

Switch from Biermann-RestConf to RFC8040 interface

Status: 2020-06-27: under review

The OpenDaylight project has implemented the RestConf interface according [RFC8040](#). The previous in OpenDaylight implemented RestConf(-like) interface is supported too and both interfaces can be installed in parallel within the Apache Karaf container. Please see the [documentation of NetConf project of OpenDaylight](#).

This wiki shows difference between both interfaces based on examples.

The syntax is as defined by the [vsCode/rest-client](#) - [sample code](#).

Info regarding [RFC8040](#) and [RFC8527](#)

1. RFC8040 defines RESTCONF
2. The latest open-source implementation of RFC8040 is today in OpenDaylight Aluminium. However OpenDaylight Sodium and OpenDaylight Magnesium are good enough for our purposes.
3. RFC8527 extends RFC8040 to support new datastores as defined in [RFC8525](#), but RFC8527 does not make RFC8040 obsolete.

Description	Biermann	RestConf (RFC8040)
Get OpenDaylights YANG capabilities.	<pre>GET {{baseUrl}}/restconf/modules Authorization: Basic {{user}} {{password}} Accept: application/json</pre>	<pre>GET {{baseUrl}}/rests/data/ietf- yang-library:modules-state Authorization: Basic {{user}} {{password}} Accept: application/yang-data+json</pre>
Get NetConf Topology from configuration datastore	<pre>GET {{baseUrl}}/restconf/config/network-topology:network-topology/topology-netconf Authorization: Basic {{user}} {{password}} Accept: application/json</pre>	<pre>GET {{baseUrl}}/rests/data/network- topology:network-topology/topolog y=topology-netconf?content=config Authorization: Basic {{user}} {{password}} Accept: application/yang-data+json</pre>
Get NetConf Topology from operational datastore	<pre>GET {{baseUrl}}/restconf/operational/network-topology:network-topology/topology/topology-netconf Authorization: Basic {{user}} {{password}} Accept: application/json</pre>	<pre>GET {{baseUrl}}/rests/data/network- topology:network-topology/topolog y=topology-netconf?content=nonconf ig Authorization: Basic {{user}} {{password}} Accept: application/yang-data+json</pre>
Get NetConf Topology from both datastores at the same time	n/a	<pre>GET {{baseUrl}}/rests/data /network-topology:network-topology /topology=topology-netconf Authorization: Basic {{user}} {{password}} Accept: application/yang-data+json</pre>
Get only the node-ids from both datastores	n/a	<pre>GET {{baseUrl}}/rests/data /network-topology:network-topology /topology=topology-netconf? fields=node(node-id) Authorization: Basic {{user}} {{password}} Accept: application/yang-data+json</pre>
Get only the node-ids from operational datastores	n/a	<pre>GET {{baseUrl}}/rests/data /network-topology:network-topology /topology=topology-netconf? content=nonconfig&fields=node(node- id) Authorization: Basic {{user}} {{password}} Accept: application/json</pre>
Get only the node-ids and their connection-status	n/a	<pre>GET {{baseUrl}}/rests/data /network-topology:network-topology /topology=topology-netconf? fields=node(node-id;netconf-node- topology:connection-status) Authorization: Basic {{user}} {{password}} Accept: application/yang-data+json</pre>
Get only the node-ids and their capabilities	n/a	

Get **capabilities** of a given **node-id**

```
GET {{baseUrl}}/restconf
/operational/network-topology:
network-topology/topology
/topology-netconf/node/{{mountPoint}}/netconf-node-topology:available-capabilities
Authorization: Basic {{user}}
{{password}}
Accept: application/json
```

Create new NetworkElement connection with SDN-R **data-provider**.

```
POST {{baseUrl}}/restconf/operations/data-provider:create-network-element-connection
Authorization: Basic {{user}}
{{password}}
Accept: application/json
Content-Type: application/json
```

```
{
  "data-provider:input": {
    "id": "new-mountpoint-name",
    "node-id": "new-mountpoint-name",
    "host": "10.10.10.10",
    "port": "830",
    "username": "netconf",
    "password": "netconf",
    "is-required": "true"
  }
}
```

Note: the http body is the same, but pay attention to the namespace of the input and output object.

Read SDN-R Connection List

```
POST {{baseUrl}}/restconf/operations/data-provider:read-network-element-connection-list
Authorization: Basic {{user}}
{{password}}
Accept: application/json
Content-Type: application/json
```

```
{
  "data-provider:input": {
    "filter": [{
      "property": "status",
      "filtervalue": "Connecting"
    }],
    "sortorder": [{
      "property": "node-id",
      "sortorder": "descending"
    }],
    "pagination": {
      "size": 5,
      "page": 3
    }
  }
}
```

Some complex filtering

n/a

```
GET {{baseUrl}}/rests/data
/network-topology:network-topology
/topology=topology-netconf?
content=nonconfig&fields=node
(node-id;netconf-node-topology:
available-capabilities/available-capability/capability)
Authorization: Basic {{user}}
{{password}}
Accept: application/yang-data+json
```

```
GET {{baseUrl}}/rests/data
/network-topology:network-topology
/topology=topology-netconf/node={{mountPoint}}/netconf-node-topology:available-capabilities?
fields=available-capability(capability)
Authorization: Basic {{user}}
{{password}}
Accept: application/yang-data+json
```

```
POST {{baseUrl}}/rests/operations/data-provider:create-network-element-connection
Authorization: Basic {{user}}
{{password}}
Accept: application/yang-data+json
Content-Type: application/yang-data+json
```

```
{
  "data-provider:input": {
    "id": "new-mountpoint-name",
    "node-id": "new-mountpoint-name",
    "host": "10.10.10.10",
    "port": "830",
    "username": "netconf",
    "password": "netconf",
    "is-required": "true"
  }
}
```

