DevNet Experts.

Topic- YANG

+91 9892028199

devnetexperts@gmail.com

OVERVIEW

- What is YANG?
- What is data model?
- YANG File Structure
- Sample YANG Data Model
- JSON Encoding
- XML Encoding

What is YANG?

- Stands for ***Y***et ***A***nother ***N***ew ***G***eneration
- Basically a modeling language for network devices
- Used to define data models of the network devices
- Maintained by NETMOD - an IETF working group
- Latest version of YANG is 1.1
- Full specification is documented in **RFC 7950**

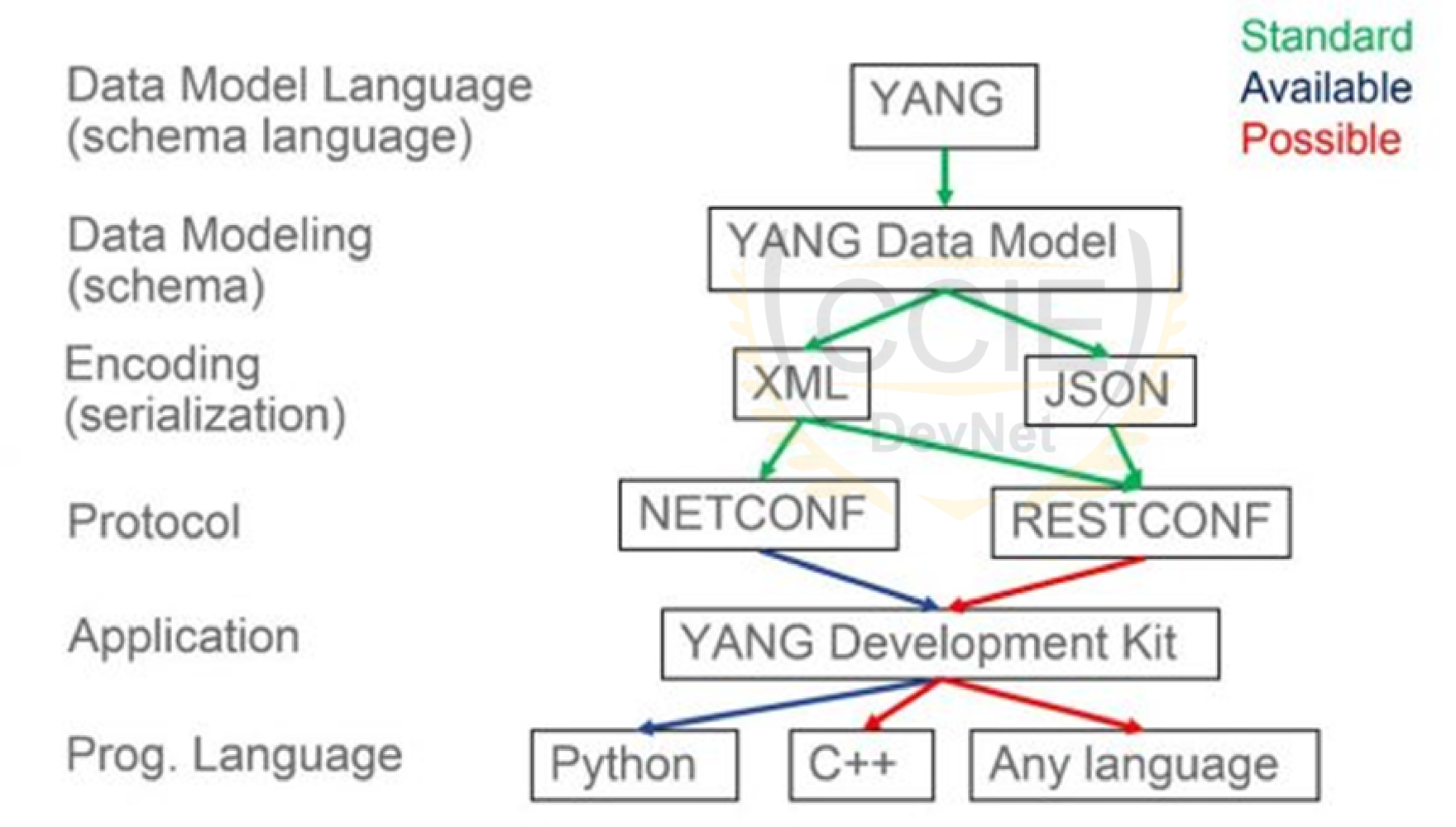
What is data model ?

