Relate Data Test Task v4

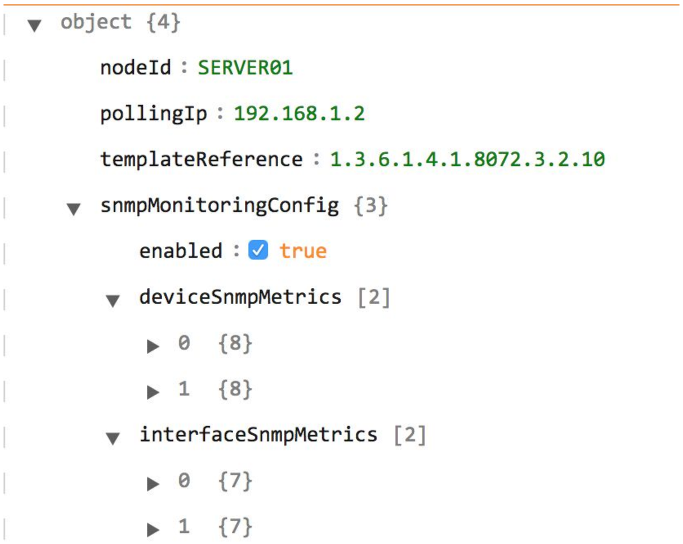
This test task should be completed using the following libraries in React.

AG GRID:
https://www.ag-grid.com/best-react-data-grid/
https://www.ag-grid.com/example.php#/

RC-COLLAPSE
https://react-component.github.io/collapse/examples/simple.html

No backend requirement as data can be read from JSON file.

JSON Structure Explained



```
▼ object {4}
     nodeId : SERVER01
     pollingIp : 192.168.1.2
     templateReference : 1.3.6.1.4.1.8072.3.2.10
   ▼ snmpMonitoringConfig {3}
        enabled : ☑ true
      ▼ deviceSnmpMetrics [2]
         ▶ 0  {8}
         ▶ 1  {8}
      ▼ interfaceSnmpMetrics [2]
         ▶ 0  {7}
         ▶ 1  {7}
```

Image: 1

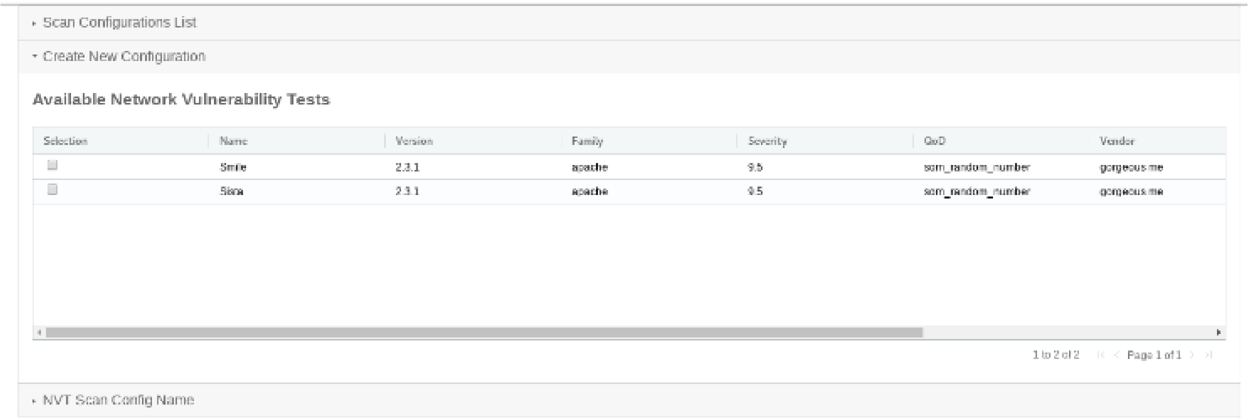
You can **discard** NodeId, pollingID and templateReference information.

Table Header should be called “ SNMP Monitoring Configuration”

snmpMonitoringConfig should be represented as a **checkbox** with label “Enabled”.

deviceSnmpMetrics and **interfaceSnmpMetrics** should be represented as collapsable items. For reference see image 2. For this component, you should use the library **RC-COLLAPSE** library to create collapsable menus or items.

“Instead of Scan Configuration List and Create New Configuration you should have Device Metrics & InterfaceMetrics.”



Available Network Vulnerability Tests

Selection	Name	Version	Family	Severity	QoD	Vendor
□	Smile	2.3.1	apache	9.5	som_random_number	gorgeous.me
□	Sara	2.3.1	apache	9.5	som_random_number	gorgeous.me

1 to 2 of 2 |< < Page 1 of 1 > >|

Image: 2

Each Collapsable item opens an AG GRID Table as per the image 3 which represents each item from device and interface snmp metrics of the JSON.

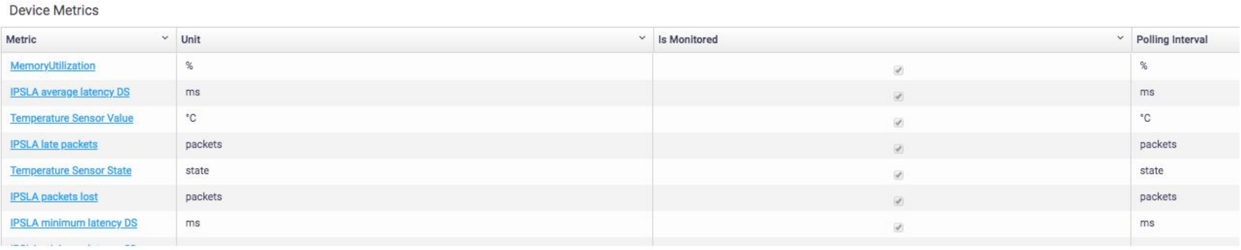
*The data for the table content comes from **deviceSnmpMetrics** and **interfaceSnmpMetrics** in the JSON file.*

```
"enabled": true,
   "deviceSnmpMetrics": [
    {
     "metricName": "net-snmp.HOST-RESOURCES-MIB.cpuLoad",
      "metricDisplayName": "CPU Load",
      "isMonitored": true,
      "metricGroup": "CPU",
      "monitoredIndices": {
        "196608": {
          "isMonitored": true
        }
      },
"pollingInterval": 300,
```

