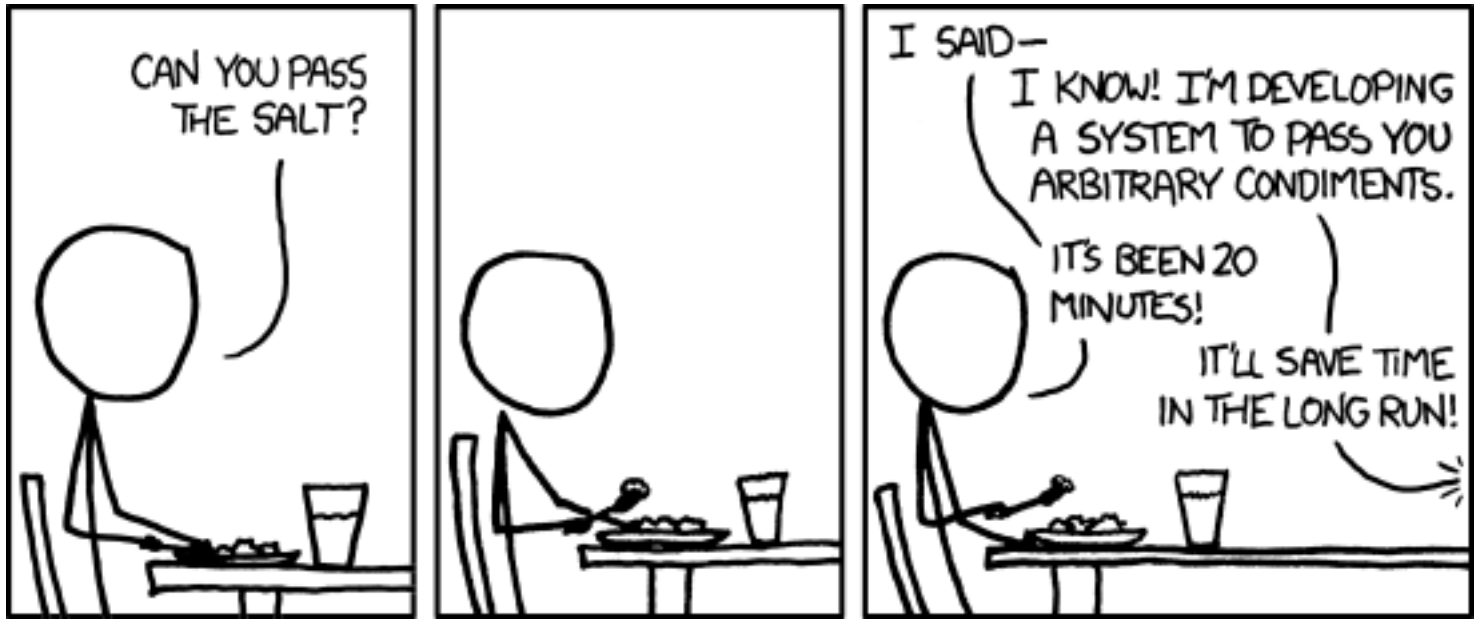
Interfaces that are good for machines



Aren’t very good for machines

© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

But its easier to make machine interfaces friendlier to people



Than people interfaces friendlier to machines

© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

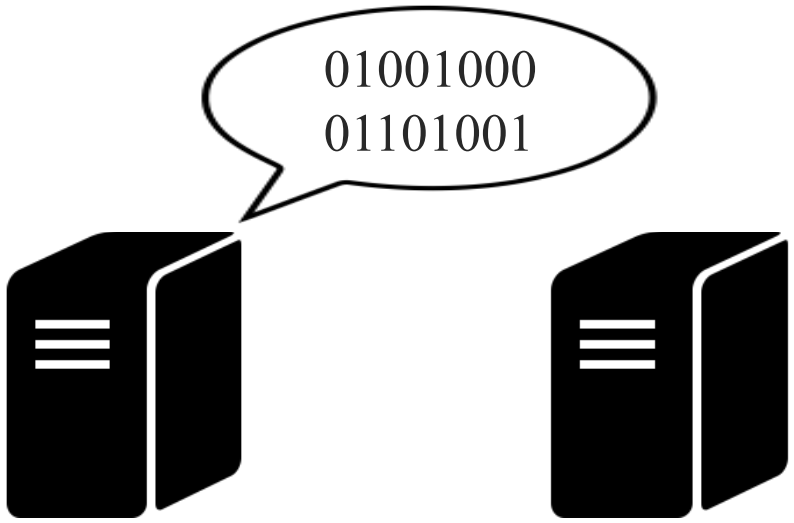
NSO CLI is a human-oriented interface

```
ncs-run — ncs_cli -C — 80×23
patrhuyn@ncs#
patrhuyn@ncs# devices device sjc12-31-sw1.cisco.com ?
Possible completions:
  check-sync              Check if the NCS config is in sync with the device
  check-yang-modules      Check if NCS and the device have compatible YANG
                          modules
  compare-config          Compare the actual device config with the NCS copy
  config                  NCS copy of the device configuration
  connect                 Connect to the device
  delete-config           Delete the config in NCS without deleting it in the
                          device
  disconnect              Close all sessions to the device
  live-status             Status data fetched from the device
  live-status-protocol    Additional protocols for the live-tree (read-only)
  netconf-notifications   NETCONF notifications from the device
  ping                    ICMP ping the device
  pioneer
  scp-from                Secure copy file to the device
  scp-to                  Secure copy file to the device
  ssh                     SSH connection configuration
  sync-from               Synchronize the config by pulling from the device
  sync-to                 Synchronize the config by pushing to the device
patrhuyn@ncs# devices device sjc12-31-sw1.cisco.com
```

A human-oriented interface provides a way for a person to interact and communicate with a machine.

© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Machine-Oriented Interfaces



- A machine-oriented interface provides a way for one machine to interact and communicate with another machine.
- We obtain maximum value from our automation when they use machine-oriented interfaces.
- But what is the fundamental difference between human-oriented and machine-oriented interfaces?

© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Example - HTML

Unstructured vs Structured Data

Wayne Rooney Forward 31 David de Gea Goalkeeper 26

Is there a hierarchy?

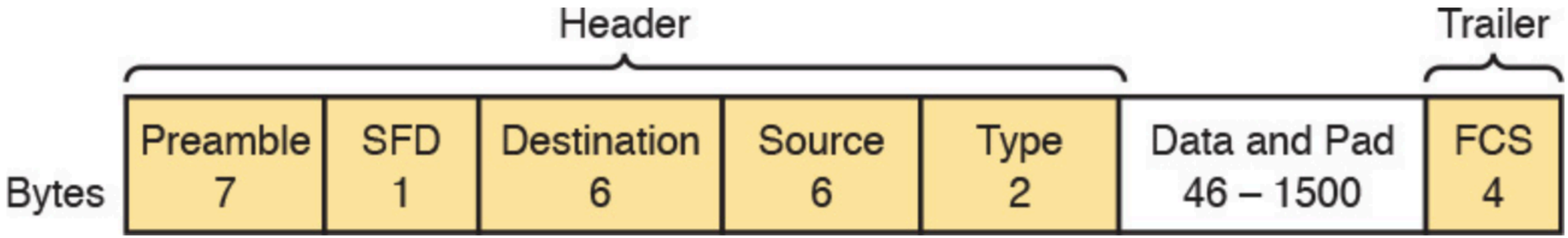
What are the possible fields?

Where does it begin and where does it end?

```
<Player>
  <Name>Wayne Roonery</Name>
  <Position>Forward</Position>
  <Age>31</Age>
</Player>
<Player>
  <Name>David de Gea</Name>
  <Position>Goalkeeper</Position>
  <Age>26</Age>
</Player>
```

© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Unstructured vs Structured Data cont.



Ethernet is designed with a machine-oriented interface in mind

© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Problem with Cisco CLI commands

Cts sxp connection peer 10.10.10.10 password default mode local listener hold-time 3600 36000

- What fields correspond to what? What fields are allowed? How do I compare fields?
- When making network automation, the burden of knowing structure and syntax of the commands is placed on the developer.

© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Current network automation is this:

Making a machine use a human-oriented interface.

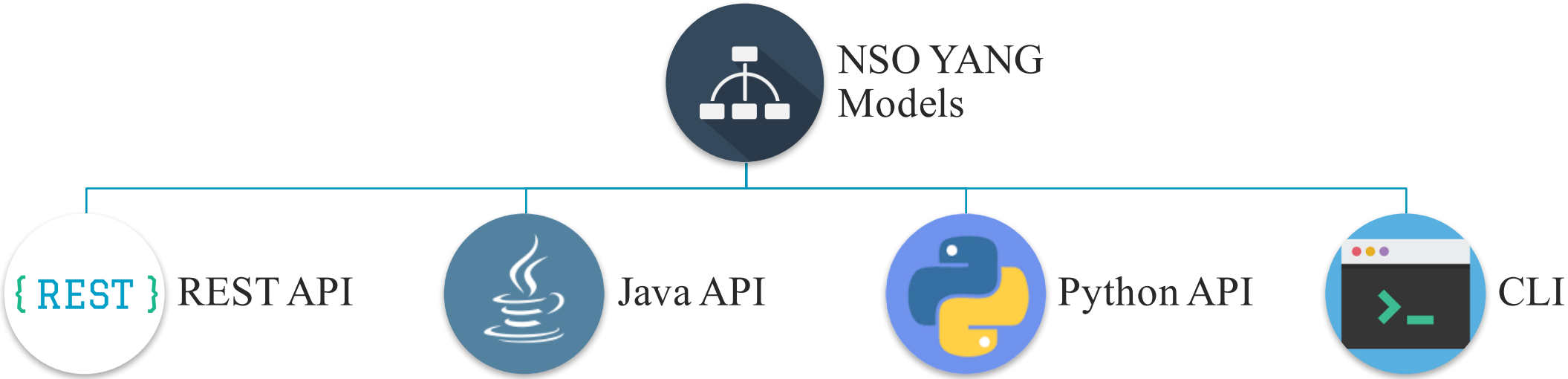
© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

So how do can we interact with Cisco CLI in a machine oriented way?

First, we need a way to translate CLI into a structured Data Form…

Introduction to Yang

Everything in NSO is YANG



All interfaces are defined by the NSO Yang Model

© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Example – Looking at NSO YANG Files

So what is Yang?

It is not the opposite of
Yin

Stands for:

Yet
Another
Next
Generation
(Data Modeling Language)

YANG is a data modeling language used to model configuration and state data manipulated by the Network Configuration Protocol (NETCONF)

© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

That's nice.. But what *IS* it really?

Yang allows for abstractions to be defined within the Yang language and then mapped into a markup language.

