gNMI konfigurieren und pYANG in IOS XR implementieren

Inhalt

Einleitung

In diesem Dokument wird eine Kurzbeschreibung von gNMI in Cisco IOS® XR sowie die Verwendung von PYANG und die Überprüfung von Modellbäumen beschrieben.

Voraussetzungen

Anforderungen

Cisco empfiehlt, dass Sie über Kenntnisse in folgenden Bereichen verfügen:

- Cisco IOS XR-Plattform.
- Python.
- Netzwerkmanagement-Protokolle.

Verwendete Komponenten

Dieses Dokument ist nicht auf bestimmte Hardwareversionen beschränkt, sondern gilt für 64-Bit-Versionen (eXR).

Die Informationen in diesem Dokument beziehen sich auf Geräte in einer speziell eingerichteten Testumgebung. Alle Geräte, die in diesem Dokument benutzt wurden, begannen mit einer gelöschten (Nichterfüllungs) Konfiguration. Wenn Ihr Netzwerk in Betrieb ist, stellen Sie sicher, dass Sie die möglichen Auswirkungen aller Befehle kennen.

Hintergrundinformationen

gNMI-Definition

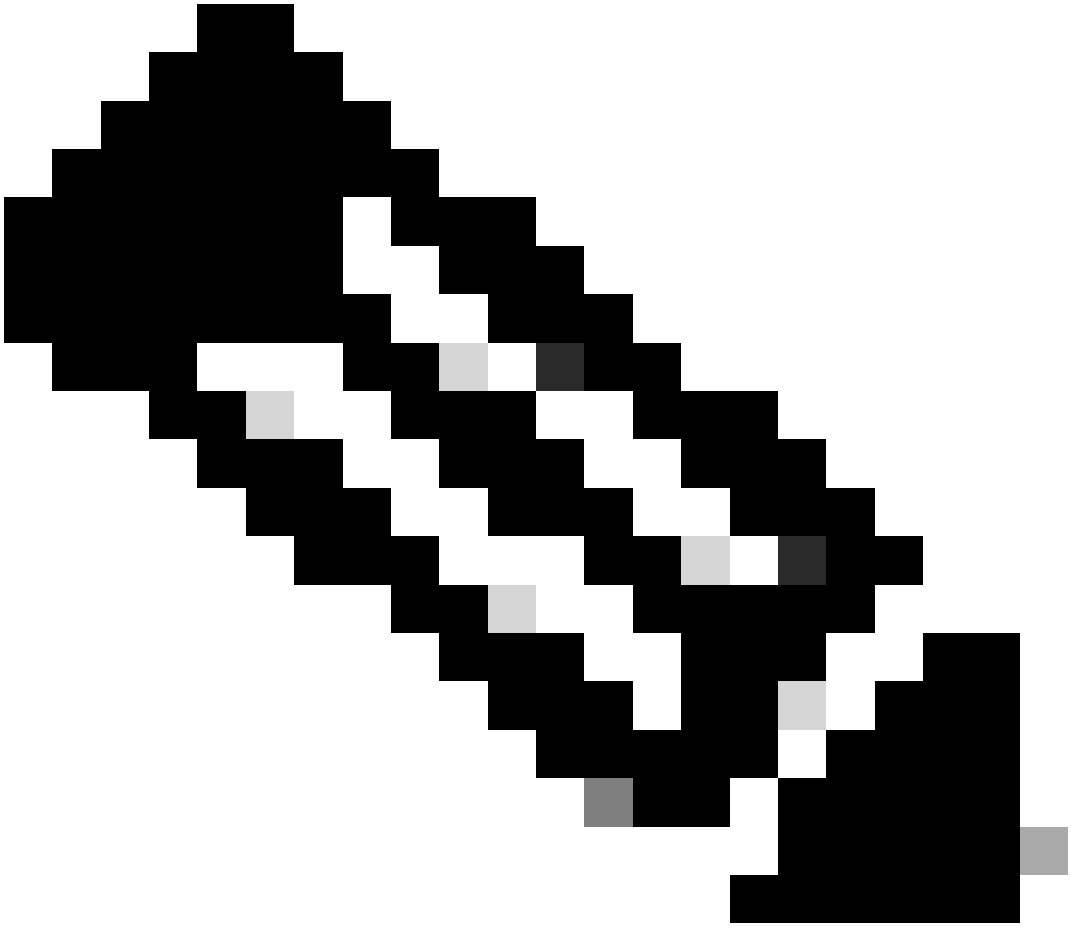
Insgesamt gibt es verschiedene Netzwerkkonfigurationsprotokolle, unter anderem von NETCONF, RESTCONF, gNMI (Google Remote Procedure Calls (gRPC), gRPC Network Management Interface). Diese Modelle werden für die Konfiguration der Netzwerkgeräte verwendet und dienen stets der Automatisierung mechanischer Prozesse.