- Assume that we are talking about a ***router***
- ***Router*** will have some ***interfaces*** configured
- Each ***interface*** will have an ***interface name***, an ***IP Address*** and a ***subnet mask***
- The interface will either be ***enabled*** or ***disabled***

What is Data model (cont.)

- What we just did is describe the data about a router in human readable and understandable language
- But for a computer to understand and princess this data, we need to define the same in a standard way as per certain strict syntax rules
- This makes the data to be defined in a consistent way to be processed by computers
- This standard syntax used to define data is known as ***data model***

The BIg Picture.



YANG File Structure

- YANG is a text file with extension .yang
- Starts with a keyword **module**, which is a root element
- All other content goes into this element
- A module defines a single data model
- An external module can be imported into this using **import** statement
- A submodule can be included in this using **include** statement
- YANG has many built in data types to represent test and numbers
- Derived data types can be created using **typedef** statement

YANG Derived Data Type

- Sown below is a custom or derived data type based on the built in string data type.
- It is used to represent an IP Address.

```
typedef dotted-quad {
  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])';
  }
  description
    "Four octets written as decimal numbers and
     separated with the '.' (full stop) character.";
}
```

YANG File Structure (cont.)

- YANG models data using hierarchical tree based structure using nodes
- The node types are:

1. **Leaf node** - Contains a single value of specific type
2. **Leaf-list node** - Contains a sequence of leaf nodes
3. **Container node** - Contains grouping of related nodes containing only child nodes, which can be of any of the four types of nodes
4. **List node** - contains a sequence of list entries, each of which is uniquely identified by one or more key leafs

Sample YANG Data Model.

The data being modeled using YANG can be either of the following.

- **Configuration Data**

1. Read/ Write configuration fields
2. E.g. Interface name, IP Address, Subnet mask, admin enabled/ disabled etc.

- **State Data**

1. Defined using statement config false;
2. Readonly operational data fields
3. E.g. Packet counter, Physical up/ down status

Sample Data Model - Config Data

```
list interface {
  key "name";
  leaf name {
    type string;
    mandatory "true";
    description
      "Interface name. Example value: GigabitEthernet 0/0/0";
  }
  leaf address {
    type dotted-quad;
    mandatory "true";
    description
      "Interface IP address. Example value: 10.10.10.1";
  }
  leaf subnet-mask {
    type dotted-quad;
    mandatory "true";
    description
      "Interface subnet mask. Example value: 255.255.255.0";
  }
  leaf enabled {
    type boolean;
    default "false";
    description
      "Enable or disable the interface. Example value: true";
  }
}
```

Sample Data Model - State Data

```
list interface-state {
  config false;
  key "name";
  leaf name {
    type string;
    description
      "Interface name. Example value: GigabitEthernet 0/0/0";
  }
  leaf oper-status {
    type enumeration {
      enum up;
      enum down;
    }
    mandatory "true";
    description
      "Describes whether the interface is physically up or down";
  }
}
```

XML Encoding

```xml
<data xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <interfaces xmlns="http://ultraconfig.com.au/ns/yang/ultraconfig-interfaces">
    <interface>
      <name>GigabitEthernet 0/0/0</name>
      <address>10.10.10.1</address>
      <subnet-mask>255.255.255.0</subnet-mask>
    </interface>
    <interface>
      <name>GigabitEthernet 0/0/1</name>
      <address>192.168.1.1</address>
      <subnet-mask>255.255.255.0</subnet-mask>
    </interface>
  </interfaces>
</data>
```

XML representation of an instantiation of the data model.

Python - YANG

- Python provides a package called pyang to work with YANG data models
- It can be installed from PyPI using the following command

1. pip install pyang

Python - YANG (cont.)

- YANG model can be validated using pyang as follows

1. pyang test-config.yang
2. There will be no output if validation is successful
3. In case of validation errors, output will show the error messages

```
nexadmin@DESKTOP-89IJ1T7: ~/temp/yang_test
nexadmin@DESKTOP-89IJ1T7:~/temp/yang_test$ ls
test-config.yang
nexadmin@DESKTOP-89IJ1T7:~/temp/yang_test$ pyang test-config.yang
nexadmin@DESKTOP-89IJ1T7:~/temp/yang_test$
```

```
nexadmin@DESKTOP-89IJ1T7: ~/temp/yang_test
nexadmin@DESKTOP-89IJ1T7:~/temp/yang_test$ pyang test-config.yang
test-config.yang:75: error: type "inumeration" not found in module "test-config"
nexadmin@DESKTOP-89IJ1T7:~/temp/yang_test$
```

DevNet Experts

YANG

DEMO.