

Travaux pratiques - Explorer les modèles YANG

Objectifs

Partie 1 : Lancer la DEVASC VM

Partie 2 : Explorer un modèle YANG sur GitHub

Partie 3 : Explorer un modèle YANG à l'aide de pyang

Contexte/scénario

Les modèles YANG définissent la structure exacte, les types de données, la syntaxe et les règles de validation du contenu des messages échangés entre un périphérique géré et un autre système communiquant avec le périphérique. Travailler avec des fichiers en utilisant le langage YANG peut être un peu écrasant pour le niveau de détails dans ces fichiers.

Dans ce laboratoire, vous apprendrez à utiliser l'outil **pyang** open source pour transformer les modèles de données YANG à partir de fichiers utilisant le langage YANG, en un format beaucoup plus facile à lire. En utilisant la transformation de vue «tree», vous identifierez les éléments clés du modèle YANG ietf-interfaces.

Les Ressources requises

- 1 PC avec le système d'exploitation de votre choix
- Boîte virtuelle ou VMWare
- Machine virtuelle DEVASC

Instructions

Partie 1 : Lancer la VM DEVASC

Si vous n'avez pas encore terminé le **TP - Installez le DEVASC-LAB**, faites-le maintenant. Si vous avez déjà terminé ce laboratoire, lancez le DEVASC VM maintenant.

Partie 2 : Explorer un modèle YANG sur GitHub

Dans cette partie, vous allez installer le module pyang dans votre VM DEVASC et explorer comment il transforme les fichiers YANG. Pyang simplifie le travail avec les fichiers YANG. Le module est livré avec un exécutable en ligne de commande pyang qui transforme les fichiers YANG en un format plus lisible par l'homme.

Étape 1: Explorez les modèles Cisco IOS XE YANG dans le référentiel GitHub.

- Ouvrez Chromium et accédez à <https://github.com/YangModels/yang>.
- Sous la branche **master**, accédez aux modèles YANG pour la version 16.9.3 de Cisco IOS XE en cliquant sur les répertoires suivants : **vendor > cisco > xe > 1693**.
- Faites défiler ci-dessous tous les modèles Cisco YANG et trouvez où commencent les modèles IETF. Cherchez **ietf-interfaces.yang**.
- Cliquez sur **ietf-interfaces.yang** et faites défiler tous les nœuds de conteneur, les nœuds de feuille et les nœuds de liste. Si vous êtes familier avec la sortie de la commande IOS show interfaces, vous devez reconnaître certains ou tous les nœuds. Par exemple, autour de la ligne 221, vous verrez la feuille activée.

```
leaf enabled {
  type boolean;
  default "true";
  description
    "This leaf contains the configured, desired state of the
    interface
    Systems that implement the IF-MIB use the value of this
    leaf in the 'running' datastore to set
    IF-MIB.ifAdminStatus to 'up' or 'down' after an ifEntry
    has been initialized, as described in RFC 2863.
    Changes in this leaf in the 'running' datastore are
    reflected in ifAdminStatus, but if ifAdminStatus is
    changed over SNMP, this leaf is not affected.";
  reference
    "RFC 2863: The Interfaces Group MIB - ifAdminStatus";
}
```

Étape 2: Copiez le modèle ietf-interfaces.yang dans un dossier sur votre machine virtuelle.

- Ouvrez le code VS.
- Cliquez sur **Fichier > Ouvrir le dossier...** et accédez au répertoire **devnet-src**.
- Cliquez sur **OK**.
- Ouvrez une fenêtre de terminal dans VS Code : **Terminal > New Terminal**.
- Créez un sous-répertoire appelé **pyang** dans le répertoire **/devnet-src**.

```
devasc @labvm : ~/labs/devnet-src$ mkdir pyang
devasc@labvm:~/labs/devnet-src$
```