In our case, XML

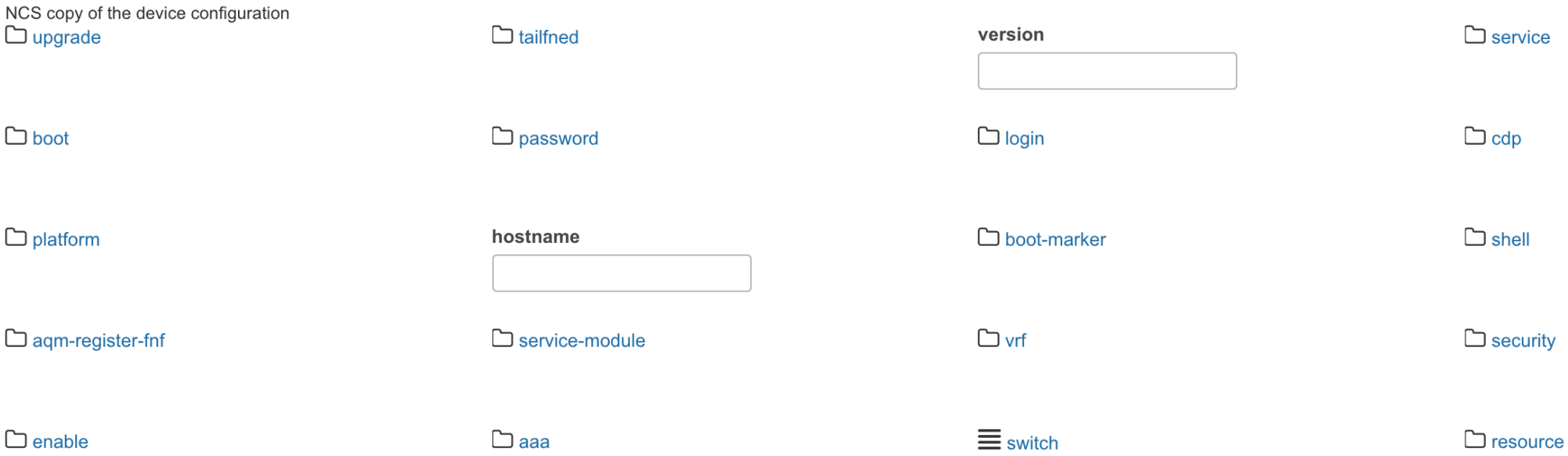
© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Yang types and a way to think about them

- Container
  - Groups things together
- List
  - A collection of containers
- Leaf
  - A end node of data
- Leaf-List
  - A list of single items

© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

The NSO UI uses this idea as well



*This is a UI representation of the Cisco-IOS Config Yang Model

© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Modeling a Football team in Yang

- Team should have a name.
- Has multiple players.
- Players have names.
- They have specific positions.
- They have an age.

Yang

```
Container FootballTeam {
  Leaf TeamName  {type string;}
  List Player  {
    Leaf PlayerName {type string;}
    Leaf Position {type string;}
    Leaf Age {type uint8;}
    }
}
```

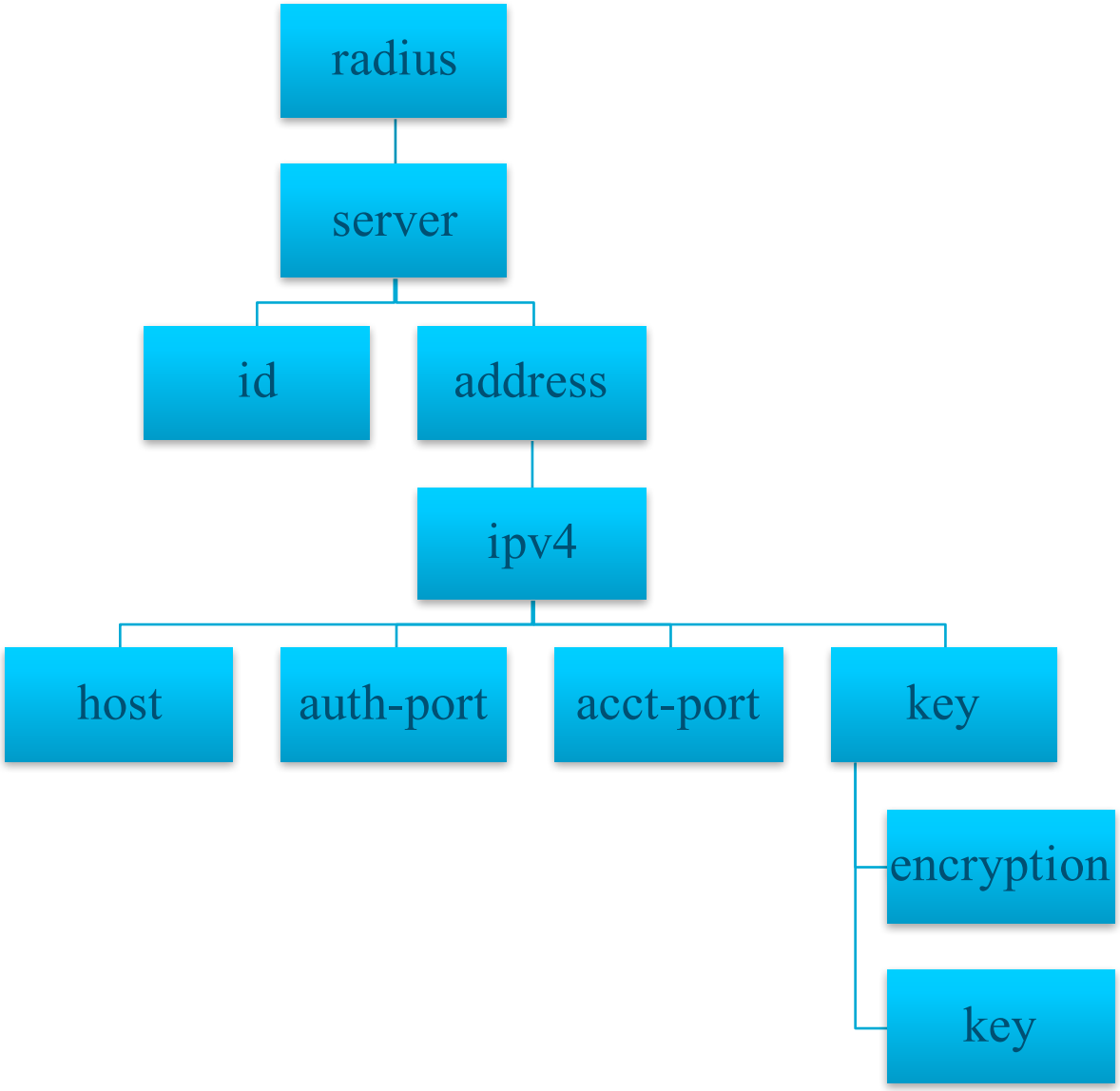
© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Modeling Cisco IOS commands in Yang

```
radius server <AAA Server>
    address ipv4 <IP Address> auth-port 1812 acct-port 1813
    key 7 <Encrypted Key>
```

```
container radius {
  list server {
    leaf id {type string;}
    container address {
      container ipv4 {
        leaf host {type string;}
        leaf auth-port {type uint16;}
        leaf acct-port {type uint16;}
        container key {
          leaf encryption {type enumeration;}
          leaf key {type string;}}}}}}
```

© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Confidential



© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Examples of YANG

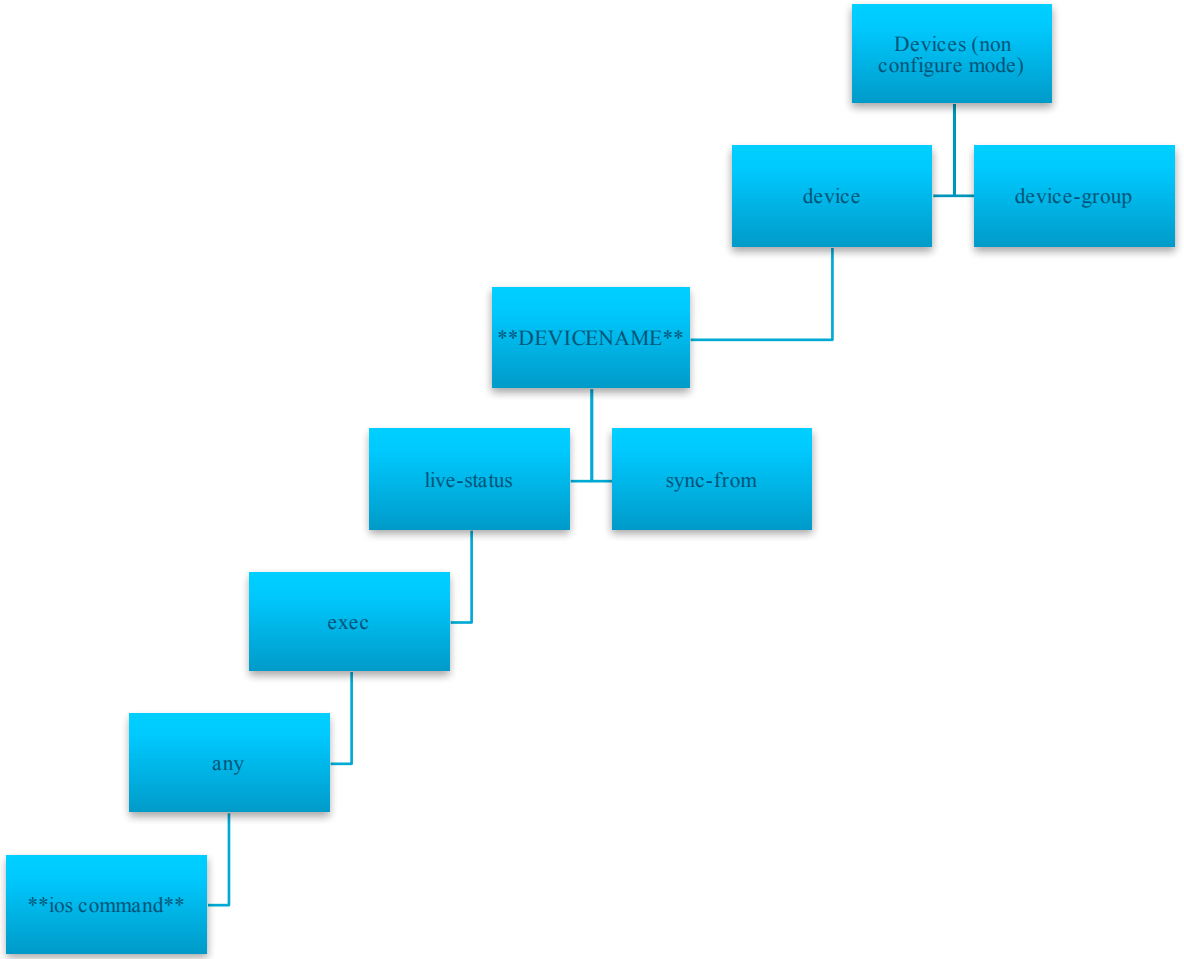
```
container access {
  description "Set access mode characteristics of the interface";
  leaf vlan {
    description "VLAN ID of the VLAN when this port is in access mode";
    type uint16 {
      range "1..4094";
    }
  }
}
container voice {
  description "Voice appliance attributes";
  leaf vlan {
    description "Vlan for voice traffic";
    type uint16 {
      range "1..4094";
    }
  }
}
```

```
//ip access-list resequence
container resequence {
  description
    "Resequence Access List";
  leaf numbers {
    type union {
      type ios-types:std-acl-type;
      type ios-types:ext-acl-type;
    }
  }
  leaf start-seq-no {
    type uint64 {
      range "1..2147483647";
    }
  }
  leaf step-seq-no {
    type uint64 {
      range "1..2147483647";
    }
  }
}
```

© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

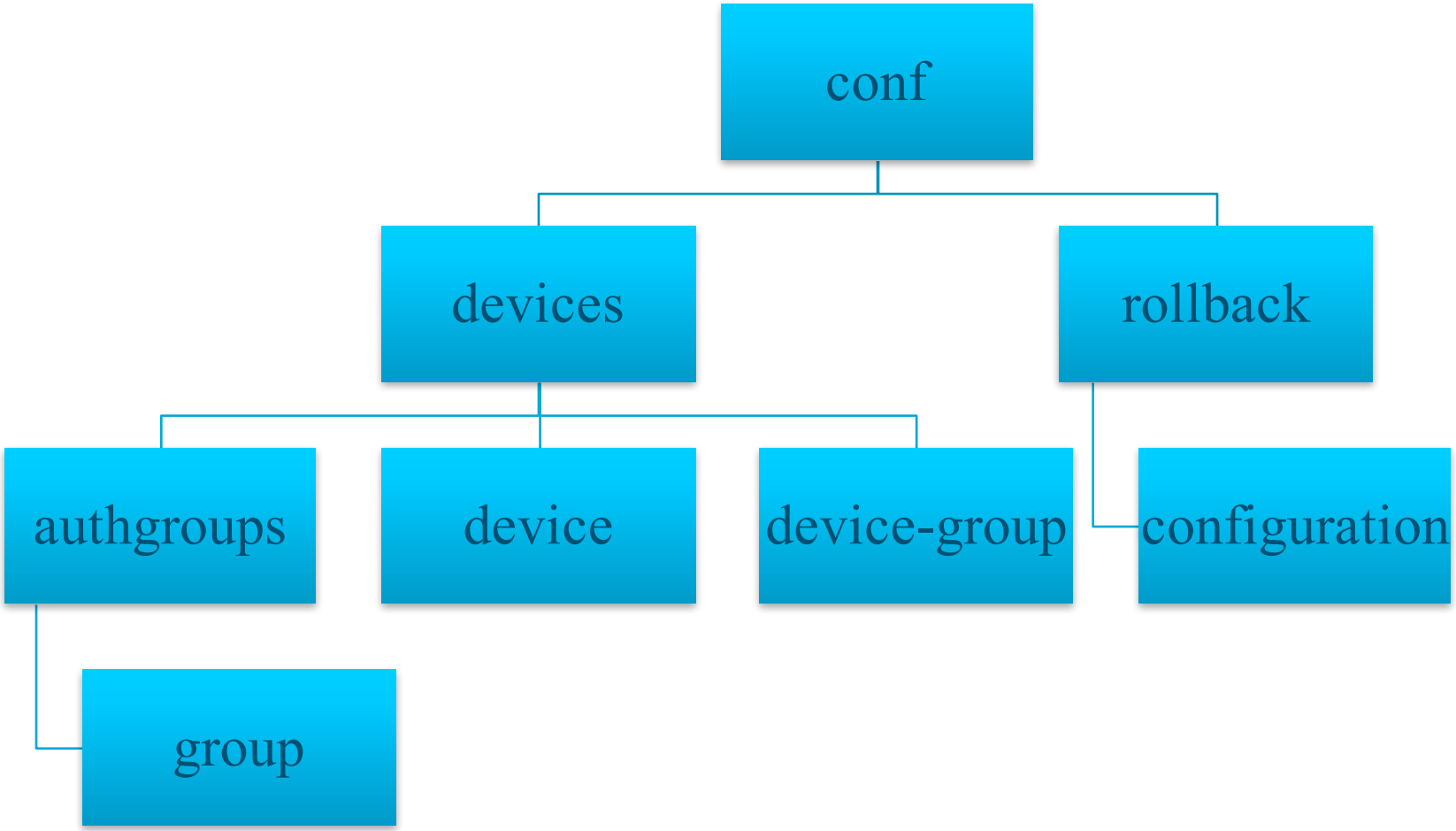
Hierarchical Data

NSO data tree (non configure mode)



© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

NSO data tree (configure mode)



© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

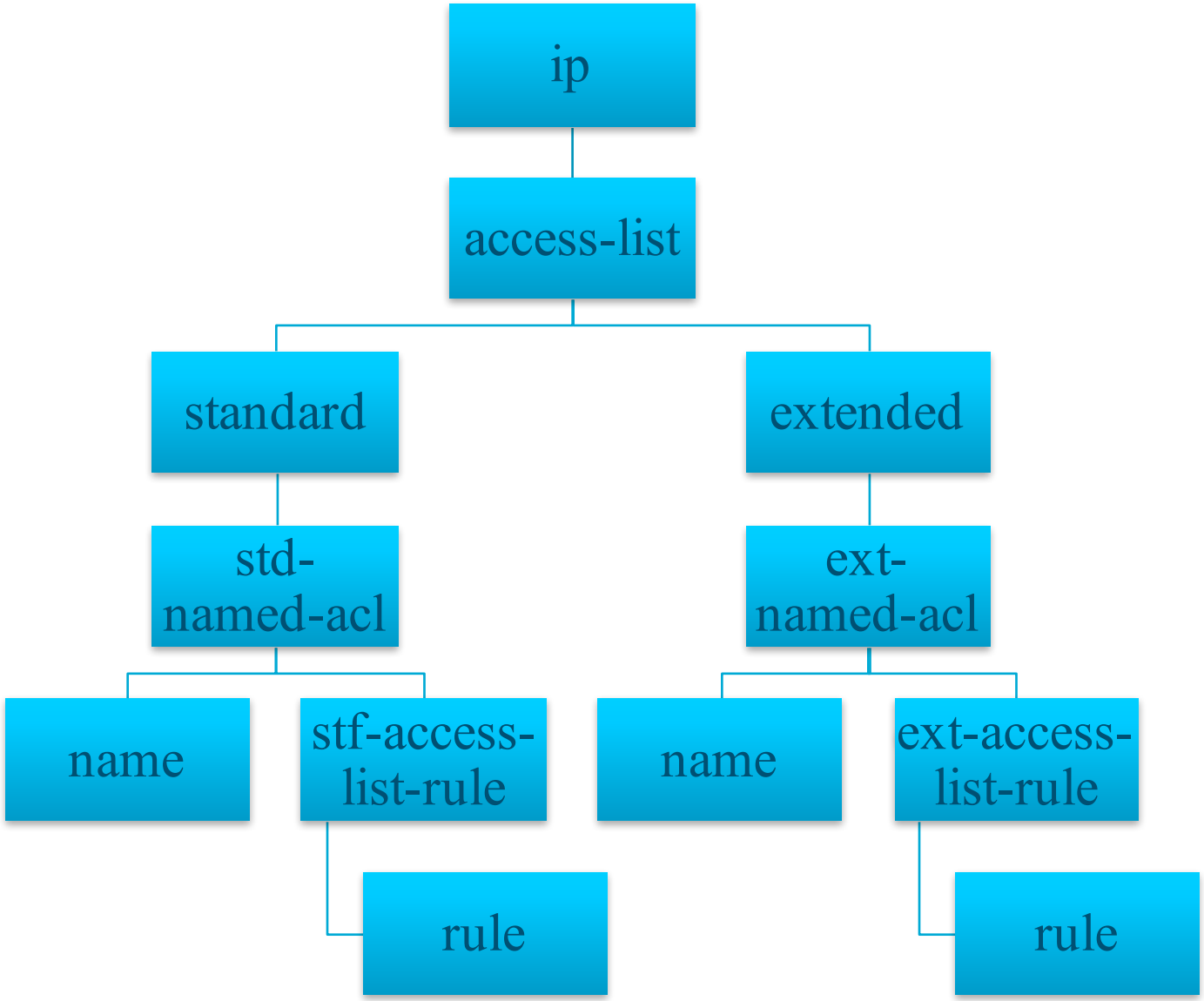
Lab – Navigating Data

Modeling Cisco IOS commands in Yang again

```
ip access-list standard <NAME>
    permit <IP address 1>
    permit <IP address 2>
ip access-list extended <Name>
   permit <rule>
   deny <rule>
```

```
Container ip
  container access-list {
    container standard
      list std-named-acl {
        leaf name {type std-acl-type;}
        list std-access-list-rule {
          leaf rule {type string;}}}}
    container extended {
      list ext-named-acl {
        leaf name {type string;}
        list ext-access-list-rule {
          leaf rule {type string;}}}}
```

© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Confidential



© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

So far we have been navigating the model.

But how is it built?

Demo – Structure of YANG module

Structure of a YANG Model

- YANG files are called modules
- Have a namespace and prefix
- Can import other YANG models
- Allows for meta data about the model
  - Description
  - Revision
  - Author

```
module ubvpn {

  namespace "http://example.com/ubvpn";
  prefix ubvpn;

  import ietf-inet-types {
    prefix inet;
  }
  import tailf-common {
    prefix tailf;
  }
  import tailf-ncs {
    prefix ncs;
  }

  description
    "This YANG Module defines service parameters and meta data for the UBVPN Service.";

  revision 2017-03-17 {
    description
      "
      First Edition. Beginning of development.
      Requriements to be flushed out and doucmented.";
  }
```

© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

YANG Types

Node Structure

- A node in a YANG Model follows a structure
- First, we define the node type and name

```
leaf mask {
    type inet:ipv4-address;
    tailf:info "A.B.C.D;;OSPF wild card bits";
}
```

© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Node Structure

- A node in a YANG Model follows a structure
- Second, we define data type of the node
- Standard types are:
  - String
  - Uint64
  - Ennumeration
- NSO enables custom types

```
leaf mask {
    type inet:ipv4-address;
    tailf:info "A.B.C.D;;OSPF wild card bits";
}
```

© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Node Structure

- A node in a YANG Model follows a structure
- Last, we define node modifiers
- These are extra pieces of info that tells NSO about the node or other things to do
- Common are:
  - Info
  - description
  - hidden

```
leaf mask {
    type inet:ipv4-address;
    tailf:info "A.B.C.D;;OSPF wild card bits";
}
```

© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Leaf

- A single thing with no sub-attributes
- You can specify the type of the Leaf
  - string
  - uint16
  - enumeration
  - custom types like “inet:ipv4-address”
- Add descriptions for NSO use
  - Use tailf:info

```
leaf mask {
    type inet:ipv4-address;
    tailf:info "A.B.C.D;;OSPF wild card bits";
}
```

© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Leaf-List

- A list of simple things
  - Each simple thing has the same type
  - Example:
    - switchport trunk allowed vlan 10,30,240,330,430,550,555,1079
- Different from List
  - “List” is a list of complex things

```
leaf-list vlans {
   type uint16 {
      tailf:info "WORD;;VLAN IDs of the allowed VLANs when this port is in trunking mode";
   }
}
```

© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Container

- A single complex thing
  - Has multiple attributes
  - Can have leaves, leaf-lists, lists, or more containers

```
container ip {
    list vrf {…}
    container mcr-conf {…}
    container access-list {...}
    ...
}
```

© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

List

- Multiple same, complex things
  - Each have multiple attributes
- Uses keys to determine the correct element
- Different from leaf-list
  - Leaf-lists are all simple things
  - Lists are much more common than leaf-lists

```
// aaa authentication login *
list login {
    key name;
    leaf name {type string;}
    leaf group;
    leaf local;
    leaf none;
}
```

© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

YANG Types

- Once we define a node and its name we need to define its type
- YANG enables us to do this with the type statement
- Common types:
  - string
  - uint64
  - enumeration
  - boolean
  - leafref

```
leaf Hub {
  tailf:info "The UBVPN Hub the site is in. To be mapped to ASA for configuration";
  mandatory true;
  type enumeration{
    enum "SJC";
    enum "BGL";
    enum "TYO";
    enum "AER";
  }
}
```

© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

YANG Type Modifiers

- When we define a leaf in NSO, we can control the data allowed
- YANG Type Modifiers enable us to create coarse or finely defined models
- Enables us to to limit and control what and how data is entered into the model

```
leaf lab_id {
      tailf:info "Lab ID";
      mandatory true;
      type uint32 {
        range "1000..99999";
      }
    }
```

© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

YANG Type Modifiers

- When we define a leaf in NSO, we can control the data allowed
- Node types have specific modifiers
- Lists
  - Key, limit
- Container
  - Presence
- Leaf
  - Mandatory, pattern, range

© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Building Our Own Types

- NSO enables us to define our Own Types for re-use
- We do this through the typedef statement
- Can be as simple or complex as we want

```
typedef ipv4-address {
  type string {
    pattern
      '(([0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])\.){3}'
    +  '([0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])'
    + '(%[\p{N}\p{L}]+)?';
  }
```

© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Exercise as a team

- As a small team (3-4 People) create your own type
- Create a type to represent one of the following:
  - PIN in the Network
  - Common Interface Type (FastEthert, GigabitEthernet, etc..)
  - Lab-ID
- Needs:
  - Node statement
  - Type statement
  - Type Modifiers

© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Groupings

- NSO enables use to build re-usable models
- Groupings are groupings of node types that can be referenced
- Referenced through “uses group-name”

```
uses ncs:service-data;
ncs:servicepoint ubvpn-servicepoint;
```

```
grouping interface-name-grouping {
  choice interface-choice {

    leaf Null {
      tailf:info "Null interface";
      tailf:cli-allow-join-with-value {
        tailf:cli-display-joined;
      }
      tailf:non-strict-leafref {
        path "/ios:interface/Null/name";
      }
      type uint8 {
        tailf:info "<0-0>;;Null interface number";
        range "0";
      }
    }
```

© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Exercise as a team

- As a small team (3-4 People) create your own grouping
- Create a grouping to represent one of the following:
  - ACL
  - Interface (type, number, description)
  - Lab-ID
- Needs:
  - Grouping statement
  - Node statements
  - Type statements

© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Lunch

Lab – YANG Modeling

What does YANG Stand For?

© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

What does XML Stand For?

© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Demo –
Show Running Config in XML
See Day 1 Slides

XML Intro
(Taken from W3Schools)

What is XML?

- XML stands for eXtensible Markup Language
- XML is a markup language much like HTML
- XML was designed to store and transport data
- XML was designed to be self-descriptive
- XML is a W3C Recommendation

XML and HTML were designed with different goals:

- XML was designed to carry data - with focus on what data is
- HTML was designed to display data - with focus on how data looks
- XML tags are not predefined like HTML tags are

© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

XML Does Not DO Anything

Maybe it is a little hard to understand, but XML does not DO anything.