Diese Protokolle verwenden unterschiedliche Datenmodelle, damit die Benutzer verstehen können, was der Netzwerkgeräteprozess verarbeitet, d. h. es handelt sich um eine strukturierte Information, ein Schema, die Informationen normalisiert und wie sie vom Gerät, in diesem Fall vom Router, verbraucht werden.

gNMI überwacht die Datenverarbeitung und stellt RPC (Remote Procedure Calls) bereit, um die verschiedenen Geräte im Netzwerk zu steuern.

gNMI verfügt über vier Funktionen:

- Funktionen: gNMI fragen den Router die Modelle, die im Router installiert sind, dies wird weiter in diesem Dokument erläutert.
- Abrufen: Jede Leaf-Komponente im Datenbaum kann für den Router angefordert werden. Bei diesem Vorgang werden die angeforderten Informationen angefordert.
- Set: Leafs werden als Variablen betrachtet, was ihnen die Änderungsfunktionen bereitstellt, Set Operation Assist on this allow the user to update a value in the data model.
- Abonnieren: Diese Funktion wird in der Telemetrie verwendet und unterstützt Sie beim Abrufen von Daten aus einem bestimmten Modul im Modell.

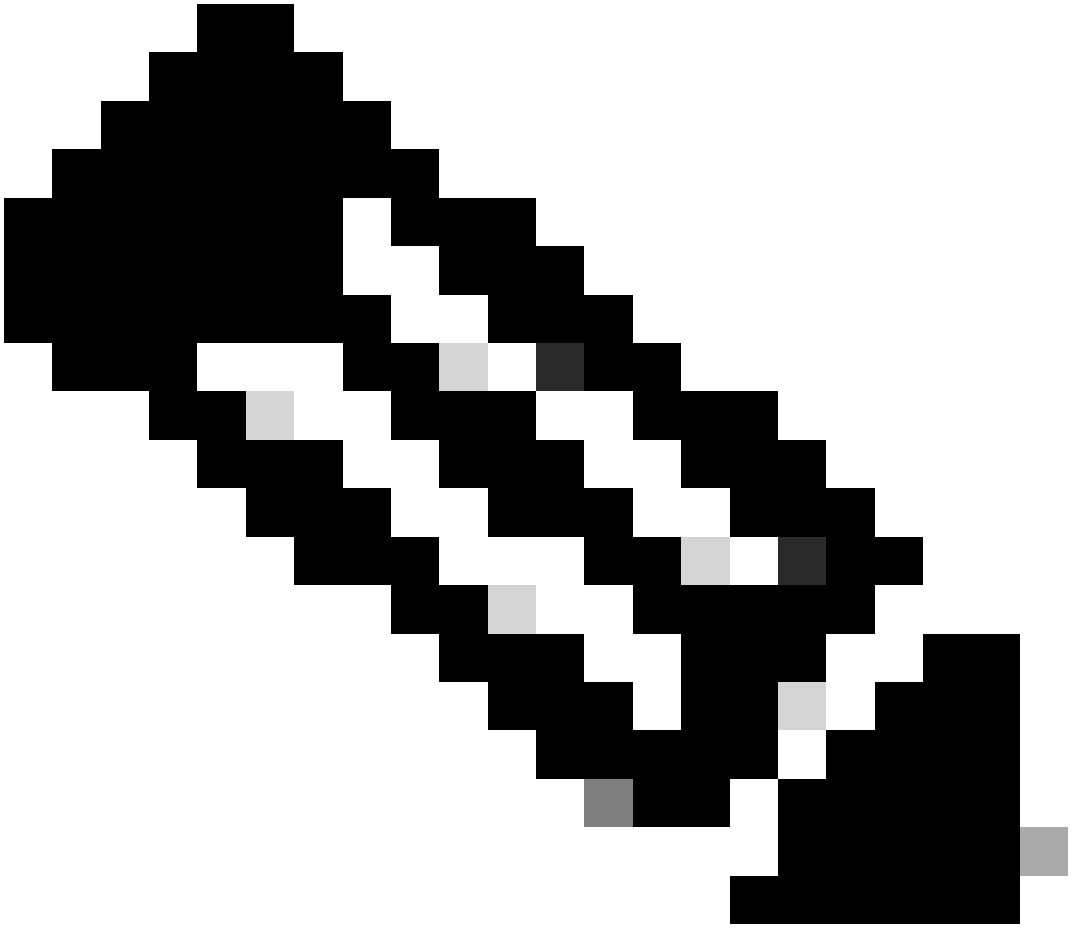


Hinweis: Cisco hat zahlreiche Informationen zu diesem Thema weitergegeben. Weitere Informationen von gRPC erhalten Sie über den folgenden Link: xrdocs blog - OpenConfig gNMI

gNMI-Funktionen

Netzwerkmanagement-Protokoll	gNMI
Genutzter Transport	HTTP/2
Unterstützt von	Kreditorenneutral
Kodierung	Proto Buff

Proto Buff ist die sprachneutrale, plattformneutrale Methode zur Deserialisierung und Serialisierung von Daten zwischen zwei Geräten, bei der jede Anforderung eine Antwort hat.



