



Cisco NIR DCNM REST API Examples

This chapter contains the following sections:

- [all_resources\(\)](#), on page 1
- [anomalies_details\(\)](#), on page 2
- [anomalies_summary\(\)](#), on page 3
- [flows_details\(\)](#), on page 3
- [flows_summary\(\)](#), on page 5
- [flows_top_flows\(\)](#), on page 6
- [flows_top_nodes\(\)](#), on page 8
- [get_fabrics_anomaly_summary\(\)](#), on page 9
- [get_fabrics_list\(\)](#), on page 9
- [get_nodes_list\(\)](#), on page 10
- [get_protocols_details\(\)](#), on page 11
- [get_protocols_resources\(\)](#), on page 12
- [get_protocols_topentities\(\)](#), on page 13
- [get_protocols_topnodes\(\)](#), on page 14
- [health_diagnostics\(\)](#), on page 15
- [service_health\(\)](#), on page 15
- [utilization_node_details\(\)](#), on page 16
- [utilization_top_nodes\(\)](#), on page 17

all_resources()

```
Get all resources .
REST URL      :
               GET /api/telemetry/utilization/resources.json
Parameters    :
               None
Example       :
               curl -k -i -XGET
               'https://<ip:port>/appcenter/Cisco/NIR/apiserver-api/api/telemetry/utilization/resources.json'
Response      :
               {
                 "totalResultsCount": 5,
                 "totalItemsCount":5,
                 "entries": [
                   {
                     "categoryName": "",
```

```

        "resourceName": "EndPoints",
    }
    <-- SNIP LIST OF ALL OTHER RESOURCES -->
    {
    }
    ]
}

```

anomalies_details()

Get the anomalies in the system

REST URL :

```
GET /api/telemetry/anomalies/details.json
```

Parameters :

```

startTs (optional) => Start timestamp, default:now-1h
endTs   (optional) => End timestamp, default:current-time
count   (optional) => Num.of nodes in response, default:10
orderBy (optional) => Sort per the given field

```

Example :

```
curl -ksb -XGET
```

```
'https://<ip:port>/appcenter/Cisco/NIR/apiserver-api/api/telemetry/anomalies/details.json'
```

Response :

```

{
  "totalItemsCount": 90,
  "totalResultsCount": 90,
  "offset": 0,
  "entries": [
    {
      "anomalyId": "QUE000000000018",
      "category": "System Resource",
      "startTs": "2018-09-19T16:45:05.679Z",
      "endTs": "2018-09-19T16:58:05.778Z",
      "entityName": "svc_ifc_policyelem",
      "severity": "critical",
      "anomalyType": "build-up",
      "nodeNames": [
        "leaf2"
      ],
      "resourceType": "queue",
      "resourceName": "recvQ",
      "anomalyStr": "[svc_ifc_policyelem] : Unexpected build-up of 7487 message[s]
in recvQ",
      "anomalyScore": 83
    },
    {
      "anomalyId": "QUE000000000007",
      "category": "System Resource",
      "startTs": "2018-09-19T15:16:10.420Z",
      "endTs": "2018-09-19T16:49:01.289Z",
      "entityName": "svc_ifc_policyelem",
      "severity": "critical",
      "anomalyType": "build-up",
      "nodeNames": [
        "leaf1"
      ],
      "resourceType": "queue",
      "resourceName": "recvQ",
      "anomalyStr": "[svc_ifc_policyelem] : Unexpected build-up of 7502 message[s]
in recvQ",
      "anomalyScore": 83
    }
  ]
}