This note is a note to Tove from Jani, stored as XML:

```
<note>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

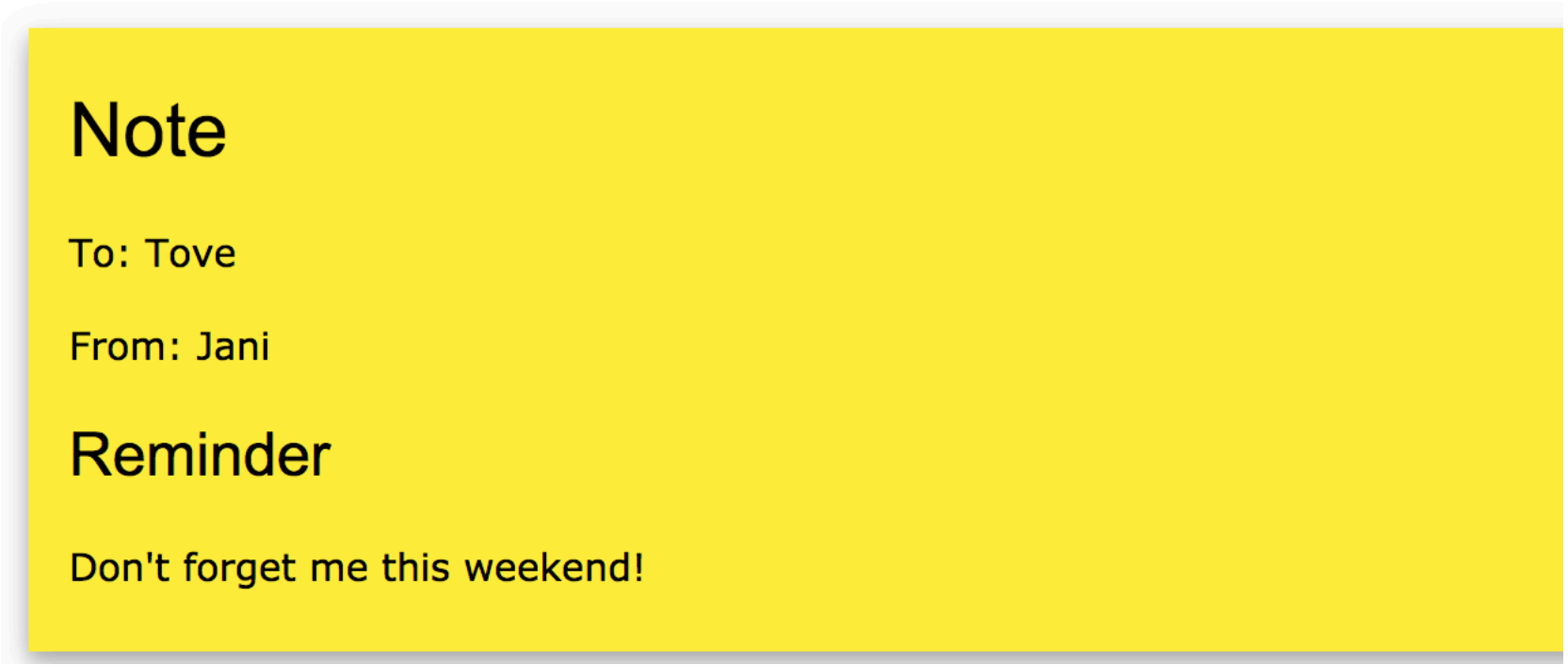
The XML above is quite self-descriptive:

- It has sender information.
- It has receiver information
- It has a heading
- It has a message body.

© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Visualize the Data

Someone must write a piece of software to send, receive, store, or display it:



© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

How Can XML be Used?

XML Separates Data from Presentation

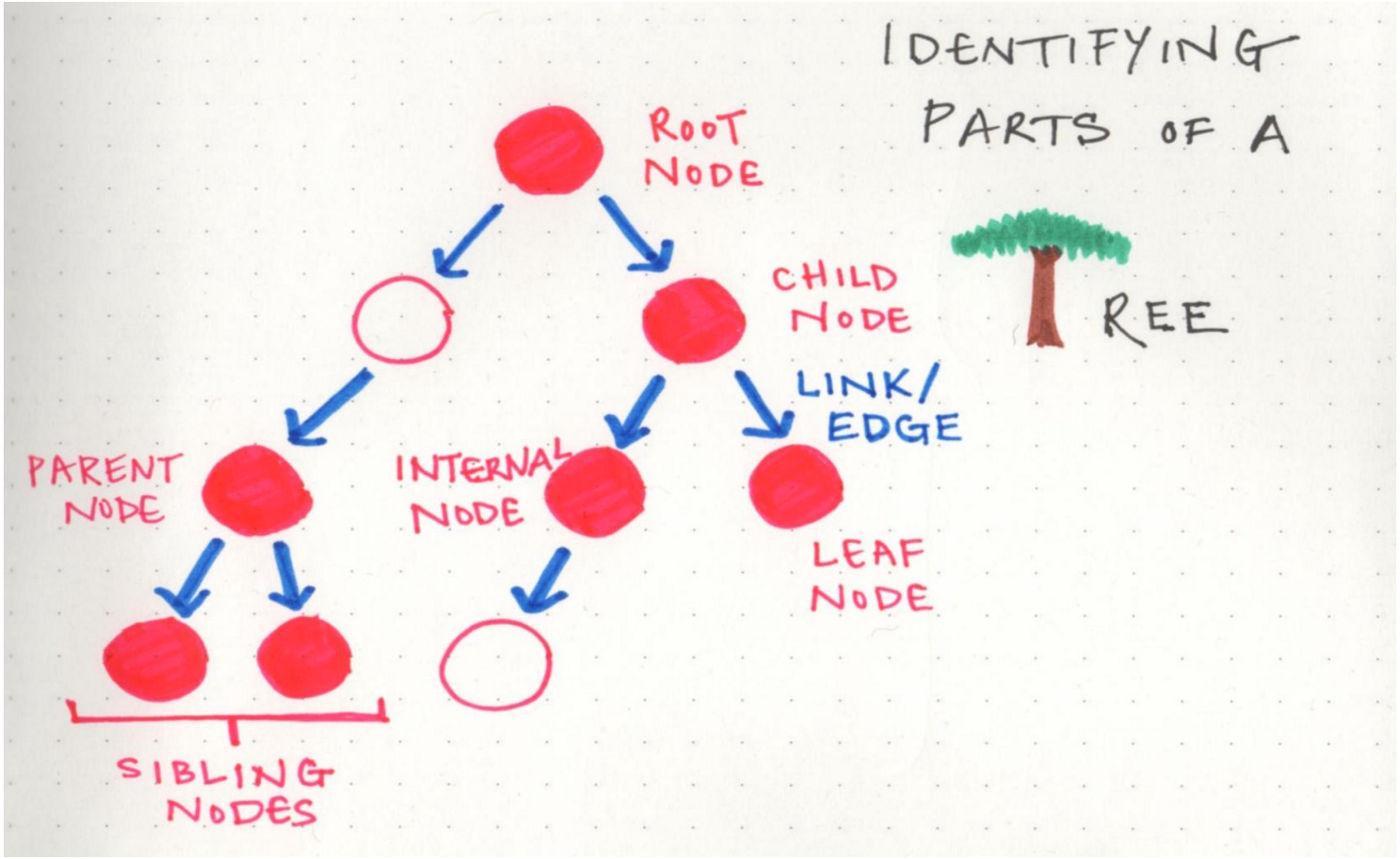
XML does not carry any information about how to be displayed.
The same XML data can be used in many different presentation scenarios.
Because of this, with XML, there is a full separation between data and presentation.

© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

XML Tree Structure

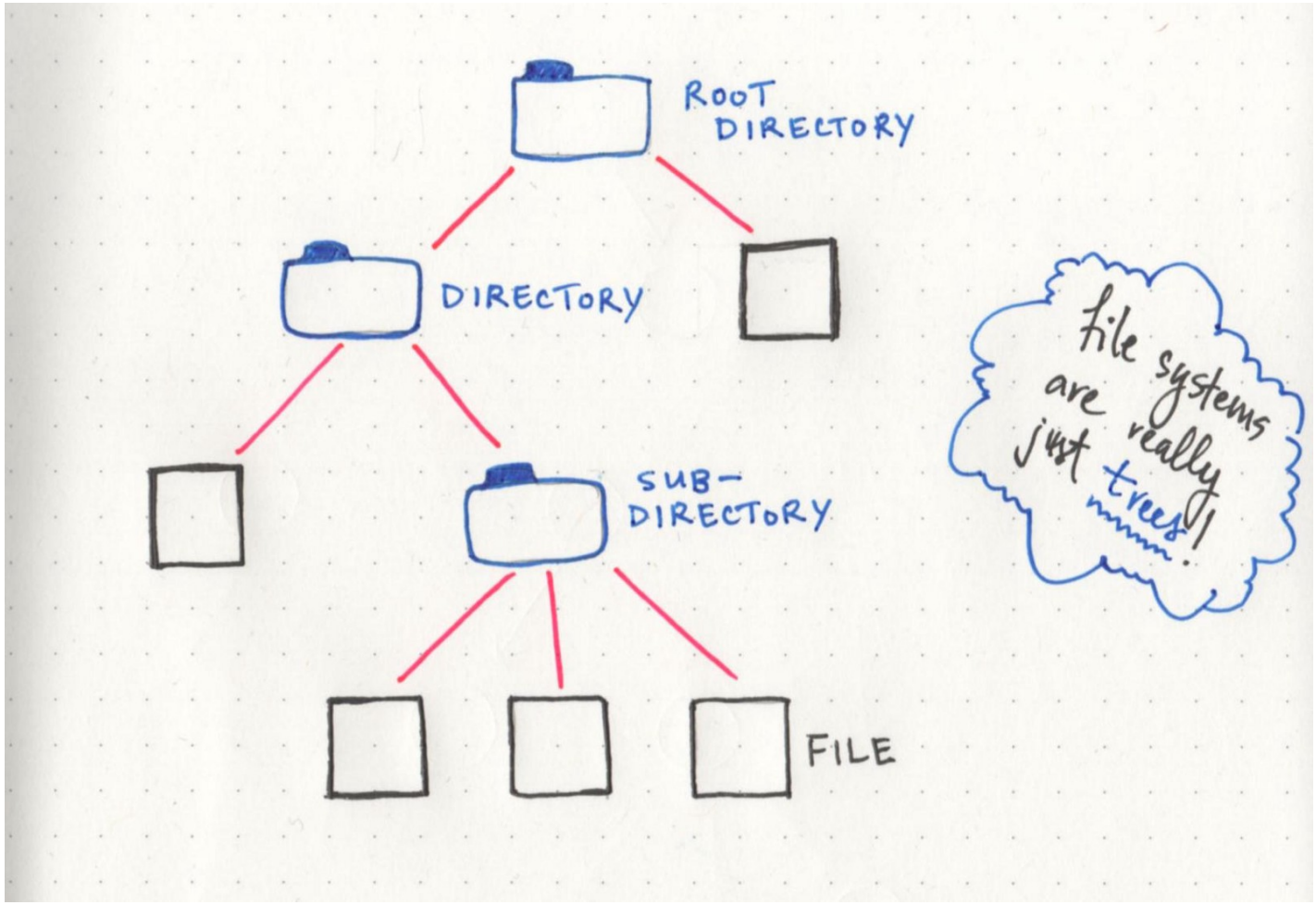
© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

XML Tree Structure



© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

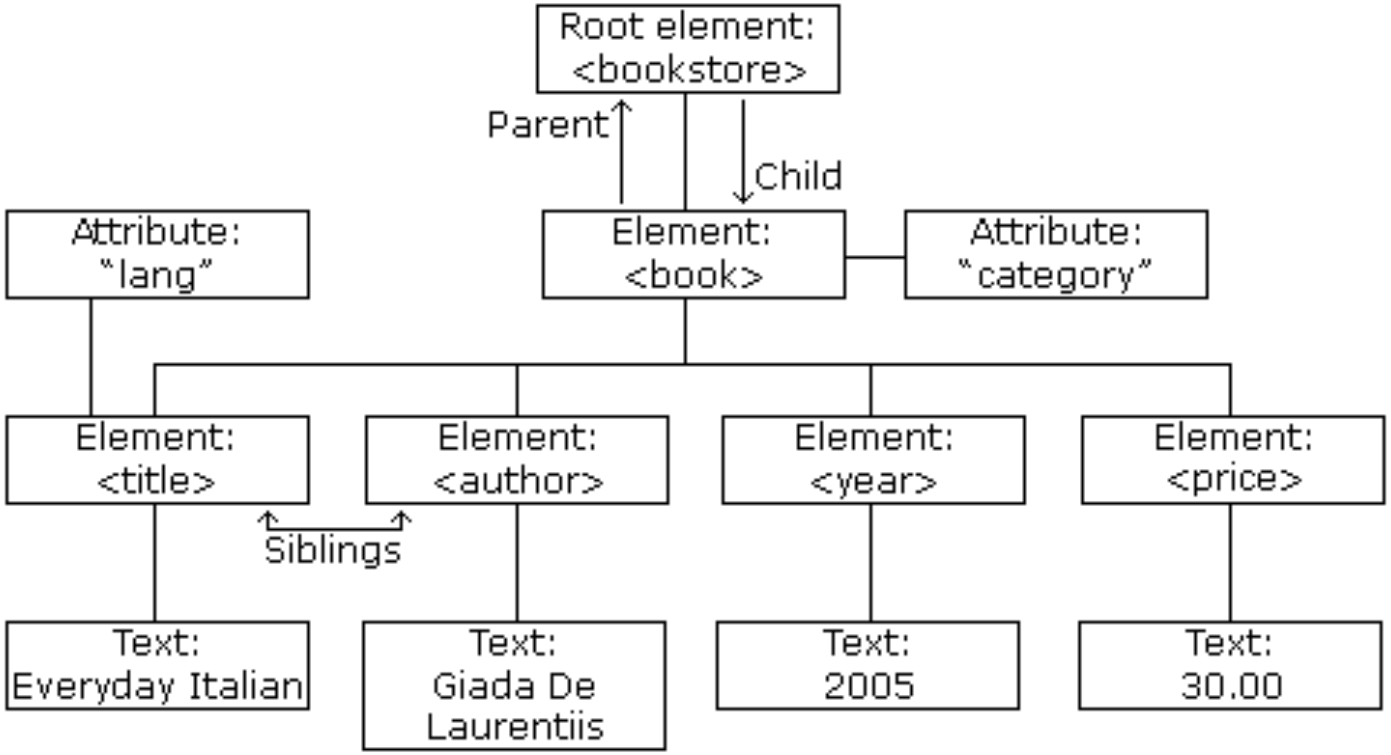
XML Tree Strucute - Example



© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

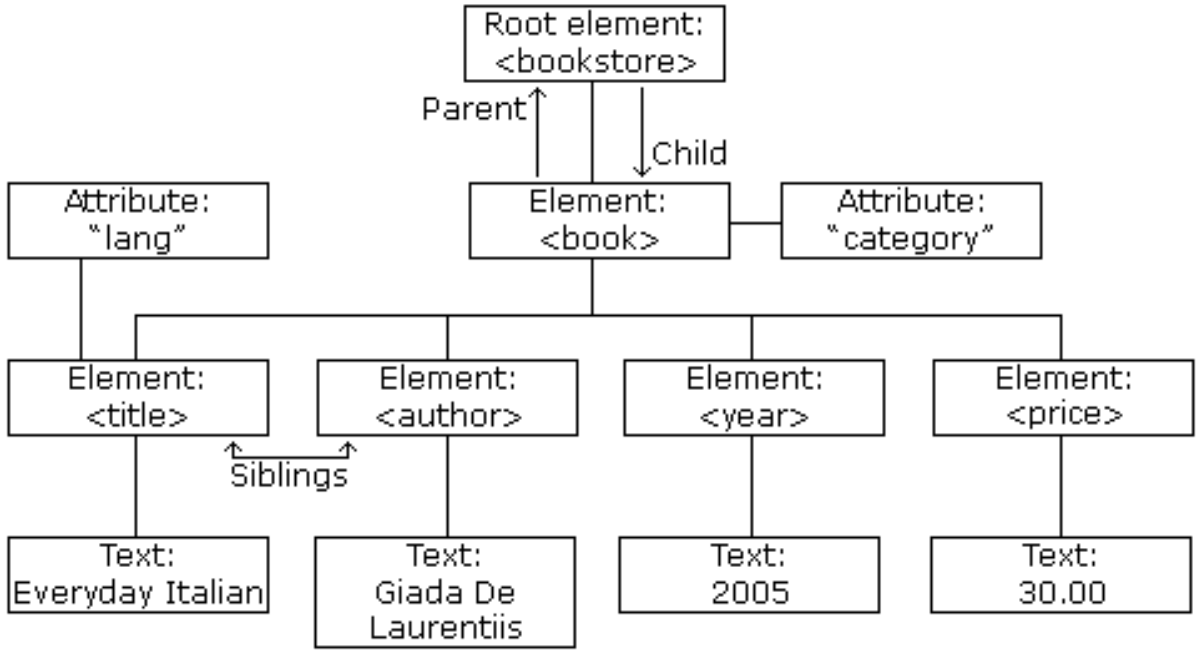
XML Tree Structure

XML documents form a tree structure that starts at "the root" and branches to "the leaves".



© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

An Example XML Document