> Hinweis: Für weitere Informationen zu gRCP und Proto Buff klicken Sie auf den nächsten Link: grpc Guide.

---

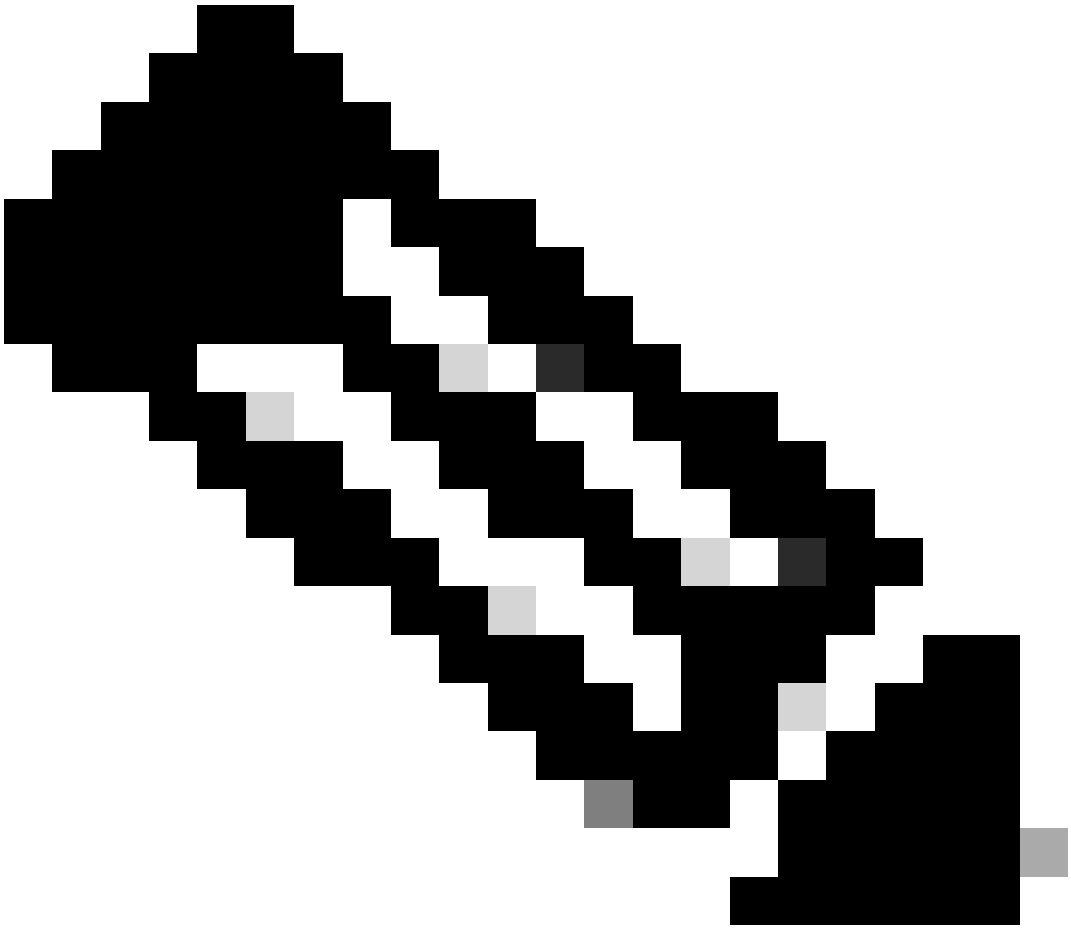
gNMI-Basiskonfiguration in Cisco IOS XR

Der nächste Schritt ist die Basiskonfiguration für den Router:

```
RP/0/RSP0/CPU0:XR(config)#grpc
RP/0/RSP0/CPU0:XR(config-grpc)#address-family ipv4
RP/0/RSP0/CPU0:XR(config-grpc)#max-request-total 256
RP/0/RSP0/CPU0:XR(config-grpc)#max-request-per-user 32
```

```
grpc
 address-family ipv4
 max-request-total 256
```

```
max-request-per-user 32
```



> Hinweis: Ein Port kann basierend auf dem Setup konfiguriert werden, der Standard, ohne TLS zu verwenden ist 57400, für weitere Informationen klicken Sie auf: github - grpc Getting Started

---

pYANG als Validator

pYANG ist ein YANG-Validator, der in Python geschrieben wurde. Diese Bibliothek für Python Unterstützung bei der Überprüfung der YANG-Modelle und auch, sie zu kennen.

Damit dies wie in der Dokumentation (pYANG-Dokumentation) ausgeführt werden kann, wird empfohlen, eine virtuelle Umgebung im Computer zu erstellen.

