odl-telemetry Documentation

Release stable/neon

Jul 11, 2019

Contents

This documentation provides critical information needed to help you write code for the Telemetry project.

CHAPTER 1

Developer Guides

1.1 Telemetry Developer Guide

1.1.1 Telemetry Architecture

- **Configurator**
  - Responsible for providing configuration info to device, it use the openconfig models.
- **Collector**
  - Implements a gRPC server to receive data from device and save them to thirdparty database. Also, the data will be sent to thirdparty application according to the notification mechanism.

1.1.2 APIs in Telemetry

The sections below give details about the configuration settings for the components that can be configured.

Configurator

API Description

- telemetry/configurator/api/src/main/yang/telemetry-configurator-api.yang
  - **add-telemetry-sensor**
    - Add telemetry sensor(s) to data store, the sensor contains the target data you want it reports.
  - **query-telemetry-sensor**
    - Acquire telemetry sensor(s) configured from data store.
  - **delete-telemetry-sensor**
    - Delete telemetry sensor(s) in the data store and device.

  - **add-telemetry-destination**
    - Add telemetry destination(s) to data store, the destination stands for where the device sends data to.
  - **query-telemetry-destination**
    - Acquire telemetry destination(s) configured from data store.
  - **delete-telemetry-destination**
    - Delete telemetry destination(s) in the data store and device.
  - **configure-node-telemetry-subscription**
    - Configure subscription(s) to device, the subscription contains sensors and destinations as mentioned above, also other parameters such as frequency, transfer protocol etc.
  - **query-node-telemetry-subscription**
    - Acquire telemetry subscription(s) configured in device.
  - **delete-node-telemetry-subscription**
    - Delete telemetry subscription(s) in the data store and device.
  - **delete-node-telemetry-subscription-sensor**
    - Delete sensor(s) under subscription(s) yang construction in the data store and device.
  - **delete-node-telemetry-subscription-destination**
    - Delete destination(s) under subscription(s) yang construction in the data store and device.

Collector

API Description

- telemetry/collector/datastorage/src/main/yang/telemetry-datastorage.yang
  - **data-store**
    - Save data which received from device to thirdparty database.

1.1.3 Sample Configurations

1. Add Telemetry Sensor

REST API : *POST /restconf/operations/telemetry-configurator-api:add-telemetry-sensor*

Sample JSON Data

```
{
     "input": {
                "telemetry-sensor-group":[{
                                "telemetry-sensor-group-id":"sensor1",
                                "telemetry-sensor-paths":[{
                                        "telemetry-sensor-path":"path1",
                                        "sensor-exclude-filter":"filter1"
                                },
                                {       "telemetry-sensor-path":"path2",
                                        "sensor-exclude-filter":"filter2"
```

(continues on next page)

(continued from previous page)

```
                                    }]
                            },
                            {       "telemetry-sensor-group-id":"sensor2",
                                    "telemetry-sensor-paths":[{
                                            "telemetry-sensor-path":"path3",
                                            "sensor-exclude-filter":"filter3"
                                    },
                                    {       "telemetry-sensor-path":"path4",
                                            "sensor-exclude-filter":"filter4"
                                    }]
                            }]
    }
}
```

2. Query Telemetry Sensor

REST API : *POST /restconf/operations/telemetry-configurator-api:query-telemetry-sensor*

Sample JSON Data

```
{
    "input": {

    }
}
```

3. Delete Telemetry Sensor

REST API : *POST /restconf/operations/telemetry-configurator-api:delete-telemetry-sensor*

Sample JSON Data

```
{
    "input": {
         "telemetry-sensor-group":[{
                                    "sensor-group-id":"sensor1"
                                },
                                {
                                  "sensor-group-id":"sensor2"
                                }]
     }
}
```

4. Add Telemetry Destination

REST API : *POST /restconf/operations/telemetry-configurator-api:add-telemetry-destination*

Sample JSON Data

```
{
    "input": {
        "telemetry-destination-group":[{
                                        "destination-group-id":"destination1",
```

(continues on next page)

(continued from previous page)

```
                                        "destination-profile":[{
                                                                "destination-address
↪":"10.42.89.218",
                                                                "destination-port":
↪"50051"
                                                            }]
                                    },
                                    {
                                        "destination-group-id":"destination2",
                                        "destination-profile":[{
                                                                "destination-address
↪":"10.42.89.15",
                                                                "destination-port":
↪"50051"
                                                            }]
                                    }]
    }
}
```

5. Query Telemetry Destination

REST API : *POST /restconf/operations/telemetry-configurator-api:query-telemetry-destination*

Sample JSON Data

```
{
    "input": {

    }
}
```

6. Delete Telemetry Destination

REST API : *POST /restconf/operations/telemetry-configurator-api:delete-telemetry-destination*

Sample JSON Data

```
{
    "input": {
        "telemetry-destination-group":[{
                                        "destination-group-id":"destination1"
                                    },
                                    {
                                      "destination-group-id":"destination2"
                                    }]
    }
}
```

7. Configure subscription

REST API : *POST /restconf/operations/telemetry-configurator-api:configure-node-telemetry-subscription*