Metric	***Unit***	***Monitored***	***Polling Interval***
Memory Utilization	*%*	☑	*300*

Tables should be sortable using default functions of AG Grid Library. One table for Device Metrics and one table for Interface Metrics each in its collapsable item.

An illustrative example of how this table may look is as in Image 3.



Device Metrics

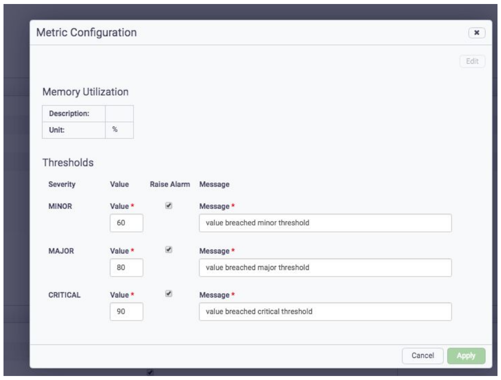
Metric	Unit	Is Monitored	Polling Interval
MemoryUtilization	%	✓	%
IPSLA average latency DS	ms	✓	ms
Temperature Sensor Value	°C	✓	°C
IPSLA late packets	packets	✓	packets
Temperature Sensor State	state	✓	state
IPSLA packets lost	packets	✓	packets
IPSLA minimum latency DS	ms	✓	ms

Image : 3

Each Device or Interface Metric has properties in JSON called “thresholdConfig” that contains information that should be displayed in “MODAL window” (Image 4) when the user clicks on any device or interface Metric.

This information comes from the following in the JSON

```
"thresholdConfig": [
        {
          "severity": "MINOR",
          "value": 60,
          "message": "value breached minor threshold",
          "raiseAlarm": true
        },
        {
          "severity": "MAJOR",
          "value": 80,
          "message": "value breached major threshold",
          "raiseAlarm": true
        },
        {
          "severity": "CRITICAL",
          "value": 90,
          "message": "value breached critical threshold",
          "raiseAlarm": true
        }
      ],
      "unit": "%"
```



Metric Configuration

Memory Utilization

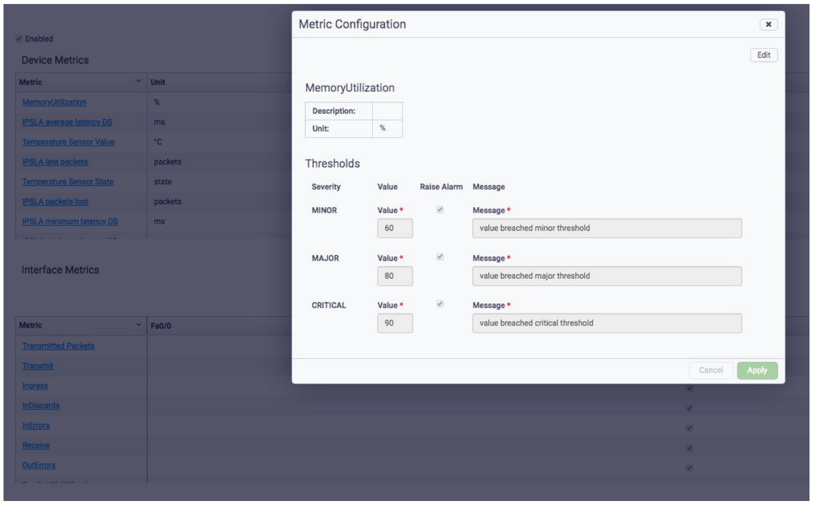
Description:	
Unit:	%

Thresholds

Severity	Value	Raise Alarm	Message
MINOR	Value * 60	☑	Message * value breached minor threshold
MAJOR	Value * 80	☑	Message * value breached major threshold
CRITICAL	Value * 90	☑	Message * value breached critical threshold

Image: 4

Image 5 shows a representation of the desired solution where Device Metrics and Interface Metrics are collapsable items and contain AG Grid tables.



☑ Enabled

Device Metrics

Metric	Unit
MemoryUtilization	%
IPSLA average latency DS	ms
Temperature Sensor Value	°C
IPSLA late packets	packets
Temperature Sensor State	state
IPSLA packets lost	packets
IPSLA minimum latency DS	ms

Interface Metrics

Metric	Fa0/0
Transmitted Packets	
Transmit	
Ingress	
InDiscards	
InErrors	
Receive	
OutErrors	

Metric Configuration

MemoryUtilization

Description:	
Unit:	%

Thresholds

Severity	Value	Raise Alarm	Message
MINOR	Value * 60	☑	Message * value breached minor threshold
MAJOR	Value * 80	☑	Message * value breached major threshold
CRITICAL	Value * 90	☑	Message * value breached critical threshold

Image: 5

Expected Results

You can deploy the application as a react app on either Heroku or Github and send a link of the code as well as the deployed application.

Please do not send just code as we will not deploy and check it.

Time limit for this task is maximum 2 days.