Ausführung der venv-Dokumentation in der virtuellen Umgebung

Folgendes ist erforderlich:

```
python -m venv <name of the directory>
```

Zum Beispiel (im MacOS-Terminal):

```
% mkdir test
% cd test
% python3 -m venv virtual_env
% ls
virtual_env
```

Um PYANG in dieser virtuellen Umgebung zu installieren, kopieren Sie das Verzeichnis, und fügen Sie den nächsten Befehl ein:

```
% cd virual_env
% git clone https://github.com/mbj4668/pyang.git
% cd pyang
% pip install -e .
```

Für diese Demonstration wurde python3 pip verwendet. Sobald pip install -e ausgegeben wurde, aktivieren Sie venv: source <Verzeichnis der virtuellen Umgebung>/bin/activate (für MacOS).

```
% source virtual_env/bin/activate
```

```
% python3 -m pip install pyang
Collecting pyang
  Downloading pyang-2.6.0-py2.py3-none-any.whl (594 kB)
     |████████████████████████████████| 594 kB 819 kB/s
Collecting lxml
  Downloading lxml-5.1.0-cp39-cp39-macosx_11_0_arm64.whl (4.5 MB)
     |████████████████████████████████| 4.5 MB 14.2 MB/s
Installing collected packages: lxml, pyang
Successfully installed lxml-5.1.0 pyang-2.6.0
```

```
% pyang -h
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
```

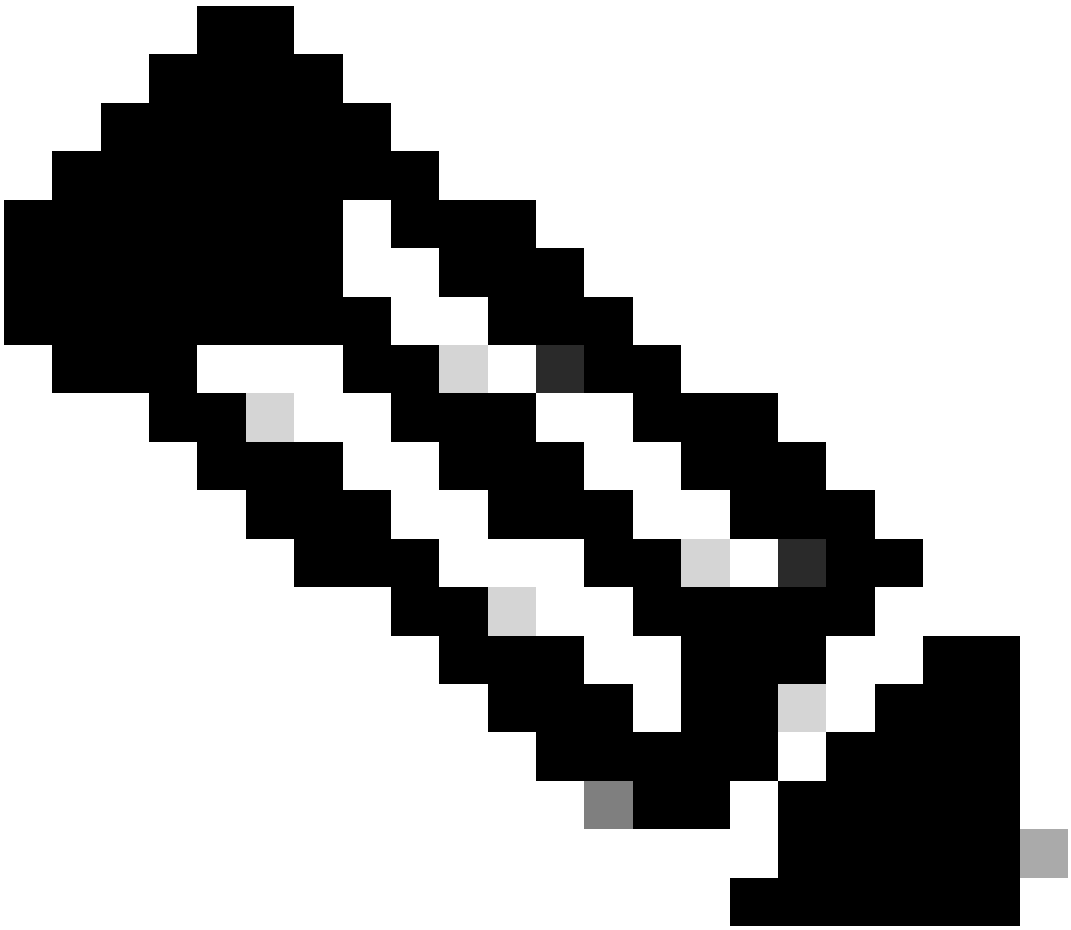
```
<snip>
```

Wenn pYANG installiert ist und funktioniert, fahren Sie mit Modellen Download.

Auf dem nächsten Link werden alle Modelle angezeigt, die auf Cisco IOS XR ausgeführt werden: die Cisco IOS XR-Modelle.