```

```
    ]
  }
```

anomalies_summary()

```
Get summary of the anomalies in the system
REST URL :
  GET /api/telemetry/anomalies/summary.json
Parameters :
  startTs (optional) => Start timestamp, default:now-1h
  endTs (optional) => End timestamp, default:current-time
Example :
  curl -ksb -XGET
'https://<ip:port>/appcenter/Cisco/NIR/apiserver-api/api/telemetry/anomalies/summary.json'
Response :
  {
    "totalAnomalyCount": 2,
    "totalAnomalyScore": 120.0,
    "entries": [
      {
        "severity": "warning",
        "anomalyCount": 1,
        "anomalyScore": 40.0
      },
      {
        "severity": "major",
        "anomalyCount": 1,
        "anomalyScore": 80.0
      }
    ]
  }
```

flows_details()

```
Get detailed flows
REST URL :
  GET /api/telemetry/flows/details.json
Parameters :
  startTs (mandatory) => Start timestamp,
  endTs (mandatory) => End timestamp, default:current-time
  filter (optional) => Lucene format filter
{srcIp,srcPort,dstIp,dstPort,ProtocolName,ingressVrf,egressVrf}, default:null
  statName (optional) => Stat name {flow:latency, flow:epmove, flow:pktdrop,
flow:ingressburstmax, flow:egressburstmax, flow:ingressPktCount, flow:egressPktCount}
  granularity (optional) => Granularity of time period
  fabricName (optional) => limit the records pertaining to this fabricName
Example:
  curl -k -i -XGET
'https://<ip:port>/appcenter/Cisco/NIR/apiserver-api/api/telemetry/flows/details.json'
Response:
  {
    "nodeName": null,
    "description": "",
    "statName": null,
    "entries": [
      {
        "flowId": "44.3.3.26:0",
        "srcIp": "44.3.3.26",
```

```

"dstIp": "42.2.2.22",
"srcPort": "0",
"dstPort": "0",
"protocol": "61",
"protocolName": "ANY-HOST",
"ingressVrf": "ctx4_1",
"egressVrf": "ctx4_1",
"flowType": "IPv4",
"ingressTenant": "tele4",
"egressTenant": "tele4",
"stats": [
  {
    "ingressPktCount": 6875,
    "ingressByteCount": 8250000,
    "egressPktCount": 0,
    "egressByteCount": 0,
    "ingressBurst": 0,
    "ingressBurstMax": 4800,
    "egressBurst": 0,
    "egressBurstMax": 0,
    "hashCollision": 0,
    "latency": 0,
    "srcMoveCount": 0,
    "dstMoveCount": 0,
    "moveCount": 0,
    "dropPktCount": 0,
    "dropNodes": [
      "telemetry-hw-spine1"
    ],
    "paths": [
      [
        {
          "node": "telemetry-hw-leaf3",
          "nodeType": "Leaf",
          "ingressVifs": [
            "eth1/1"
          ],
          "egressVifs": [
            "eth1/49"
          ]
        },
        {
          "node": "telemetry-hw-spine1",
          "nodeType": "Spine",
          "asicDropCode": 128,
          "dropReason": "",
          "dropType": "info",
          "ingressVifs": [
            "eth2/2"
          ],
          "egressVifs": [
            ""
          ]
        }
      ]
    ],
    "nodeName": [
      "telemetry-hw-leaf3",
      "telemetry-hw-spine1"
    ],
    "ingressNodes": [
      "telemetry-hw-leaf3"
    ],
    "egressNodes": [],

```

```

        "anomalyScore": 1,
        "dropReasons": [],
        "srcEpg": "testl3out",
        "dstEpg": "",
        "ts": "2019-02-01T19:18:56.458Z",
        "originTs": "2019-02-01T19:18:38.445Z",
        "terminalTs": "2019-02-01T19:20:42.419Z"
    }
],
"srcEpg": "testl3out",
"dstEpg": ""
}
]
}

```

flows_summary()

Browse flows.

