

Lab – Explore YANG Models Using the pyang Tool

Objectives

Part 1: Install the pyang Python module

Part 2: Download YANG models for the IOS XE

Part 3: Use the pyang command line tool to transform the YANG models

Background / Scenario

YANG models define the exact structure, data types, syntax and validation rules for the content of messages exchanged between a managed device and another system communicating with the device. Working with files using the YANG language can be a bit overwhelming for the level of details in these files.

In this lab, you will learn how to use the open source pyang tool to transform YANG data models from files using the YANG language, into a much more easily human readable format. Using the “tree” view transformation, you will identify what are the key elements of the ietf-interfaces YANG model.

Required Resources

- Access to the Internet
- Python 3.x environment

Instructions

Part 1: Install the pyang Python module

In this part, you will install pyang module into your Python environment. Pyang is a python module that simplifies working with YANG files. The Pyang Python module comes with a pyang command line executable that transforms YANG files into a more human readable format (tree, html, etc.).

Step 1: Use pip to install pyang.

- a. Start a new Windows command prompt (cmd).
- b. Install pyang using pip in the Windows command prompt:

```
pip install --no-binary pyang pyang
```

NOTE: on mac or linux you can simply “pip install pyang” but temporarily on Windows the binary WHL file won’t include the Windows executable pyang file.

- c. Verify that pyang has been successfully installed. In the command prompt, type:

```
pyang -v
```

to display the installed pyang version.

Download YANG models for the IOS XE

Explore YANG models on the YangModels/yang GitHub repository.

- a. Using a web browser, navigate to <https://github.com/YangModels/yang>:
- b. Navigate to the vendor -> cisco -> xe -> 1693 directory. This directory represents all the YANG models that are supported in Cisco operating system IOS XE in version 16.9.3.

- c. Explore the ietf-interfaces.yang model.

Step 2: Download the YANG models for the IOS XE VM

- a. Unpack the YANG models from the official GitHub repo for cisco-xe-1693.zip archive file that contains a snapshot of the files in the GitHub repository.

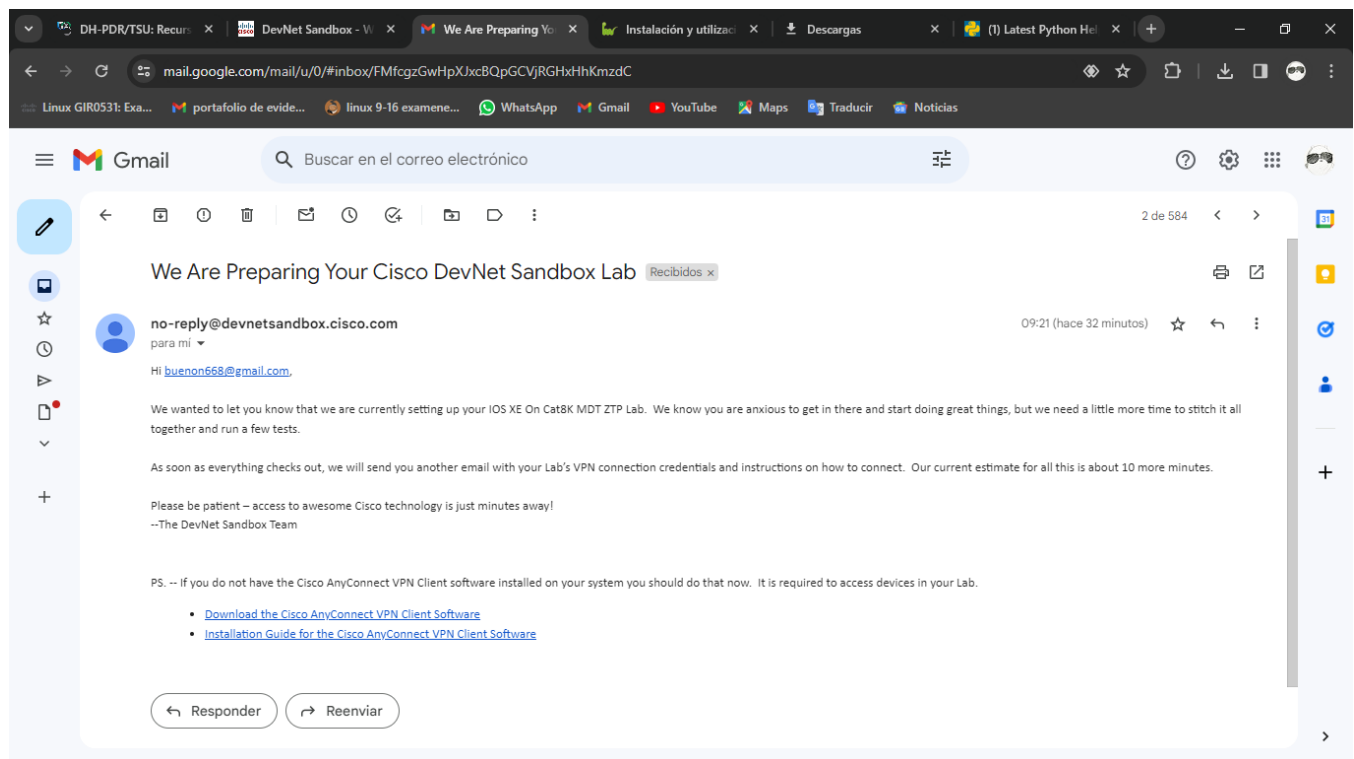
Part 2: Use the pyang command line tool to transform the YANG models