Es wird empfohlen, Clone dieses Modelle im venv-Verzeichnis mit dem nächsten Code-Link: https://github.com/YangModels/yang.git

---



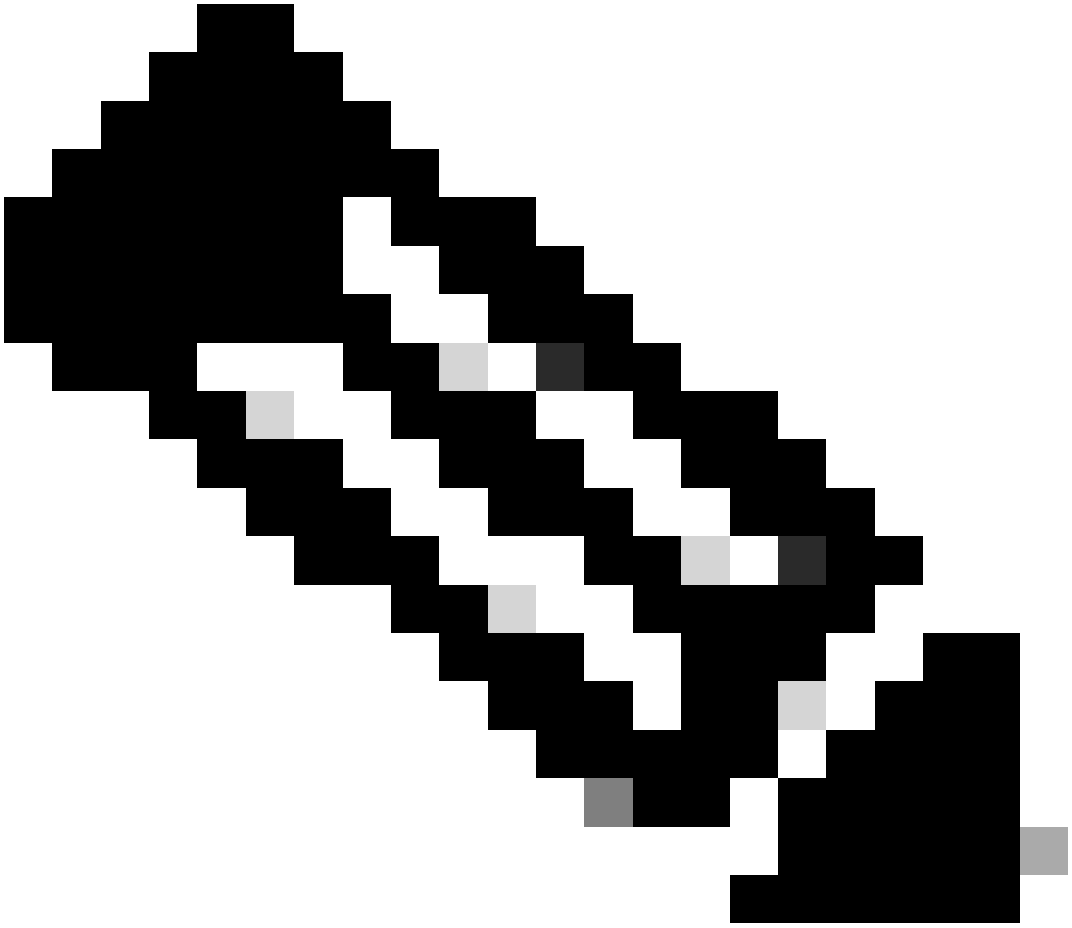
Hinweis: Dies geschieht nicht, wenn die virtuelle Umgebung aktiviert ist.

```
% git clone https://github.com/YangModels/yang.git
Cloning into 'yang'...
remote: Enumerating objects: 54289, done.
remote: Counting objects: 100% (1910/1910), done.
remote: Compressing objects: 100% (323/323), done.
remote: Total 54289 (delta 1643), reused 1684 (delta 1586), pack-reused 52379
```

```
Receiving objects: 100% (54289/54289), 116.64 MiB | 8.98 MiB/s, done.
Resolving deltas: 100% (42908/42908), done.
Updating files: 100% (112197/112197), done.
```

Aktivieren Sie die virtuelle Umgebung erneut, und testen Sie die nächste Abfrage: pyang -f tree yang/vendor/cisco/xr/711/Cisco-IOS-XR-ifmgr-cfg.yang.

```
(virtual_env) % pyang -f tree yang/vendor/cisco/xr/711/Cisco-IOS-XR-ifmgr-cfg.yang
yang/vendor/cisco/xr/711/Cisco-IOS-XR-ifmgr-cfg.yang:5: error: module "Cisco-IOS-XR-types" not found in
yang/vendor/cisco/xr/711/Cisco-IOS-XR-ifmgr-cfg.yang:8: error: module "cisco-semver" not found in searc
module: Cisco-IOS-XR-ifmgr-cfg
  +--rw global-interface-configuration
  |  +--rw link-status?   Link-status-enum
  +--rw interface-configurations
     +--rw interface-configuration* [active interface-name]
        +--rw dampening
        |  +--rw args?                 enumeration
        |  +--rw half-life?            uint32
        |  +--rw reuse-threshold?      uint32
        |  +--rw suppress-threshold?   uint32
        |  +--rw suppress-time?        uint32
        |  +--rw restart-penalty?      uint32
        +--rw mtus
        |  +--rw mtu* [owner]
        |     +--rw owner    xr:Cisco-ios-xr-string
        |     +--rw mtu      uint32
        +--rw encapsulation
        |  +--rw encapsulation?         string
        |  +--rw capsulation-options?   uint32
        +--rw shutdown?                      empty
        +--rw interface-virtual?             empty
        +--rw secondary-admin-state?         Secondary-admin-state-enum
        +--rw interface-mode-non-physical?   Interface-mode-enum
        +--rw bandwidth?                     uint32
        +--rw link-status?                   empty
        +--rw description?                   string
        +--rw active                         Interface-active
        +--rw interface-name                 xr:Interface-name
```