REST URL :
GET /api/telemetry/flows/summary.json

Parameters :
startTs (optional) => Start timestamp, default:now-1h
endTs (optional) => End timestamp, default:current-time
filter => Lucene format filter, default:null
fabricName (optional) => limit the records pertaining to this fabricName

Example:

```
curl -k -i -XGET
```

```
'https://<ip:port>/appcenter/Cisco/NIR/apiserver-api/api/telemetry/flows/summary.json'
```

Response:

```

{
  "nodeName": null,
  "description": "",
  "statName": null,
  "entries": [
    {
      "flowId": "44.3.3.26:0",
      "srcIp": "44.3.3.26",
      "dstIp": "42.2.2.22",
      "srcPort": "0",
      "dstPort": "0",
      "protocol": "61",
      "protocolName": "ANY-HOST",
      "ingressVrf": "ctx4_1",
      "egressVrf": "ctx4_1",
      "flowType": "IPV4",
      "ingressTenant": "tele4",
      "egressTenant": "tele4",
      "stats": [
        {
          "ingressPktCount": 6875,
          "ingressByteCount": 8250000,
          "egressPktCount": 0,
          "egressByteCount": 0,
          "ingressBurst": 0,
          "ingressBurstMax": 4800,
          "egressBurst": 0,
          "egressBurstMax": 0,
          "hashCollision": 0,
          "latency": 0,
          "srcMoveCount": 0,
          "dstMoveCount": 0,
          "moveCount": 0,

```

flows_top_flows()

```

        "dropPktCount": 0,
        "dropNodes": [
            "telemetry-hw-spine1"
        ],
        "paths": [
            [
                {
                    "node": "telemetry-hw-leaf3",
                    "nodeType": "Leaf",
                    "ingressVifs": [
                        "eth1/1"
                    ],
                    "egressVifs": [
                        "eth1/49"
                    ]
                },
                {
                    "node": "telemetry-hw-spine1",
                    "nodeType": "Spine",
                    "asicDropCode": 128,
                    "dropReason": "",
                    "dropType": "info",
                    "ingressVifs": [
                        "eth2/2"
                    ],
                    "egressVifs": [
                        ""
                    ]
                }
            ]
        ],
        "nodeNameNames": [
            "telemetry-hw-leaf3",
            "telemetry-hw-spine1"
        ],
        "ingressNodes": [
            "telemetry-hw-leaf3"
        ],
        "egressNodes": [],
        "anomalyScore": 1,
        "dropReasons": [],
        "srcEpg": "test13out",
        "dstEpg": "",
        "ts": "2019-02-01T19:18:56.458Z",
        "originTs": "2019-02-01T19:18:38.445Z",
        "terminalTs": "2019-02-01T19:20:42.419Z"
    },
    "srcEpg": "test13out",
    "dstEpg": ""
}
]
}

```

flows_top_flows()

Get flows top flows.

REST URL :

GET /api/telemetry/flows/topFlows.json

Parameters :

startTs (optional) => Start timestamp, default:now-1h

endTs (optional) => End timestamp, default:current-time

granularity (optional) => Granularity of time period
 statName (optional) => Stat name {flow:latency, flow:epmove, flow:pktdrop}
 fabricName (optional) => limit the records pertaining to this fabricName

Example:

```
curl -k -i -XGET
```

```
'https://<ip:port>/appcenter/Cisco/NIR/apiserver-api/api/telemetry/flows/topFlows.json'
```

