



Pydantify

NetAutoberfest 2025

Urs Baumann 02.10.2025

Urs Baumann



```
>>> qr = QRCode()  
>>> qr.add_data("https://www.linkedin.com/in/ubaumannch")  
>>> qr.print_ascii()
```





How do engineers work with YANG?

```
1 <config>
2   <if:interfaces xmlns:if="urn:ietf:params:xml:ns:yang:ietf-interfaces">
3     {% for interface in vlan_interfaces %}
4       <if:interface>
5         <if:name>Vlanif{{ interface.name }}</if:name>
6         <if:description>uplink</if:description>
7         <if:type xmlns:iana-if-type="urn:ietf:params:xml:ns:yang:iana-if-type">iana-if-type:propVirtual</if:type>
8         <ip:ipv4 xmlns:ip="urn:ietf:params:xml:ns:yang:ietf-ip">
9           <ip:address>
10             <ip:ip>{{ interface.ip.split("/")@[0] }}</ip:ip>
11             <ip:prefix-length>{{ interface.ip.split("/")@[1] }}</ip:prefix-length>
12           </ip:address>
13         </ip:ipv4>
14       </if:interface>
15     {% endfor %}
16   </if:interfaces>
17 </config>
```



What I want I



What I want II

Netconf-XML_Artifact ready

Artifact Definition SRL_NetconfXML-Artifact Device leaf02

text

```
1 [
2 {
3     "srl_nokia-interfaces:interface": [
4         {
5             "srl_nokia-interfaces:name": "Loopback0",
6             "srl_nokia-interfaces:admin-state": "enable",
7             "srl_nokia-interfaces:subinterface": [
8                 {
9                     "srl_nokia-interfaces:index": 0,
10                    "srl_nokia-interfaces:ipv4": [
11                        "srl_nokia-interfaces:address": [
12                            {
13                                "srl_nokia-interfaces:ip-prefix": "10.255.255.1/32"
14                            }
15                        ]
16                    }
17                ]
18            ],
19        },
20        {
21            "srl_nokia-interfaces:name": "ethernet-1/1",
22            "srl_nokia-interfaces:description": "leaf02 to spine01",
23        }
24    ]
25},
26 {
27     "srl_nokia-interfaces:name": "ethernet-1/1",
28     "srl_nokia-interfaces:description": "leaf02 to spine01",
29 }
```

Raw



What I want III

- Python objects with typing, validation and documentation
- Consumable / as intuitive and user-friendly as possible
- IDE autocompletion
- IDE error checks
- IDE type-hints
- Serialization / Deserialization

Eco-System



- pyangbind
- ygot
- YANG Development Kit (ydk-gen)



How it started

- Midterm Project @ Eastern Switzerland University of Applied Sciences
Walther, Dominic and Jovicic, Dejan (2023)
Make Model Driven Network Automation Pythonic
<https://eprints.ost.ch/id/eprint/1089/>



Now

- <https://pydantify.github.io/pydantify/>
- <https://github.com/pydantify/pydantify>
- <https://pypistats.org/packages/pydantify>



- <https://github.com/srl-labs/pydantic-srlinux/>
experimental project
- <https://www.youtube.com/watch?v=CM3sT55zwto>
How can you use Pydantify? – Roman Dodin (Nokia)

Demo



Challanges / not supported

- Pydantify: YANG > Pyyang > **JSONSchema** > DMGC > Pydantic
- Context-aware validation
- must / if / ...
- xPath
- bits / decimal64 representation
- RegEx

Questions?