- Retournez à l'onglet Chromium où le modèle **ietf-interfaces.yang** est toujours ouvert. Faites défiler vers le haut, si nécessaire, puis cliquez sur Raw pour afficher uniquement les données du modèle YANG.
- Sélectionnez et copiez l'URL.
- Dans le terminal, allez dans le dossier pyang.
- Utilisez **wget** pour enregistrer le fichier ietf-interfaces.yang brut.

```
devasc@labvm:~/labs/devnet-src/pyang$ wget
https://raw.githubusercontent.com/YangModels/yang/master/vendor/cisco/xe/1693
/ietf-interfaces.yang
--2020-06-22 20:42:20--
https://raw.githubusercontent.com/YangModels/yang/master/vendor/cisco/xe/1693/ietf-
interfaces.yang
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 151.101.0.133,
151.101.192.133, 151.101.128.133, ...
Connecting to raw.githubusercontent.com
(raw.githubusercontent.com)|151.101.0.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 24248 (24K) [text/plain]
Saving to: 'ietf-interfaces.yang'

ietf-interfac 100% 23.68K --.-KB/s in 0.05s

2020-06-22 20:42:21 (439 KB/s) - 'ietf-interfaces.yang' saved [24248/24248]
```

```
devasc@labvm:~/labs/devnet-src/pyang$
```

Vous avez maintenant une version locale du modèle **ietf-interfaces.yang** que vous pouvez manipuler avec **pyang**.

Partie 3 : Explorez un modèle YANG en utilisant pyang

Dans cette partie, vous allez installer le module **pyang** dans votre VM DEVASC et explorer comment il transforme le modèle YANG que vous avez copié à partir de GitHub. Pyang simplifie le travail avec les fichiers YANG. Le module est livré avec un exécutable en ligne de commande **pyang** qui transforme les fichiers YANG en un format plus lisible par l'homme.

Étape 1: Vérifiez que pyang est installé et à jour.

- En code VS, ouvrez une fenêtre de terminal.
- Vérifiez que pyang est déjà installé avec la commande **pyang -v**. Votre numéro de version peut être différent de celui indiqué ici. Vous pouvez également

```
devasc@labvm:~/labs/devnet-src$ pyang -v
pyang 2.2.1
devasc@labvm:~/labs/devnet-src$
```

- (Facultatif) Vous pouvez vérifier que vous disposez des dernières mises à jour pyang à l'aide de la commande **pip3** suivante. Toutes les mises à jour après l'écriture de ce laboratoire seront téléchargées et installées.

```
devasc@labvm:~/labs/devnet-src$ pip3 install pyang --upgrade
Requirement already up-to-date: pyang in ./local/lib/python3.8/site-packages (2.2.1)
Requirement already satisfied, skipping upgrade: lxml in ./local/lib/python3.8/site-packages (from pyang) (4.5.0)
devasc@labvm:~/labs/devnet-src$
```

Étape 2: Transformez le modèle ietf-interfaces.yang.

- Accédez au répertoire **pyang**.

```
devasc@labvm:~/labs/devnet-src$ cd pyang
devasc@labvm:~/labs/devnet-src/pyang$
```

- Entrez **pyang -h** | plus pour explorer les options de transformation du modèle YANG. Recherchez l'option **-f** comme indiqué ci-dessous. Vous utiliserez l'option de mise en forme de l' **arborescence** .

```
devasc@labvm:~/labs/devnet-src/pyang$ pyang -h | more
Usage: pyang [options] [<filename>...]
```

Validates the YANG module in <filename> (or stdin), and all its dependencies.

Options :

-h, --help Show this help message and exit
-v, --version Show version number and exit

<output omitted>

-f FORMAT, --format=Format

Convert to FORMAT. Supported formats are: yang, yin, dsdl, jstree, jsonxsl, capacit, identifiants, jtox, uml, name, omni, **tree**, depend, sample-xml-skeleton

<output omitted>

```
devasc@labvm:~/labs/devnet-src/pyang$
```

- c. Transformez le modèle **ietf-interfaces.yang** en un format d'arborescence avec la commande suivante. Notez que la **feuille activée** (leaf enabled) est beaucoup plus facile à trouver et à lire dans ce format.

```
devasc@labvm:~/labs/devnet-src/pyang$ pyang -f tree ietf-interfaces.yang
```

```
ietf-interfaces.yang:6: error: module "ietf-yang-types" not found in search path
```

```
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-état
    +--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
```

```
devasc@labvm:~/labs/devnet-src/pyang$
```