Response:

```
{
  "nodeName": null,
  "description": "",
  "statName": null,
  "entries": [
    {
      "flowId": "44.3.3.26:0",
      "srcIp": "44.3.3.26",
      "dstIp": "42.2.2.22",
      "srcPort": "0",
      "dstPort": "0",
      "protocol": "61",
      "protocolName": "ANY-HOST",
      "ingressVrf": "ctx4_1",
      "egressVrf": "ctx4_1",
      "flowType": "IPV4",
      "ingressTenant": "tele4",
      "egressTenant": "tele4",
      "stats": [
        {
          "ingressPktCount": 6875,
          "ingressByteCount": 8250000,
          "egressPktCount": 0,
          "egressByteCount": 0,
          "ingressBurst": 0,
          "ingressBurstMax": 4800,
          "egressBurst": 0,
          "egressBurstMax": 0,
          "hashCollision": 0,
          "latency": 0,
          "srcMoveCount": 0,
          "dstMoveCount": 0,
          "moveCount": 0,
          "dropPktCount": 0,
          "dropNodes": [
            "telemetry-hw-spine1"
          ],
          "paths": [
            [
              {
                "node": "telemetry-hw-leaf3",
                "nodeType": "Leaf",
                "ingressVifs": [
                  "eth1/1"
                ],
                "egressVifs": [
                  "eth1/49"
                ]
              },
              {
                "node": "telemetry-hw-spine1",
                "nodeType": "Spine",
                "asicDropCode": 128,
                "dropReason": "",
                "dropType": "info",
                "ingressVifs": [
                  "eth2/2"
                ]
              }
            ]
          ]
        }
      ]
    }
  ]
}
```

```

        ],
        "egressVifs": [
            ""
        ]
    }
]
],
"nodeNames": [
    "telemetry-hw-leaf3",
    "telemetry-hw-spine1"
],
"ingressNodes": [
    "telemetry-hw-leaf3"
],
"egressNodes": [],
"anomalyScore": 1,
"dropReasons": [],
"srcEpg": "test13out",
"dstEpg": "",
"ts": "2019-02-01T19:18:56.458Z",
"originTs": "2019-02-01T19:18:38.445Z",
"terminalTs": "2019-02-01T19:20:42.419Z"
}
],
"srcEpg": "test13out",
"dstEpg": ""
}
]
}

```

flows_top_nodes()

Get flows top nodes.

```

REST URL :
    GET /api/telemetry/flows/topNodes.json
Parameters :
    startTs (optional) => Start timestamp, default:now-1h
    endTs (optional) => End timestamp, default:current-time
    granularity (optional) => Granularity of time period
    statName (optional) => Stat name {flow:latency, flow:epmove, flow:pktdrop},
    default:flow-latency
    fabricName (optional) => limit the records pertaining to this fabricName
Example:
    curl -k -i -XGET
'https://<ip:port>/appcenter/Cisco/NIR/apiserver-api/api/telemetry/flows/topNodes.json'
Response:

```

```

{
    "entries": [
        {
            "nodeName": "telemetry-hw-spine1",
            "description": "",
            "stats": [
                {
                    "ts": "2019-02-01T19:16:32.002Z",
                    "latency": 6
                }
            ]
        },
        {
            "nodeName": "telemetry-hv-leaf1",
            "description": "",
            "stats": [

```



```

        {
            "ts": "2019-02-01T19:16:32.002Z",
            "latency": 5
        }
    ]
}

```

get_fabrics_anomaly_summary()

```

Get fabric anomaly summary.
REST URL   :
            GET /api/telemetry/fabricsSummary.json
Parameters :
    fabricName (mandatory) => Name of the Fabric
    startTs           => Start timestamp, default:current-time - 1 hour
    endTs             => End timestamp, default:current-time
    include="anomalyScore" => Requires the Latest Maximum anomaly scores of the fabric,
    default:'no'
    history           => Requires the timeseries data of sum(anomaly scores, default:'no'
                        data, default=5m
                        granularity => applicable if history = "yes" , granulairy of the timeseries
Example     :
    curl -k -i -XGET
'https://<ip:port>/appcenter/Cisco/NIR/apiserver-api/api/telemetry/fabricsSummary.json'
Response    :
    {
        "anomalyScore" : "X"
        "entries": [
            {
                totalAnomalyScore ; X
                ts : now
            }
            .....
            {
                totalAnomalyScore ; X
                ts : now
            }
        ],
        "totalResultsCount": N,
        "totalItemsCount": N
    }