Hinweis: Beachten Sie, dass Leafs ein Datenformat wie String, uint32 usw. haben, während die Roots diese Informationen nicht anzeigen. Operationen wie GET und SET sind dazu bestimmt, diese Werte zu beziehen/zu aktualisieren.

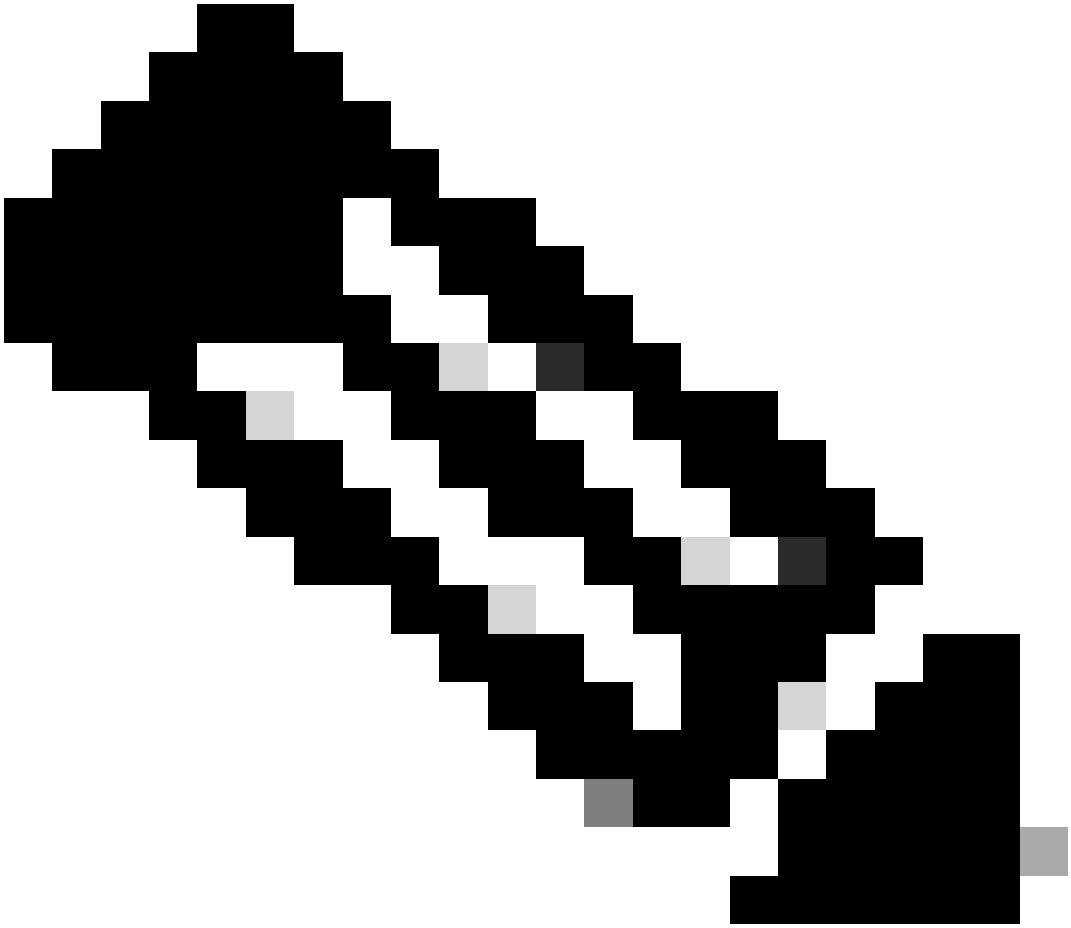
Ein weiterer Hinweis ist, dass die meisten Modelle Erweiterungen erfordern, um die vollständige Konfiguration zu erhalten. In der CLI-Ausgabe gibt es die grundlegende Schnittstellenverwaltungskonfiguration. Falls IPv4 angezeigt werden muss, verwenden Sie den nächsten Schritt:

```
% pyang -f tree yan2/vendor/cisco/xr/711/Cisco-IOS-XR-ifmgr-cfg.yang  yan2/vendor/cisco/xr/711/Cisco-IO
module: Cisco-IOS-XR-ifmgr-cfg
  +--rw global-interface-configuration
  |  +--rw link-status?   Link-status-enum
  +--rw interface-configurations
     +--rw interface-configuration* [active interface-name]
        +--rw dampening
        |  +--rw args?                enumeration
        |  +--rw half-life?           uint32
```

```
|  +--rw reuse-threshold?      uint32
|  +--rw suppress-threshold?   uint32
|  +--rw suppress-time?        uint32
|  +--rw restart-penalty?      uint32
+--rw mtus
|  +--rw mtu* [owner]
|     +--rw owner    xr:Cisco-ios-xr-string
|     +--rw mtu      uint32
+--rw encapsulation
|  +--rw encapsulation?         string
|  +--rw capsulation-options?   uint32
+--rw shutdown?                                empty
+--rw interface-virtual?                       empty
+--rw secondary-admin-state?                   Secondary-admin-state-enum
+--rw interface-mode-non-physical?             Interface-mode-enum
+--rw bandwidth?                               uint32
+--rw link-status?                             empty
+--rw description?                             string
+--rw active                                   Interface-active
+--rw interface-name                           xr:Interface-name
+--rw ipv4-io-cfg:ipv4-network
|  +--rw ipv4-io-cfg:bgp-pa
|  |  +--rw ipv4-io-cfg:input
|  |  |  +--rw ipv4-io-cfg:source-accounting?        boolean
|  |  |  +--rw ipv4-io-cfg:destination-accounting?   boolean
|  |  +--rw ipv4-io-cfg:output
|  |     +--rw ipv4-io-cfg:source-accounting?        boolean