- a. Start a Windows command prompt and navigate to the directory with the extracted archive file.
- b. Use the pyang tool to transform YANG files to a human readable format, for example using the “tree” format transformation:

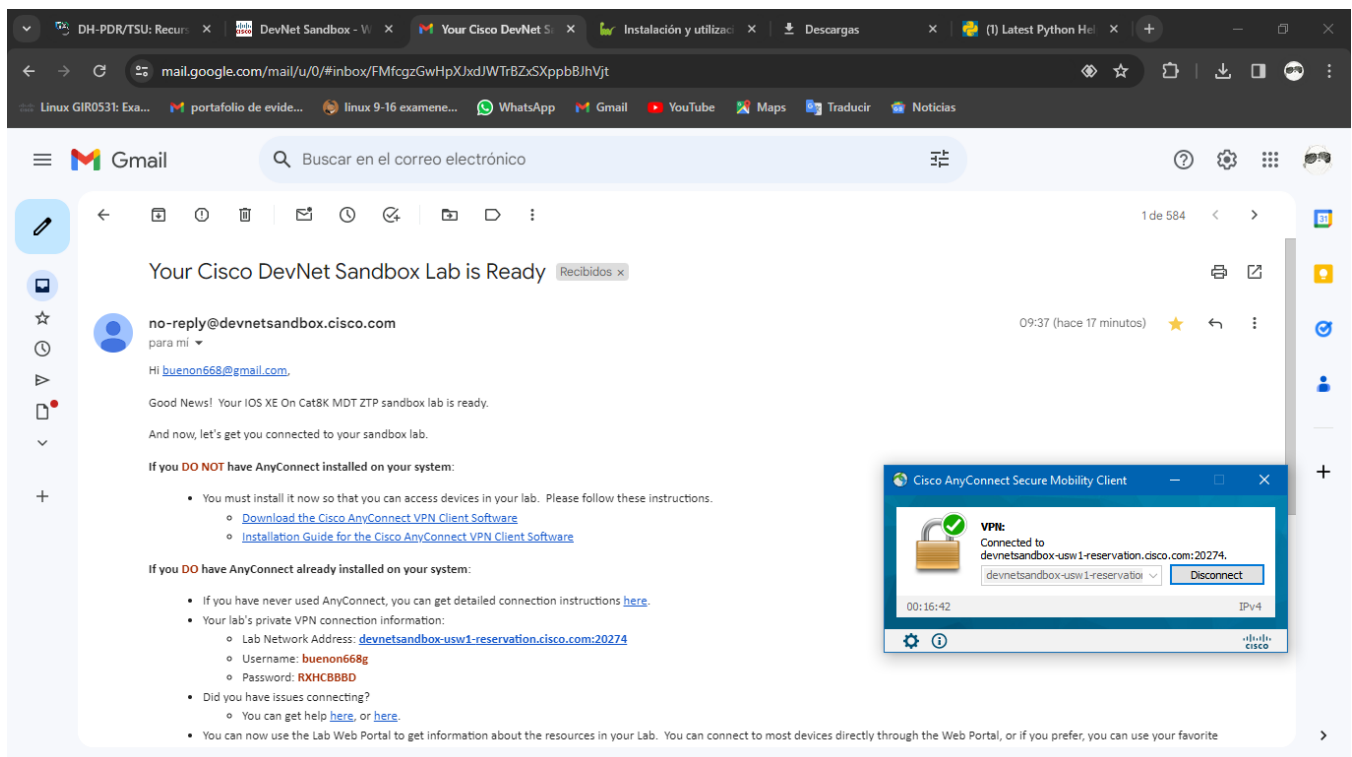
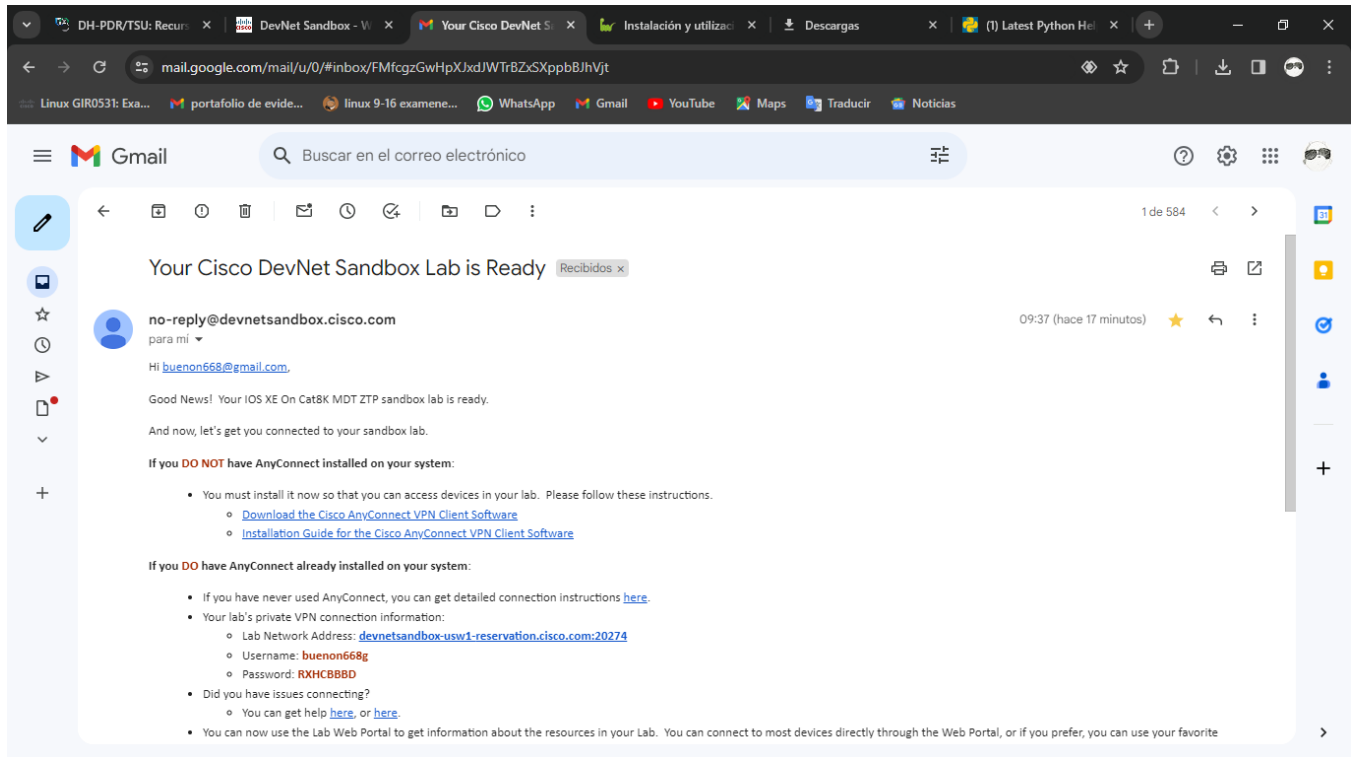
```
pyang -f tree ietf-interfaces.yang
```