```xml
<?xml version="1.0" encoding="UTF-8"?>
<bookstore>
  <book category="cooking">
    <title lang="en">Everyday Italian</title>
    <author>Giada De Laurentiis</author>
    <year>2005</year>
    <price>30.00</price>
  </book>
  <book category="children">
    <title lang="en">Harry Potter</title>
    <author>J K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>
  <book category="web">
    <title lang="en">Learning XML</title>
    <author>Erik T. Ray</author>
    <year>2003</year>
    <price>39.95</price>
  </book>
</bookstore>
```

© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Quiz: Write out an xml tree using tags for a cricket team

© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Cisco IOS in XML

Cutsheet

radius server [AAA Server]

address ipv4 [IP Address] auth-port [Port #] acct-port [Port #]

key [Level] [Encrypted Key]

Actual Configuration

radius server primary-trustsec-radius

address ipv4 10.0.0.1 auth-port 1812 acct-port 1813

key 7 ABCDEF

```
<radius xmlns="urn:ios">
 <server>
  <id>primary-trustsec-radius</id>
    <address>
     <ipv4>
      <acct-port>1813</acct-port>
      <auth-port>1812</auth-port>
      <host>10.1.1.1</host>
     </ipv4>
    </address>
    <key>
     <type>7</type>
     <secret>ABCDEF</secret>
    </key>
</radius>
```

© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

XML Tree Structure

XML documents are formed as **element trees**.
An XML tree starts at a **root element** and branches from the root to **child elements**.
All elements can have sub elements (child elements):

```
<root>
  <child>
    <subchild>.....</subchild>
  </child>
</root>
```

The terms parent, child, and sibling are used to describe the relationships between elements. Parent have children. Children have parents. Siblings are children on the same level (brothers and sisters).
All elements can have text content (Harry Potter) and attributes (category="cooking").

© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

XML Syntax Rules

XML Documents Must Have a Root Element

XML documents must contain one **root** element that is the **parent** of all other elements:

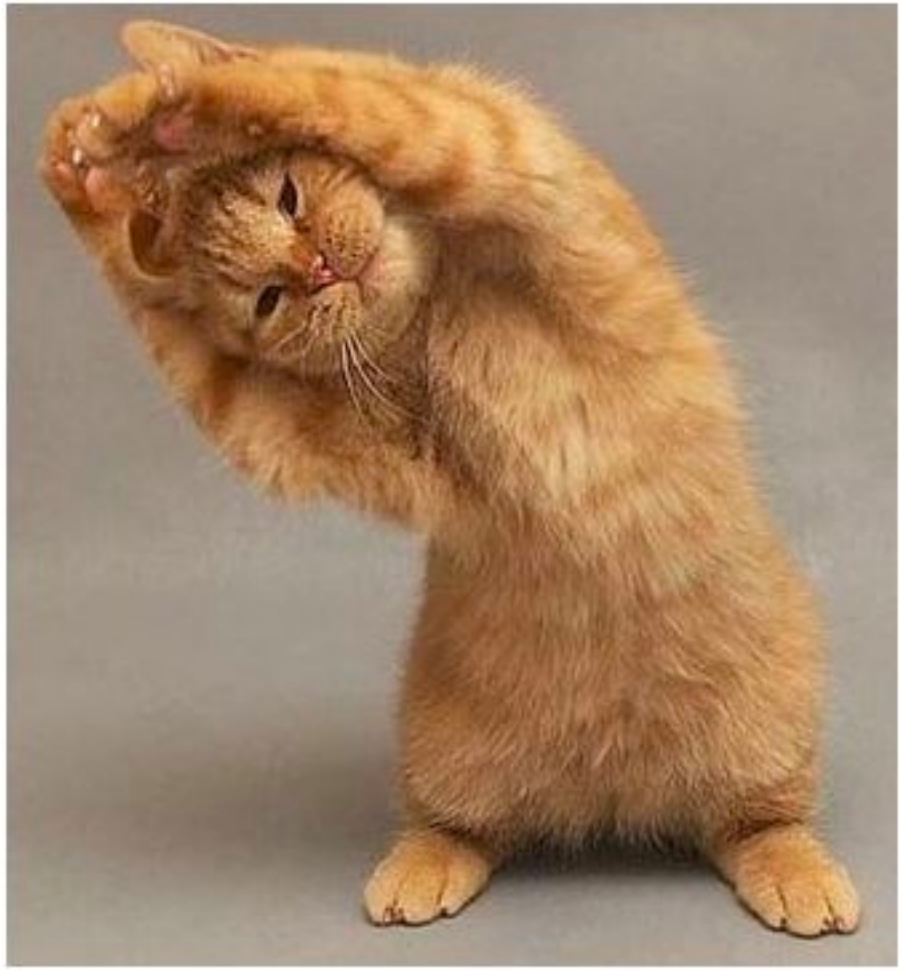
```
<root>
  <child>
    <subchild>.....</subchild>
  </child>
</root>
```

In this example **<note>** is the root element:

```
<?xml version="1.0" encoding="UTF-8"?>
<note>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Stretch Break!



© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

XML Elements (actual data)

An XML element is everything from (including) the element's start tag to (including) the element's end tag.

```
<price>29.99</price>
```

An element can contain:
- text
- attributes
- other elements
- or a mix of the above

An element with no content is said to be empty.
In XML, you can indicate an empty element like this:

```
<element></element>
```

You can also use a so called self-closing tag:

```
<element />
```

© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

XML Attributes (meta-data)

Attributes are designed to contain data related to a specific element. (meta-data)
Attribute values must always be quoted. Either single or double quotes can be used.

For example, a person's gender, the <person> element can be written like this:

```
<person gender="female">
```

Some things to consider when using attributes are:

- attributes cannot contain multiple values (elements can)
- attributes cannot contain tree structures (elements can)
- attributes are not easily expandable (for future changes)

© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

```xml
<bookstore>
  <book category="children">
    <title>Harry Potter</title>
    <author>J K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>
  <book category="web">
    <title>Learning XML</title>
    <author>Erik T. Ray</author>
    <year>2003</year>
    <price>39.95</price>
  </book>
</bookstore>
```

In the example above:
<title>, <author>, <year>, and <price> have **text content** because they contain text (like 29.99).
<bookstore> and <book> have **element contents**, because they contain elements.
<book> has an **attribute** (category="children").

© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

XML Namespaces (using prefixes)

In XML, element names are defined by the developer. This often results in a conflict when trying to mix XML documents from different XML applications.

Name conflicts in XML can easily be avoided using a name prefix.

When using prefixes in XML, a **namespace** for the prefix must be defined.
The namespace can be defined by an **xmlns** attribute in the start tag of an element.
The namespace declaration has the following syntax. xmlns:*prefix*="*URI*".

A **Uniform Resource Identifier** (URI) is a string of characters which identifies an Internet Resource.
The most common URI is the **Uniform Resource Locator** (URL) which identifies an Internet domain address. Another, not so common type of URI is the **Universal Resource Name** (URN).

© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

NSO Namespaces

XMLTemplate

```
<config-template xmlns="http://tail-f.com/ns/config/1.0">
    <devices xmlns="http://tail-f.com/ns/ncs">
        <device>
            <name>{/Devices}</name>
            <config>
              <ip xmlns="http://cisco.com/ned/asa">
                  <local>
                      <pool>
                          <id>{$Partner_site_code}-{$Country_Code}-pool</id>
                          <address>{$Address_pool_start}-{$Address_pool_end}</address>
                          <mask>{$Address_pool_mask}</mask>
                      </pool>
                  </local>
              </ip>
```

CLI Template

```
svl-gm-joe-asa-fw1
  ip local pool site-partner-pool 10.0.0.0-10.0.0.255 mask 255.255.255.0
  group-policy site-partner internal
  group-policy site-partner attributes
  address-pools value site-partner-pool
  vpn-simultaneous-logins 1
exit
```

© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

QoS Example

```
<policy-map xmlns="urn:ios">
  <name>classify</name>
  <description>QoS 2.3.2-</description>
  <class>
    <name>qos-scavenger</name>
    <set>
      <dscp>
        <value>8</value>
      </dscp>
    </set>
  </class>
  <class>
    <name>qos-medium-priority</name>
    <set>
      <dscp>
        <value>16</value>
      </dscp>
    </set>
  </class>
```

```
<interface xmlns="urn:ios">
  <GigabitEthernet>
    <name>1/0/1</name>
    <service-policy>
      <input>TRUST-MARKING</input>
    </service-policy>
  </GigabitEthernet>
  <TenGigabitEthernet>
    <name>1/1/1</name>
    <service-policy>
      <input>TRUST-MARKING</input>
    </service-policy>
  </TenGigabitEthernet>
  <Vlan>
    <name>10</name>
    <service-policy>
      <input>classify</input>
    </service-policy>
  </Vlan>
```

© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Back to NSO

The NSO Database is an XML database.

All devices' in NSO have their configuration stored in the XML Database

All Data is stored in XML, and is in a hierarchy (see previous slides)
You can see the XML and save it to a file, use it as a template, or use it to import the data into NSO

admin@ncs# show running-config devices device rtp5-itnlab-dt-sw1.cisco.com | display xml

© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

NSO XML device data

```
<config xmlns="http://tail-f.com/ns/config/1.0">
  <devices xmlns="http://tail-f.com/ns/ncs">
  <device>
    <name>rtp5-itnlab-dt-sw1.cisco.com</name>
    <address>rtp5-itnlab-dt-sw1.cisco.com</address>
    <port>22</port>
    <ssh>
      <host-key>
        <algorithm>ssh-rsa</algorithm>
        <key-data>AAAAB3NzaC1yc2EAAAADAQABAAAAgQC0uw3/fNscjQqUQhxLEh4Y82vSZB7K5yR2h+XFaB0c
432kN16CC5QukBBDn06o80N4E009EaBFz7VVuNYDa5dz1SoeJJtGXmP4/mwxUNmRYIUEAOdF
5CU9YqlcGLawSnj+ndBYoxkSbV0IA3wHy6JYwEB/Rw6CH4dGWW+lju7ujw==</key-data>
      </host-key>
    </ssh>
    <authgroup>jabelk.web.auth</authgroup>
    <device-type>
      <cli>
        <ned-id xmlns:ios-id="urn:ios-id">ios-id:cisco-ios</ned-id>
      </cli>
    </device-type>
    <state>
      <admin-state>unlocked</admin-state>
    </state>
```

© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Config data for boot vars

```
<config>
      <hostname xmlns="urn:ios">rtp5-itnlab-dt-sw1</hostname>
      <boot-marker xmlns="urn:ios">
        <boot>
         <system>
           <entry>flash bootflash:cat4500es8-universalk9.SPA.03.06.05.E.152-2.E5.bin</entry>
         </system>
         <system>
           <entry>flash slot0:cat4500es8-universalk9.SPA.03.06.05.E.152-2.E5.bin</entry>
         </system>
         <system>
           <entry>flash bootflash:cat4500es8-universalk9.SPA.03.08.02.E.152-4.E2.bin</entry>
         </system>
         <system>
           <entry>flash slot0:cat4500es8-universalk9.SPA.03.08.02.E.152-4.E2.bin</entry>
         </system>
        </boot>
      </boot-marker>
```

© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

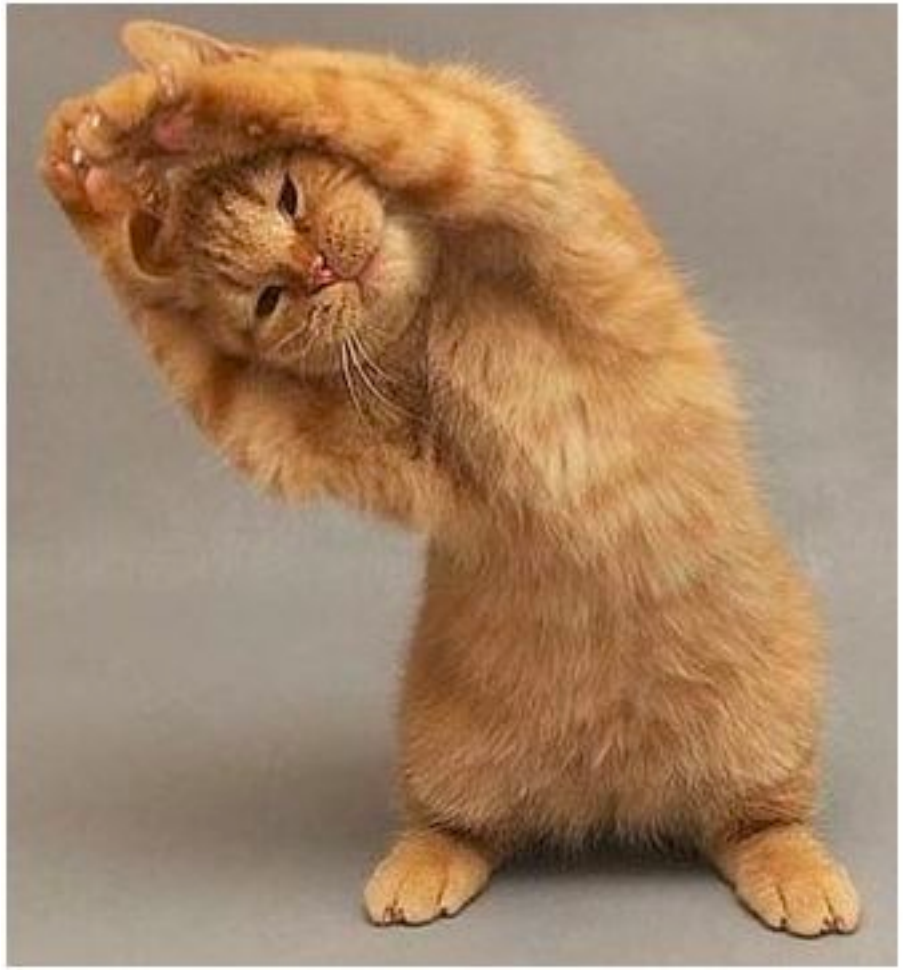
So why do we need this?

XML Summary

- NSO Stores and represents its Database in XML
- XML is NSO's 'Native' Language
- While we rarely interact directly with XML in NSO, its very important to understand how NSO is representing and navigating the data
- Our service packages will leverage XML configuration templates
- We pass variable into XML to generate config for our designs
- Knowing XML makes troubleshooting much easier

© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Stretch Break!



© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

REST APIs

Pre-Quiz: What is a REST API?

© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Building Blocks of NSO REST API

- Everything is within a hierarchy, beginning with ‘LINUX_HOSTNAME:8080/api’
- Typical flow:
  - Choose REST API client (Postman, Bash curl, python requests, etc)
  - Choose a REST API URI (API URL path), see first bullet point
  - Choose a REST API Operation (Post, Get, Put, Delete, etc.)
  - Add authentication to the request (default is Basic Auth, admin/admin)
  - Send request and check status (200, 400, 401 etc) and output

© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

REST API Key Components

- A working URI path (it looks like a website address)
- Necessary headers (like login credentials for auth, if want JSON response, add it)
- A way to deliver the rest call (use postman, curl, python requests)

© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

REST API

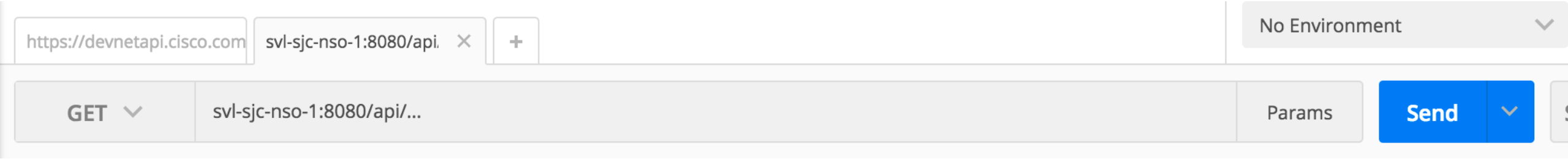
- POSTMAN: Free Chrome App - https://www.getpostman.com/apps
- Examples:
  - GET servername:8080/api/running/devices
  - GET servername:8080/api/running/devices/device/(name)/config?deep
  - POST servername:8080/api/running/devices/device/(name)/config/hostname

- For more info see NSO Docs:
  - nso_northbound-4.4.pdf – Chapter 3

© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

URI in Postman

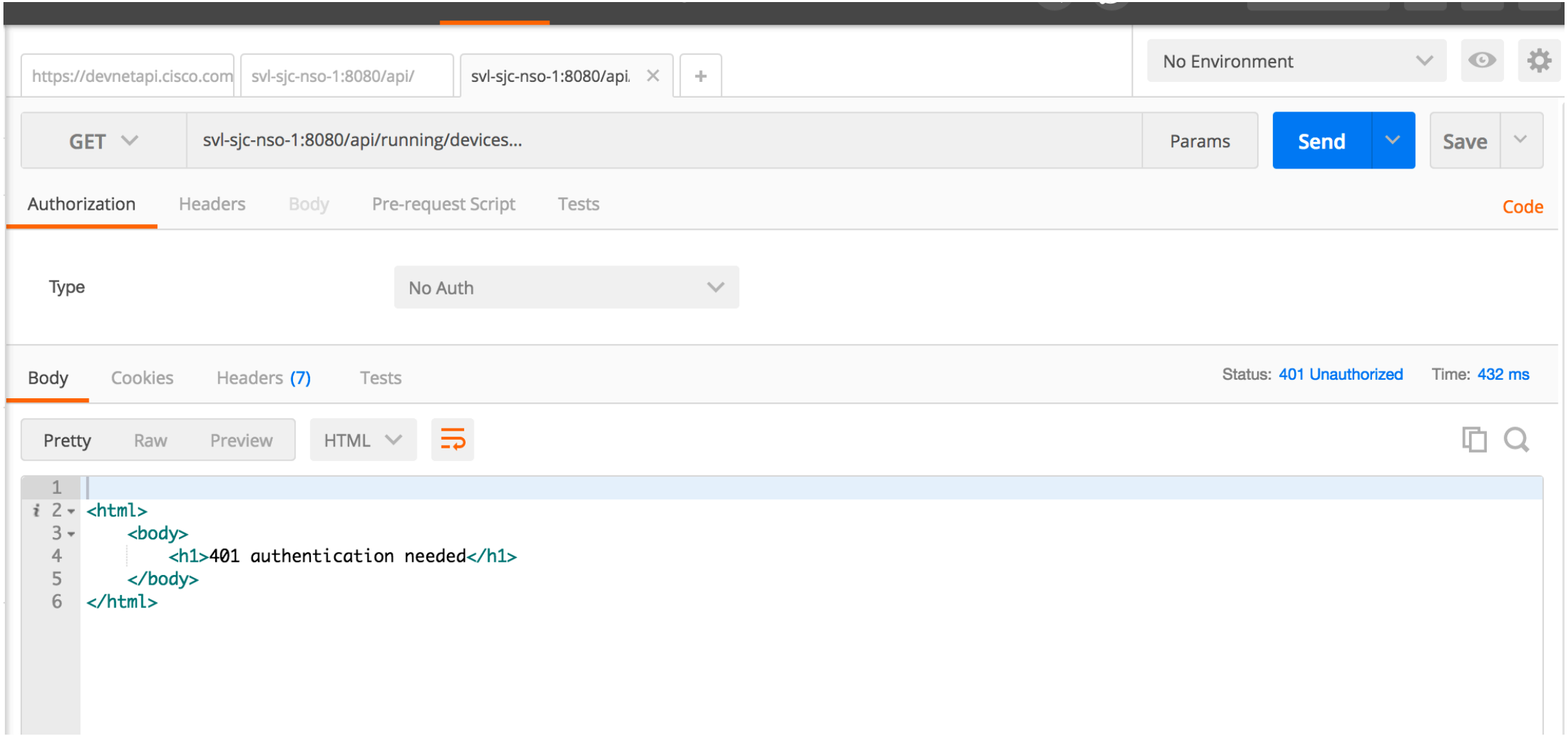
Example:
svl-sjc-nso-1:8080/api/running/devices



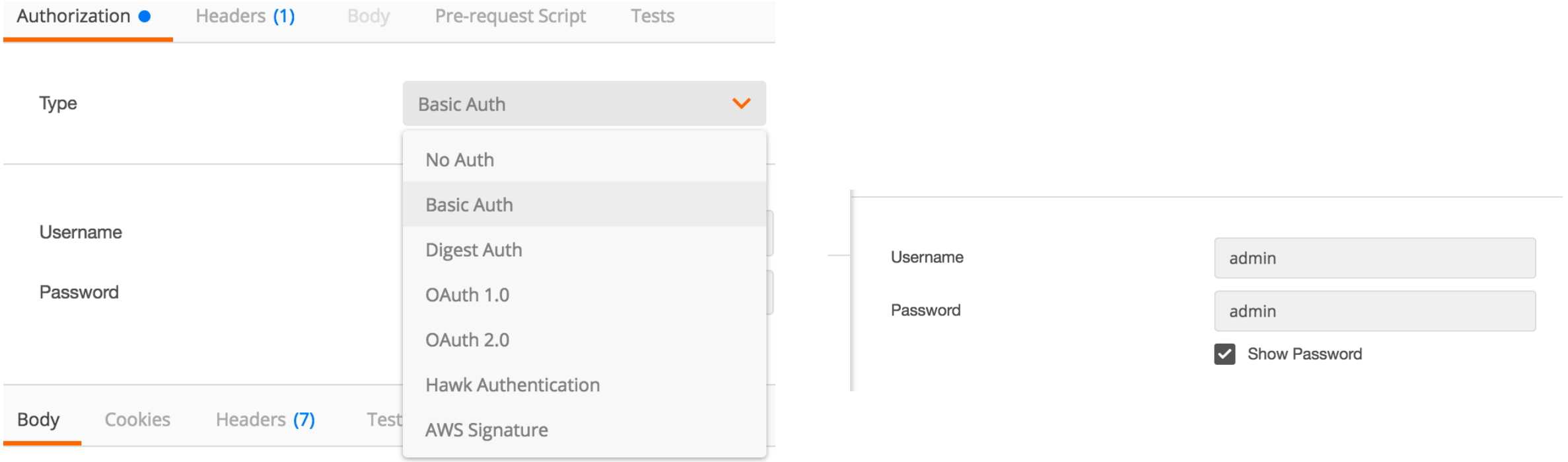
Use localhost instead of svl-sjc-nso-1 if you are on a VM

© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

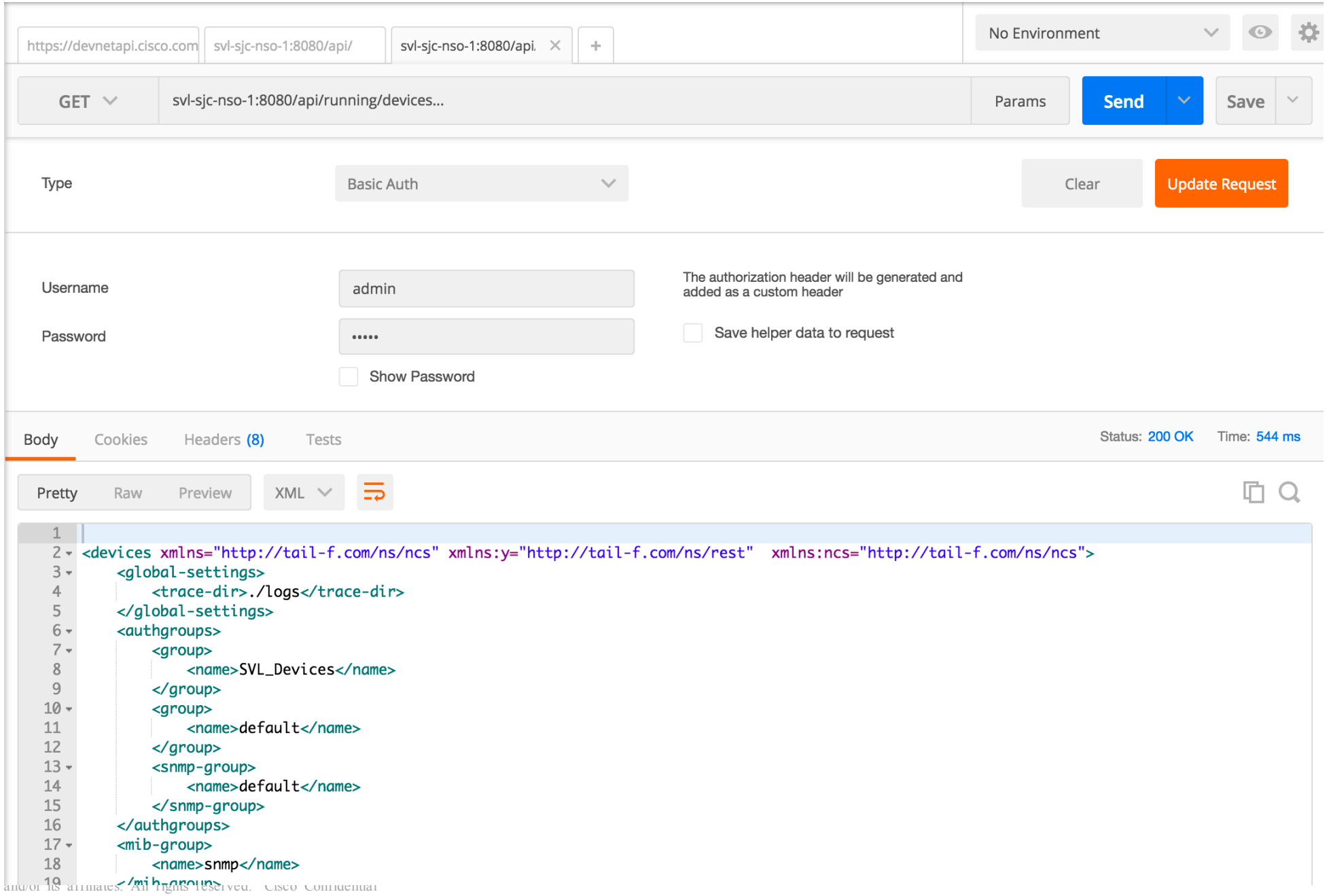
You Forgot Authentication!



Adding REST Authentication in Postman



© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Confidential



© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Now You Try

- Try using the REST API to:
  - Get a specific devices configuration
  - Get just the hostname for a device

  - Advanced:
  - Change the devices hostname via a POST call

© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

NSO Query Rest API

NSO Query API

- One of the most powerful things about a database is the ability to query it
- NSO REST API gives us a way to query and get information out of the NSO cDB

© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

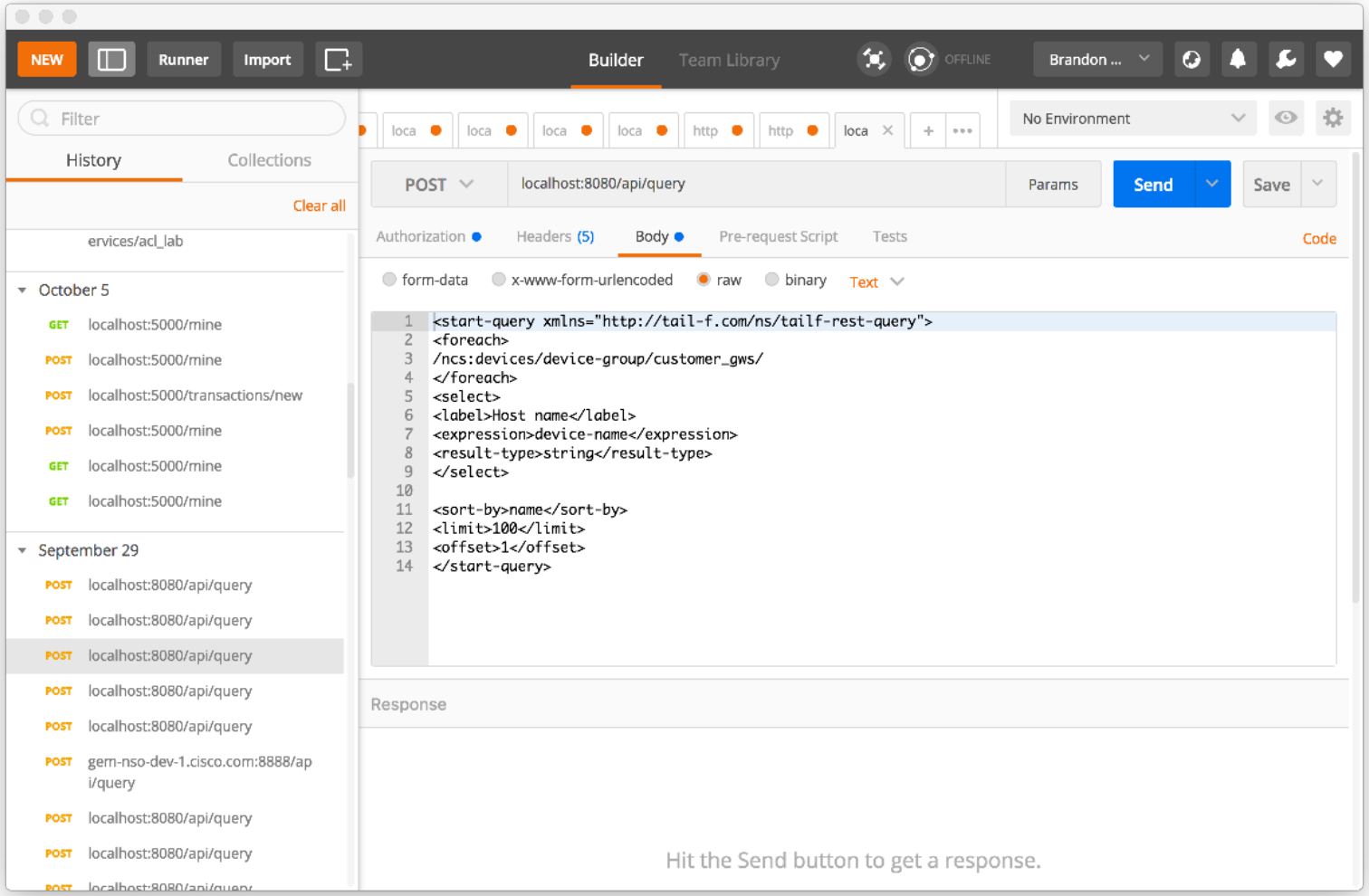
NSO Query API

- The NSO REST API also exposes a query functionality for quickly retrieving information from the cDB via xPath expressions.
- Works by POST of a payload to http:server/api/query
- A query handle ID is returned
- Results returned via POST of the handle back to the API end point
- Can send in JSON or XML
- Requires knowledge of xPath

© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Payload Structure

- Payload has two required options. Foreach and select
- Foreach is the node to iterate over
- Select is the attributes to select from it
- XPATH for both



© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

NSO Query API Wrapper

- We have been working on simplifying the experience via a python wrapper class for ‘SQL-Like’ input

```
from nso_query_tool import NsoServer, NsoQuery

server = NsoServer('server', "user", "pass")
select = ['name', "config/ios:ip/http/server", "platform/model", "platform/version"]
  _from = [{"device-group":"acc1-pl"}, {"device":"acc3-cn-sw1"}]
  where = ["config/ios:tacacs/timeout='3'", "and", "config/ios:cdp/run ='true'"]
  query = NsoQuery(server, _from=_from, select=select, where=where)
  for item in query.results:
      print item
  print query.results.length()
```

sample results:

```
{u'platform/version': u'15.2(4)M1', u'platform/model': u'CISCO2901/K9', u'name': u'acc1-pl-cs1', u'config/ios:ip/http/server': u'false'}
{u'platform/version': u'03.06.05E', u'platform/model': u'WS-C3850-48P', u'name': u'acc1-pl-sw1', u'config/ios:ip/http/server': u'true'}
{u'platform/version': u'15.5(3)S2', u'platform/model': u'ISR4451-X/K9', u'name': u'acc1-pl-wan-gw1', u'config/ios:ip/http/server': u'false'}
{u'platform/version': u'15.5(3)S2', u'platform/model': u'ISR4451-X/K9', u'name': u'acc1-pl-wan-gw2', u'config/ios:ip/http/server': u'false'}
4
```

© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Feedback - Survey