|  |     +--rw ipv4-io-cfg:destination-accounting?   boolean
|  +--rw ipv4-io-cfg:verify
|  |  +--rw ipv4-io-cfg:reachable?      Ipv4-reachable
|  |  +--rw ipv4-io-cfg:self-ping?      Ipv4-self-ping
|  |  +--rw ipv4-io-cfg:default-ping?   Ipv4-default-ping
|  +--rw ipv4-io-cfg:bgp
|  |  +--rw ipv4-io-cfg:qppb
|  |  |  +--rw ipv4-io-cfg:input
|  |  |     +--rw ipv4-io-cfg:source?        Ipv4-interface-qppb
|  |  |     +--rw ipv4-io-cfg:destination?   Ipv4-interface-qppb
|  |  +--rw ipv4-io-cfg:flow-tag
|  |     +--rw ipv4-io-cfg:flow-tag-input
|  |        +--rw ipv4-io-cfg:source?        boolean
|  |        +--rw ipv4-io-cfg:destination?   boolean
|  +--rw ipv4-io-cfg:addresses
|  |  +--rw ipv4-io-cfg:secondaries
|  |  |  +--rw ipv4-io-cfg:secondary* [address]
|  |  |     +--rw ipv4-io-cfg:address      inet:ipv4-address-no-zone
|  |  |     +--rw ipv4-io-cfg:netmask      inet:ipv4-address-no-zone
|  |  |     +--rw ipv4-io-cfg:route-tag?   uint32
|  |  +--rw ipv4-io-cfg:primary!
|  |  |  +--rw ipv4-io-cfg:address      inet:ipv4-address-no-zone
|  |  |  +--rw ipv4-io-cfg:netmask      inet:ipv4-address-no-zone
|  |  |  +--rw ipv4-io-cfg:route-tag?   uint32
|  |  +--rw ipv4-io-cfg:unnumbered?    xr:Interface-name
|  |  +--rw ipv4-io-cfg:dhcp?          empty
|  +--rw ipv4-io-cfg:helper-addresses
|  |  +--rw ipv4-io-cfg:helper-address* [address vrf-name]
|  |     +--rw ipv4-io-cfg:address     inet:ipv4-address-no-zone
|  |     +--rw ipv4-io-cfg:vrf-name    xr:Cisco-ios-xr-string
|  +--rw ipv4-io-cfg:forwarding-enable?       empty
|  +--rw ipv4-io-cfg:icmp-mask-reply?         empty
|  +--rw ipv4-io-cfg:tcp-mss-adjust-enable?   empty
|  +--rw ipv4-io-cfg:ttl-propagate-disable?   empty
|  +--rw ipv4-io-cfg:point-to-point?          empty
```

```
|  +--rw ipv4-io-cfg:mtu?                     uint32
+--rw ipv4-io-cfg:ipv4-network-forwarding
   +--rw ipv4-io-cfg:directed-broadcast?   empty
   +--rw ipv4-io-cfg:unreachables?         empty
   +--rw ipv4-io-cfg:redirects?            empty
```

Bei dieser Abfrage werden zwei Modelle verwendet: Cisco-IOS-XR-ifmgr-cfg.yang und Cisco-IOS-XR-ipv4-io-cfg.yang. Nun wird die IPv4-Adresse als Leaf angezeigt.