Note: the http body is the same! Just the URL is different.

```
POST {{baseUrl}}/rests/operations/data-provider:read-network-element-connection-list
Authorization: Basic {{user}}
{{password}}
Accept: application/yang-data+json
Content-Type: application/yang-data+json
```

```
{
  "data-provider:input": {
    "filter": [{
      "property": "status",
      "filtervalue": "Connecting"
    }],
    "sortorder": [{
      "property": "node-id",
      "sortorder": "descending"
    }],
    "pagination": {
      "size": 5,
      "page": 3
    }
  }
}
```

Edit Config operation:merge

```
@node=Core14-ONF-NTS-Manager

PUT {{baseUrl}}/restconf/config/network-topology:network-topology/topology/topology-netconf/node/{{node}}/yang-ext:mount/network-topology-simulator:simulator-config/notification-config
Authorization: Basic {{user}}
{{password}}
Accept: application/json
Content-Type: application/json

{
  "network-topology-simulator:notification-config": {
    "network-topology-simulator:is-netconf-available": true,
    "network-topology-simulator:ves-heartbeat-period": 0,
    "network-topology-simulator:is-ves-available": false,
    "network-topology-simulator:fault-notification-delay-period": [
      60, 50, 40, 10
    ]
  }
}

# Pay attention to the namespaces!!!
```

Create MountPoint

```
@node=my-new-mount-point-2
PUT {{baseUrl}}/restconf/config/network-topology:network-topology/topology/topology-netconf/node/{{node}}
Authorization: Basic {{user}}
{{password}}
Accept: application/json
Content-Type: application/json
```

```
GET {{baseUrl}}/rests/data/network-topology:network-topology/topology=topology-netconf/node={{mountPoint}}/yang-ext:mount/core-model-1-4:control-construct/logical-termination-point=e6c5831d-37ce-4ef4-b075-733f6791bde1/layer-protocol=pRppR4R/wire-interface-2-0:wire-interface-pac/wire-interface-configuration?fields=interface-name;problem-kind-severity-list(problem-kind-name)
Authorization: Basic {{user}}
{{password}}
Accept: application/yang-data+json
```

```
@node=Core14-ONF-NTS-Manager

PUT {{baseUrl}}/rests/data/network-topology:network-topology/topology=topology-netconf/node={{node}}/yang-ext:mount/network-topology-simulator:simulator-config/notification-config
Authorization: Basic {{user}}
{{password}}
Accept: application/yang-data+json
Content-Type: application/yang-data+json

{
  "network-topology-simulator:notification-config": {
    "is-netconf-available": true,
    "ves-heartbeat-period": 0,
    "is-ves-available": false,
    "fault-notification-delay-period": [
      60, 50, 40, 10
    ]
  }
}

# Pay attention to the namespaces!!!
```

```
@node=my-new-mount-point-2
PUT {{baseUrl}}/rests/data/network-topology:network-topology/topology=topology-netconf/node={{node}}
Authorization: Basic {{user}}
{{password}}
Accept: application/yang-data+json
Content-Type: application/yang-data+json
```

```

{
  "network-topology:node": [
    {
      "node-id": "{node}",
      "netconf-node-topology:host"
: "127.0.0.1",
      "netconf-node-topology:port"
: 830,
      "netconf-node-topology:
password": "netconf",
      "netconf-node-topology:
username": "netconf",
      "netconf-node-topology:
sleep-factor": 1.5,
      "netconf-node-topology:tcp-
only": false,
      "netconf-node-topology:
reconnect-on-changed-schema": true,

      "netconf-node-topology:
default-request-timeout-millis": 6
0000,
      "netconf-node-topology:
connection-timeout-millis": 20000,
      "netconf-node-topology:max-
connection-attempts": 100,
      "netconf-node-topology:
between-attempts-timeout-millis":
2000,
      "netconf-node-topology:
keepalive-delay": 120,
      "netconf-node-topology:
concurrent-rpc-limit": 0,
      "netconf-node-topology:
actor-response-wait-time": 5
    }
  ]
}

```

Delete MountPoint

```

@node=my-new-mount-point-2
DELETE {{baseUrl}}/restconf/config
/network-topology:network-topology
/topology/topology-netconf/node/{{
node}}
Authorization: Basic {{user}}
{{password}}
Accept: application/json

```

```

{
  "network-topology:node": [
    {
      "node-id": "{node}",
      "host": "127.0.0.1",
      "port": 830,
      "password": "netconf",
      "username": "netconf",
      "sleep-factor": 1.5,
      "tcp-only": false,
      "reconnect-on-changed-
schema": true,
      "default-request-timeout-
millis": 60000,
      "connection-timeout-millis":
20000,
      "max-connection-attempts": 1
00,
      "between-attempts-timeout-
millis": 2000,
      "keepalive-delay": 120,
      "concurrent-rpc-limit": 0,
      "actor-response-wait-time": 5
    }
  ]
}

```

```

@node=my-new-mount-point-2
DELETE {{baseUrl}}/rests/data
/network-topology:network-topology
/topology=topology-netconf/node={{
node}}
Authorization: Basic {{user}}
{{password}}
Accept: application/yang-data+json

```