

Übung zu YANG

Aus YANG-Modellen werden letztendlich XML bzw. JSON-Daten erzeugt, die mit den Vorgaben konsistent sind. In der Realität werden diese dann weiterverwertet, um Content oder Filterausdrücke zu erzeugen, die dann bei Zugriffen mit NETCONF, RESTCONF, etc. mitgegeben werden.

In dieser Übung soll dies mit Hilfe von pyang aus einem vorgegebenen YANG-Modell für eine (fiktive) Interface-Konfiguration eines Routers geschehen. Wechseln sie dazu in der Shell in das Directory Desktop/Kursfiles/YANG. Hier befinden sich alle für die Übung erforderlichen Dateien.

- 1.) Geben sie mit cat das YANG-Modell interface-config.yang auf der Shell aus und analysieren Sie dessen Struktur.
- 2.) Entwickeln Sie mit pyang ein XML-Skelett:

```
pyang -f sample-xml-skeleton interface-config.yang > interface_xml.xml
```

Das XML-Skelett wird direkt in den File interface_xml.xml eingetragen. Öffnen Sie den File mit dem Text-Editor Pluma, um ihn verändern zu können.

Die Werte der Leaf Nodes sind hier offen, sodass diese als Empty Elements zu Tage treten. Entwickeln Sie daraus die Konfiguration für 2 Interfaces und vergeben Sie den Leaf Nodes Werte wie folgt:

```
Name: GigabitEthernet 0/0
MTU: 1500
Speed: 1000
IP-Adressen: 10.10.1.1/24
Admin up: True
```

```
Name: GigabitEthernet 0/1
MTU: 1500
Speed: 1000
IP-Adressen: 10.10.2.1/24
Admin up: False
```

(Str-S zum Abspeichern)

- 3.) Installieren sie in der Shell das pyang Plugin pyangbind:

```
sudo pip3 install pyangbind
```

(Die Library sollte auf der VM installiert sein.)

- 4.) Unter `interface_config_api.py` ist das mit `pyangbind` aus dem YANG-Modell erzeugte Python API zu finden. Dieses ist im Vorfeld schon für diese Übung vorbereitet worden und kann direkt verwendet werden. Damit lässt sich eine mit dem YANG-Modell harmonisierende Datenstruktur erzeugen. Diese liegt zunächst als ein System baumartig verlinkter Objekte vor, kann aber am Ende als JSON-Struktur abgerufen werden. In der Datei `example_for_pyangbind.py` ist dazu ein Beispiel zu finden. Geben Sie den Inhalt dieser Datei mit `cat` auf der Shell aus.

Die Wurzel der gesamten Datenstruktur wird hier durch das Objekt `int1` repräsentiert. Darin werden zwei Listenelemente mit den Keys `GigabitEthernet 0/0` und `GigabitEthernet 0/1` erzeugt. Es wird die Python-typische dot-Notation verwendet, um einen Pfad im YANG-Modell zu dem List Node zu identifizieren. Darauf wird die Methode `add()` angewendet, die als Argument den Key Value des zu erzeugenden Listenelements übernimmt, und die gewünschten Listenelemente erzeugt. Andere Leaf Nodes werden zunächst nicht mit Werten versehen. Die dadurch erzeugte Datenstruktur wird mit der Funktion `pybindJSON.dumps()` in JSON transformiert und dann mittels `print()` auf der Shell ausgegeben. Führen Sie das Skript aus:

```
python3 example_for_pyangbind.py
```

- 5.) Ergänzen Sie das Skript mit dem Pluma Editor um Anweisungen zur Erzeugung weiterer Leaf Nodes in der Datenstruktur nach dem folgenden Muster:

```
int1.interfaces.interface['<Key-Value>'].<Leaf-Name> = <Value>
```

Achtung: Der Leaf `ip-address` im YANG-Modell muss unter Python in `ip_address` umbenannt werden und der `<Value>` in `'...'` übergeben werden.

Verwenden Sie hier als `<Value>` die Werte aus Teil 2.) dieser Übung. Führen Sie dann das Skript aus und analysieren Sie die Rückgabe.