



# Pydantify

[OPSAWG]IETF 122

Urs Baumann 25.09.2024



# 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>
```

---



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



# What I want

- 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

# 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=CM3sT55zwt0>  
How can you use Pydantify? — Roman Dodin (Nokia)



Demo



# Challenges / not supported

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

Questions?