> Hinweis: Falls Sie Fehler wie: "yang/vendor/cisco/xr/711/Cisco-IOS-XR-ifmgr-cfg.yang:5: error: module "Cisco-IOS-XR-types" not found in search path" (Cisco-IOS-XR-Typen nicht im Suchpfad gefunden) sehen, fügen Sie den Befehl —path= hinzu.

Mit diesem fertig und geprüft, kann jeder Benutzer Informationen mit den gNMI-Operationen anfordern und das Datum ändern, für weitere Beispiele klicken Sie auf den nächsten Link: [Programmability Configuration Guide]

Falls der Benutzer eine einfache API ausführen möchte, gibt es Tools wie: grpcc.

Diese API wird über NPM installiert. Dies ist das Tool, das im Programmability Configuration Guide-Link verwendet wird. Über diesen Link werden weitere Beispiele für Benutzer zum Testen von Abfragen und Antworten bereitgestellt.

Fehlerbehebung:

Für gNMI ist es erforderlich, die Abfrage zu überprüfen, bevor Sie einen Eintrag sammeln, die meisten APIs wie:

- gnämisch
- grpcc
- gRPC

Zeigen Sie den vom Router generierten Fehler an.

Beispiele:

```
"cisco-grpc:errors": {
"error": [
{
"error-type": "application",
"error-tag": "operation-failed",
"error-severity": "error",
"error-message": "'YANG framework' detected the 'fatal' condition 'Operation failed'"
}
]
}
}
```

ODER

```
"error": [
   {
    "error-type": "application",
    "error-tag": "operation-failed",
    "error-severity": "error",
    "error-path": <path>,
"error-message": "'sysdb' detected the 'warning' condition 'A verifier or EDM callback function returne
   }
  ]
```

Hierbei handelt es sich um plattformabhängige Fehler, die entlang des Routers überprüft werden müssen. Es wird empfohlen, zu überprüfen, ob die Befehle in der Abfrage auch über die CLI im Router ausgegeben werden können.

Bei dieser Art von Fehlern oder anderen Fehlern im Zusammenhang mit der Cisco IOS XR-Plattform können Sie dem TAC folgende Informationen mitteilen:

- Verwendete Abfrage und Vorgang:

```
{
 "Cisco-IOS-XR-ifmgr-cfg:interface-configurations":
  { "interface-configuration": [
    {
      "active": "act",
      "interface-name": "Loopback0",
      "description": "LOCAL TERMINATION ADDRESS",
      "interface-virtual": [
        null
       ],
       "Cisco-IOS-XR-ipv4-io-cfg:ipv4-network": {
        "addresses": {
            "primary": {
              "address": "172.16.255.1",
              "netmask": "255.255.255.255"
       }
      }
     }
    }
   ]
 }
}
```

- Fehler, der angezeigt wird (einer der oben aufgeführten Fehler).
- Geben Sie den nächsten Befehl ein:

```
show grpc trace all
```

Testen Sie die Abfrage einige Male, und wiederholen Sie den Befehl "show grpc trace all".

Die Fehler sind variabel, zeigen aber auch die Komponente an, die ein Problem verursachen kann:

Beispiele:

- "'sysdb' hat die 'warning' Bedingung 'A verifier or EDEDM callback function return: 'not found' erkannt.": Dieser Fehler beschreibt sysdb, wie es erforderlich ist, show tech-Befehle für diesen Prozess im Router zu sammeln.

Das nächste Beispiel zeigt den Prozess show tech for sysdb, der den Fehler zeigt.

```
show tech-support sysdb
```

Für diese Ausgabe, die Fehler-Anzeige eine Komponente und der Fehler, sammeln alle show tech-support, die auf den Fehler in Zusammenhang stehen kann angezeigt werden.

- "'YANG framework' detected the 'fatal' condition 'Operation failed'": Dieser Fehler zeigt keinen Prozess im Router, was bedeutet, dass die Abfrage im Modell fehlschlägt. Geben Sie diese Informationen an das TAC weiter, um zu überprüfen, was fehlschlagen kann.

Nachdem diese Informationen gesammelt wurden, fügen Sie auch den nächsten Befehlssatz hinzu:

In XR VM:

show tech-support tctcpsr

show tech-support grapcc

show tech-support gsp

show tech-support

Informationen zu dieser Übersetzung

Cisco hat dieses Dokument maschinell übersetzen und von einem menschlichen Übersetzer editieren und korrigieren lassen, um unseren Benutzern auf der ganzen Welt Support-Inhalte in ihrer eigenen Sprache zu bieten. Bitte beachten Sie, dass selbst die beste maschinelle Übersetzung nicht so genau ist wie eine von einem professionellen Übersetzer angefertigte. Cisco Systems, Inc. übernimmt keine Haftung für die Richtigkeit dieser Übersetzungen und empfiehlt, immer das englische Originaldokument (siehe bereitgestellter Link) heranzuziehen.