- c. Explore other YANG modules, for example the Cisco Native model for CDP: Cisco-IOS-XE-cdp.yang
- d. Are there any “read only” operation data in the Cisco-IOS-XE-cdp.yang model?

- e. Is there any other YANG model that includes operational CDP data?



Lab – Explore YANG Models Using the pyang Tool



Lab – Explore YANG Models Using the pyang Tool

```
Git CMD
C:\Users\usuario>python -v
No se encontr  Python; ejecuta sin argumentos para instalar desde Microsoft Store o deshabilita este acceso directo en Configuraci n > Administrar alias de ejecuci n de la aplicaci n.

C:\Users\usuario>python --version
No se encontr  Python; ejecuta sin argumentos para instalar desde Microsoft Store o deshabilita este acceso directo en Configuraci n > Administrar alias de ejecuci n de la aplicaci n.

C:\Users\usuario>pip install --no-binary pyang pyang
Collecting pyang
  Downloading pyang-2.6.0.tar.gz (492 kB)
    ----- 492.8/492.8 kB 1.0 MB/s eta 0:00:00
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
  Preparing metadata (pyproject.toml) ... done
Collecting lxml (from pyang)
  Obtaining dependency information for lxml from https://files.pythonhosted.org/packages/77/e5/1f23e56678244258483521872507d64130cac9466902aca3f3141b8fb06b/lxml-4.9.3-cp312-cp312-win_amd64.whl.metadata
  Downloading lxml-4.9.3-cp312-cp312-win_amd64.whl.metadata (3.9 kB)
  Downloading lxml-4.9.3-cp312-cp312-win_amd64.whl (3.8 MB)
    ----- 3.8/3.8 MB 2.1 MB/s eta 0:00:00
Building wheels for collected packages: pyang
  Building wheel for pyang (pyproject.toml) ... done
  Created wheel for pyang: filename=pyang-2.6.0-py2.py3-none-any.whl size=594153 sha256=55b7524c97949b8d85268d5c18abbf95d465a4e00adf64386becd4daf1a20c9f
  Stored in directory: c:\users\usuario\appdata\local\pip\cache\wheels\d8\ae\6c\9c4bd1c1e24f5dd57ff074290f77e7d998d2b2f5030da2bf2e
Successfully built pyang
Installing collected packages: lxml, pyang
Successfully installed lxml-4.9.3 pyang-2.6.0

[notice] A new release of pip is available: 23.2.1 -> 23.3.1
[notice] To update, run: C:\Users\usuario\AppData\Local\Programs\Python\Python312\python.exe -m pip install --upgrade pip

C:\Users\usuario>
```

```
Git CMD
ModuleNotFoundError: No module named 'pkg_resources'

C:\Users\usuario>where python
C:\Users\usuario\AppData\Local\Microsoft\WindowsApps\python.exe
C:\Users\usuario\AppData\Local\Programs\Python\Python312\python.exe

C:\Users\usuario>where pyang
C:\Users\usuario\AppData\Local\Programs\Python\Python312\Scripts\pyang.exe

C:\Users\usuario>pyang -v
Traceback (most recent call last):
  File "<frozen runpy>", line 198, in _run_module_as_main
  File "<frozen runpy>", line 88, in _run_code
  File "C:\Users\usuario\AppData\Local\Programs\Python\Python312\Scripts\pyang.exe\__main__.py", line 4, in <module>
    from pyang import plugin
  File "C:\Users\usuario\AppData\Local\Programs\Python\Python312\Lib\site-packages\pyang\scripts\pyang_tool.py", line 12, in <module>
    import pkg_resources
ModuleNotFoundError: No module named 'pkg_resources'

C:\Users\usuario>pip install setuptools
WARNING: Retrying (Retry(total=4, connect=None, read=None, redirect=None, status=None)) after connection broken by 'ReadTimeoutError("HTTPConnectionPool(host='pypi.org', port=443): Read timed out. (read timeout=15)")': /simple/setuptools/
Collecting setuptools
  Obtaining dependency information for setuptools from https://files.pythonhosted.org/packages/bb/e1/ed2dd0850446b8697ad28d118df885ad04140c64ace06c4bd559f7c8a94f/setuptools-69.0.2-py3-none-any.whl.metadata
  Using cached setuptools-69.0.2-py3-none-any.whl.metadata (6.3 kB)
  Using cached setuptools-69.0.2-py3-none-any.whl (819 kB)
Installing collected packages: setuptools
Successfully installed setuptools-69.0.2

[notice] A new release of pip is available: 23.2.1 -> 23.3.1
[notice] To update, run: C:\Users\usuario\AppData\Local\Programs\Python\Python312\python.exe -m pip install --upgrade pip

C:\Users\usuario>C:\Users\usuario\AppData\Local\Programs\Python\Python312\python.exe -m pip install --upgrade pip
Requirement already satisfied: pip in c:\users\usuario\appdata\local\programs\python\python312\lib\site-packages (23.2.1)
Collecting pip
  Obtaining dependency information for pip from https://files.pythonhosted.org/packages/47/6a/453160888fab7c6a432a6e25f8afe6256d0d9f2cbd25971021da6491d899/pip-23.3.1-py3-none-any.whl.metadata
  Downloading pip-23.3.1-py3-none-any.whl.metadata (3.5 kB)
  Downloading pip-23.3.1-py3-none-any.whl (2.1 MB)
    ----- 2.1/2.1 MB 3.6 MB/s eta 0:00:00
Installing collected packages: pip
  Attempting uninstall: pip
    Found existing installation: pip 23.2.1
    Uninstalling pip-23.2.1:
      Successfully uninstalled pip-23.2.1
  Successfully installed pip-23.3.1

C:\Users\usuario>pyang -v
pyang 2.6.0

C:\Users\usuario>
```

Lab – Explore YANG Models Using the pyang Tool

```
Git CMD
C:\Users\usuario>pyang -f tree ietf-interfaces.yang
error ietf-interfaces.yang: [Errno 2] No such file or directory: 'ietf-interfaces.yang'

C:\Users\usuario>pyang -f tree ietf-interfaces.yang
error ietf-interfaces.yang: [Errno 2] No such file or directory: 'ietf-interfaces.yang'

C:\Users\usuario>pwd
"pwd" no se reconoce como un comando interno o externo,
programa o archivo por lotes ejecutable.

C:\Users\usuario>pwd
"pwd" no se reconoce como un comando interno o externo,
programa o archivo por lotes ejecutable.

C:\Users\usuario>pyang -f tree ietf-interfaces.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
C:\Users\usuario>
```