Sample JSON Data

```
{
    "input": {
        "telemetry-node":[{
                            "node-id":"node1",
                            "telemetry-subscription":[{
                                                        "subscription-name":
↪"subscription1",
                                                        "protocol-type":"STREAM_GRPC",
                                                        "encoding-type":"ENC_PROTO3",
                                                        "local-source-address":"127.0.
↪0.1",
                                                        "originated-qos-marking":"5",
                                                        "telemetry-sensor":[{
                                                                             "sensor-
↪group-id":"sensor1",
                                                                             "sample-
↪interval":"200",
↪"heartbeat-interval":"60",
↪"suppress-redundant":"false"
                                                                            },
                                                                            {
                                                                             "sensor-
↪group-id":"sensor2",
                                                                             "sample-
↪interval":"100",
↪"heartbeat-interval":"60",
↪"suppress-redundant":"false"
                                                                            }],
                                                        "telemetry-destination":[{
↪"destination-group-id":"destination1"
                                                                                 },
                                                                                 {
↪"destination-group-id":"destination2"
                                                                                 }]
                                                      }]
                          }]
    }
}
```

8. Query subscription

REST API : *POST /restconf/operations/telemetry-configurator-api:query-node-telemetry-subscription*

Sample JSON Data

```
{
    "input": {

    }
}
```

9. Delete subscription

REST API : *POST /restconf/operations/telemetry-configurator-api:delete-node-telemetry-subscription*

Sample JSON Data

```
{
    "input": {
        "telemetry-node":[{
                            "node-id":"node1",
                            "telemetry-node-subscription":[{
                                  "subscription-name":"subscription1"
                            }]
                        },
                        {
                            "node-id":"node2",
                            "telemetry-node-subscription":[{
                                  "subscription-name":"subscription1"
                            }]
                        }]
    }
}
```

10. Delete subscription sensor

REST API : *POST /restconf/operations/telemetry-configurator-api:delete-node-telemetry-subscription-sensor*

Sample JSON Data

```
{
    "input": {
        "telemetry-node":[{
                            "node-id":"node1",
                            "telemetry-node-subscription":[{
                                  "subscription-name":"subscription1",
                                  "telemetry-node-subscription-sensor":[{
                                      "sensor-group-id":"sensor1"
                                  },
                                  {
                                      "sensor-group-id":"sensor2"
                                  }]
                            }]
                        },
                        {
                            "node-id":"node2",
                            "telemetry-node-subscription":[{
                                  "subscription-name":"subscription1",
                                  "telemetry-node-subscription-sensor":[{
                                      "sensor-group-id":"sensor1"
                                  },
                                  {
                                      "sensor-group-id":"sensor2"
                                  }]
                            }]
                        }]
    }
}
```

11. Delete subscription destination

REST API : *POST /restconf/operations/telemetry-configurator-api:delete-node-telemetry-subscription-destination*

Sample JSON Data

```
{
    "input": {
        "telemetry-node":[{
                            "node-id":"node1",
                            "telemetry-node-subscription":[{
                                  "subscription-name":"subscription1",
                                  "telemetry-node-subscription-destination":[{
                                      "destination-group-id":"destination1"
                                  },
                                  {
                                      "destination-group-id":"destination2"
                                  }]
                            }]
                        },
                        {
                            "node-id":"node2",
                            "telemetry-node-subscription":[{
                                  "subscription-name":"subscription1",
                                  "telemetry-node-subscription-destination":[{
                                      "destination-group-id":"destination1"
                                  },
                                  {
                                      "destination-group-id":"destination2"
                                  }]
                            }]
                        }]
    }
}
```

12. Data storage

REST API : *POST /restconf/operations/telemetry-datastorage:data-store*

Sample JSON Data

```
{
    "input": {
        "node-id":"node1",
        "telemetry-data":[{
                            "timestamp":"20181214165033",
                            "base-path":"interfaces/interface",
                                                    "sample-interval":"30",
                                                    "key-value":[{
                                                         "key":"interface1",
                                                             "value":""
                                                    },
                                                    {
                                                         "key":"interface2",
                                                             "value":""
                                                    }]
                        },
```

(continues on next page)

(continued from previous page)

```
                            {
                                "timestamp":"20181214165034",
                                "base-path":"interfaces/interface",
                                                                "sample-interval":"30",
                                                                "key-value":[{
                                                                     "key":"interface3",
                                                                         "value":""
                                                                },
                                                                {
                                                                     "key":"interface4",
                                                                         "value":""
                                                                }]
                            }]
    }
}
```

CHAPTER 2

User Guides

2.1 Telemetry User Guide

2.1.1 Overview

Traditional way to get running information from network device, such as snmp or netflow, is becoming insufficient in sdn network. Because sdn application needs better performance and more flexible data format. It‘s necessary to implement a telemetry channel to support new app in sdn((e.g. traffic optimization).

2.1.2 Telemetry User-Facing Features

- **odl-telemetry-all**
  - This feature contains all other features/bundles of Telemetry project. If you install it, it provides all functions that the Telemetry project can support.
- **odl-telemetry-collect**
  - This feature provides a function which implements a gRPC server to receive the measurement data of device and show them to user.
- **odl-telemetry-configurator**
  - This feature mainly provides function about telemetry model data configuration.

2.1.3 How To Start

Preparing for Installation

1. Forwarding devices must support NETCONF, so that OpenDaylight can connect to them and config resoures via NETCONF.
2. Forwarding devices must support gRpc, so that OpenDaylight can complete packet in/out procedure, etc.

Installation Feature

Run OpenDaylight and install Telemetry Service *odl-telemetry-all* as shown below:

```
feature:install odl-telemetry-all
```

For a more detailed overview of the Telemetry, see the *Telemetry Developer Guide*.