

## Übung zu YANG

Aus YANG-Modellen werden letztendlich XML bzw. JSON-Daten erzeugt, die mit den Vorgaben konsistent sind. In dieser Übung soll dies mit Hilfe von pyang aus einem vorgegebenen YANG-Modell für eine (fiktive) Konfiguration eines Routers geschehen. Wechseln sie dazu in der Shell in das Directory NETY/pyangbind. Hier befinden sich alle für die Übung erforderlichen Dateien.

### Vorbereitung

- 1.) Bauen Sie über myExperTeach die Verbindung zur VM auf.
- 2.) Vorbereitende Maßnahme für die praktischen Übungen (falls nicht bereits geschehen oder vom Trainer erledigt): Starten Sie eine Linux Shell (Desktop Icon MATE-Terminal). Für die Übungen werden einige Dateien benötigt, die aus einem Repository bei GitHub geclonet werden:

```
git clone https://github.com/Wotan242/NETY.git
```

Die Dateien sollten sich nun in dem Ordner /NETY befinden.

### Teil1: XML erzeugen

- 1.) Geben sie mit cat das YANG-Modell routerconfig.yang auf der Shell aus und analysieren Sie dessen Struktur gedanklich.
- 2.) Lassen Sie sich mit pyang die Struktur des YANG-Modells als Tree visualisieren:

```
pyang -f tree routerconfig.yang
```

- 3.) Entwickeln Sie mit pyang ein XML-Skelett:

```
pyang -f sample-xml-skeleton routerconfig.yang > routerconfig_xml.xml
```

Das XML-Skelett wird direkt in den File routerconfig\_xml.xml eingetragen. Öffnen Sie den File mit dem Sublime-Editor, um ihn verändern zu können:

```
subl routerconfig_xml.xml
```

Die Werte der Leaf Nodes sind hier offen, sodass diese als Empty Elements zu Tage treten. Bei einer Leaf-List wird freundlicherweise darauf hingewiesen, dass mehrere Werte möglich sind. Entwickeln Sie (unter Zuhilfenahme von Copy and Paste) daraus die Konfiguration eines Routers mit folgenden Vorgaben:

## System-Konfiguration

Hostname: Router\_1

## Interface-Konfiguration:

Name: Loopback0  
IP-Adressen: 1.1.1.1  
Netzmaske: 255.255.255.255  
Admin up: True

Name: GigabitEthernet 0/0  
MTU: 1500  
Speed: 1000  
IP-Adressen: 10.10.1.1  
Netzmaske: 255.255.255.0  
Admin up: True

Name: GigabitEthernet 0/1  
MTU: 1500  
Speed: 1000  
IP-Adressen: 10.10.2.1  
Netzmaske: 255.255.255.0  
Admin up: False

## OSPF-Konfiguration:

RID: 1.1.1.1  
Area 0:  
  Interface GigabitEthernet 0/0  
  Interface Loopback0  
Area 1:  
  Interface GigabitEthernet 0/1

(Str-S zum Abspeichern)

## Teil2: JSON erzeugen

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

```
sudo pip3 install pyangbind
```

- 5.) Unter routerconfigbinding.py ist das mit pyangbind aus dem YANG-Modell erzeugte Python API zu finden. Dessen Compilierung aus der YANG-Vorlage ist bereits im Vorfeld erledigt worden, damit Sie sich nicht mit den etwas umfänglichen und kryptischen Kommandos auf der Shell plagen müssen.  
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 (XML geht auch) abgerufen werden. In der Datei `example_for_pyangbind_router.py` ist dazu ein Beispiel mit einer halbfertigen Konfiguration zu finden. Geben Sie den Inhalt dieser Datei mit `cat` auf der Shell aus.

Die Wurzel der gesamten Datenstruktur wird hier durch das Objekt `router1` repräsentiert. Darin werden z.B. 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.:

```
router1.router.interfaces.interface.add('GigabitEthernet 0/0')
```

Ähnliches gilt für die Areas in der OSPF-Konfiguration.

Einige Leaf Nodes haben schon Werte bekommen. Dies erfolgt nach dem Muster

```
router1.router.interfaces.interface['<Key-Value> '].<Leaf-Name> = <Value>
```

```
(<Leaf-Name> = mtu, speed, ipadress, ...)
```

bzw. für die Leaf-List in der OSPF-Konfiguration

```
router1.router.ospf.area['<Key-Value> '].interface = [<Value1>, <Value2>]
```

Die dadurch erzeugte Datenstruktur wird mit der Funktion `pybindJSON.dumps()` in JSON transformiert und dann vermittels `print()` auf der Shell ausgegeben. Führen Sie das Skript aus:

```
python3 example_for_pyangbind_router.py
```

(Sollte das mit `python3` nicht funktionieren, dann versuchen Sie es mit `python`.)

- 6.) Ergänzen Sie das Skript um Anweisungen zur Erzeugung weiterer Leaf Nodes nach den Vorgaben aus Schritt 3.) dieser Übung. Führen Sie dann das Skript aus und analysieren Sie die Rückgabe.
- 7.) Zusatzaufgabe: Übergeben Sie für den Leaf Node `ipadress` eine nicht mit dem Format einer IP-Adresse harmonisierende Eingabe, z.B. `10.10.1.1.1000`. Wie reagiert Python nun beim Aufruf des Skripts `example_for_pyangbind_router.py`? Analysieren Sie die Fehlermeldungen, insbesondere die Botschaften unter `ValueError`.