```

get_fabrics_list()

```

Get fabrics list.
REST URL   :
            GET /api/telemetry/fabrics.json
Parameters :
    filter           => Lucene format filter, default:null
Example     :
    curl -k -i -XGET
'https://<ip:port>/appcenter/Cisco/NIR/apiserver-api/api/telemetry/fabrics.json'
Response    :
    {
        "entries": [
            {

```

```

        "fabricName": "FABRIC1",
        "fabricId": "1",
        "vendor": "CISCO_N9K_STANDALONE",
        "fabricType": "VXLAN",
        "configStatus": "ENABLED",
        "switchCount": 2,
        "controllerCount": 0
    },
    {
        "fabricName": "FABRIC2",
        "fabricId": "2",
        "vendor": "CISCO_ACI",
        "fabricType": "VXLAN",
        "configStatus": "ENABLED",
        "switchCount": 4,
        "controllerCount": 3
    },
    <--snip-->
],
"totalResultsCount": 11,
"totalItemsCount": 11
}

```

get_nodes_list()

```

Get nodes list.
REST URL   :
    GET /api/telemetry/nodes.json
Parameters :
    startTs (mandatory) => Start timestamp
    endTs       => End timestamp, default:current-time
    count       => Num.of nodes in response, default:1000
    filter      => Lucene format filter, default:null
Example    :
    curl -k -i -XGET
'https://<ip:port>/appcenter/Cisco/NIR/apiserver-api/api/telemetry/nodes.json'
Response   :
    {
        "entries": [
            {
                "nodeRole": "leaf",
                "nodeId": "302",
                "nodeName": "rleaf-scrimshaw2",
                "nodeMgmtIp": "1.2.3.4"
            },
            {
                "nodeRole": "spine",
                "nodeId": "205",
                "nodeName": "swmp14-dopplebock",
                "nodeMgmtIp": "1.2.3.4"
            },
            <--snip-->
        ],
        "totalResultsCount": 11,
        "offset": 0,
        "totalItemsCount": 11
    }

```

get_protocols_details()

```

Get Telemetry Protocol Stats details.
REST URL   :
    GET /api/telemetry/protocols/details.json
Parameters :
    startTs (mandatory) => Start timestamp
    endTs    => End timestamp, default:current-time
    fabricName => limit the records pertaining to this fabricName
    nodeName => Name of node
    statName  => <protocol[:counter[:qualifier]], protocol[:counter[:qualifier]]...>

    history  => '1' or '0', default is '0', indicates time-series request
    granularity => Granularity of time period, default:5m
    orderBy  => One statName of the format <protocol[:counter[:qualifier]]>
    filter   => Lucene format filter to query for specific nodeName or sourceName,
default:null
Example    :
    curl -k -i -XGET
'https://<ip:port>/appcenter/Cisco/NIR/apiserver-api/api/telemetry/protocols/details.json'
Response   :
    {
        "totalResultsCount": 6,
        "totalItemsCount": 6,
        "offset": 0,
        "description": "Protocol statistical counters",
        "entries": [
            {
                "nodeName": "leaf-103",
                "entries": [
                    {
                        "sourceName": "phys-[eth1/14]",
                        "entries": [
                            {
                                "counterName": "InterfaceUtilisationIngress",
                                "value": 60.625,
                                "trending": "up",
                                "stats": [
                                    {
                                        "ts": "2018-10-24T05:05:00.000Z",
                                        "value": 60.625
                                    },
                                    {
                                        "ts": "2018-10-24T05:00:00.000Z",
                                        "value": 59.827586206896555
                                    },
                                    {
                                        "ts": "2018-10-24T04:55:00.000Z",
                                        "value": 59.57142857142857
                                    }
                                ]
                            }
                        ]
                    }
                ]
            },
            {
                "sourceName": "phys-[eth1/11]",
                "entries": [
                    {
                        "counterName": "LldpPktsEgress",
                        "value": 111.0,
                        "trending": "up",

```

```

        "stats": [
          {
            "ts": "2018-10-24T05:05:00.000Z",
            "value": 111.0
          },
          {
            "ts": "2018-10-24T05:00:00.000Z",
            "value": 110.10344827586206
          },
          {
            "ts": "2018-10-24T04:55:00.000Z",
            "value": 109.61904761904762
          }
        ]
      }
    ]
  }
}

```

get_protocols_resources()

Get Telemetry Protocol Stats resources.

REST URL :

```
GET /api/telemetry/protocols/resources.json
```

Parameters :

filter => Lucene format filter, default:null

fabricName => limit the records pertaining to this fabricName

Example :

```
curl -k -i -XGET
```

'https://<ip:port>/appcenter/Cisco/NIR/apiserver-api/api/telemetry/protocols/resources.json'

Response :

```

[
  {
    "protocol": "interface",
    "counter": "utilisation",
    "qualifiers": [
      "ingress",
      "egress"
    ]
  },
  {
    "protocol": "interface",
    "counter": "bytes",
    "qualifiers": [
      "ingress",
      "egress"
    ]
  },
  <--snip-->
  {
    "protocol": "lldp",
    "counter": "pkts",
    "qualifiers": [
      "ingress",
      "egress"
    ]
  },
  {
    "protocol": "lldp",

```

```

        "counter": "errors"
    }
]

```

get_protocols_topentities()

```

Get Telemetry Protocol Stats topEntities.
REST URL   :
            GET /api/telemetry/protocols/topEntities.json
Parameters :
    startTs (mandatory) => Start timestamp
    endTs    => End timestamp, default:current-time
    fabricName => limit the records pertaining to this fabricName
    statName  => parameter to find topEntities protocol[:counter[:qualifier]]
    granularity => Granularity of time period, default:5m
    filter    => Lucene format filter to query for specific nodeName or sourceName,
    default:null
Example    :
            curl -k -i -XGET
            'https://<ip:port>/appcenter/Cisco/NIR/apiserver-api/api/telemetry/protocols/topEntities.json'
Response   :
    {
        "totalResultsCount": 6,
        "totalItemsCount": 6,
        "offset": 0,
        "description": "Protocol statistical counters",
        "entries": [
            {
                "nodeName": "leaf-103",
                "entries": [
                    {
                        "sourceName": "phys-[eth1/4]",
                        "entries": [
                            {
                                "counterName": "InterfaceUtilisationIngress",
                                "value": 65.53333333333333,
                                "trending": "down",
                                "stats": [
                                    {
                                        "ts": "2018-10-24T05:20:00.000Z",
                                        "value": 65.53333333333333
                                    },
                                    {
                                        "ts": "2018-10-24T05:15:00.000Z",
                                        "value": 65.78571428571429
                                    }
                                ]
                            }
                        ]
                    }
                ]
            },
            {
                "sourceName": "phys-[eth1/14]",
                "entries": [
                    {
                        "counterName": "InterfaceUtilisationIngress",
                        "value": 59.666666666666664,
                        "trending": "up",
                        "stats": [
                            {
                                "ts": "2018-10-24T05:20:00.000Z",
                                "value": 59.666666666666664
                            }
                        ],
                    }
                ]
            }
        ]
    }

```

```

        {
            "ts": "2018-10-24T05:15:00.000Z",
            "value": 59.5
        }
    ]
}

```

<---snip-->

get_protocols_topnodes()

Get Telemetry Protocol Stats topNodes.

REST URL :

GET /api/telemetry/protocols/topNodes.json

Parameters :

startTs (mandatory) => Start timestamp
endTs => End timestamp, default:current-time
fabricName => limit the records pertaining to this fabricName
nodeName => Name of node
statName => interface:utilization
summarize => '1' or '0', default is '0', summarizes across protocols

Example :

curl -k -i -XGET

'https://<ip:port>/appcenter/Cisco/NIR/apiserver-api/api/telemetry/protocols/topNodes.json'

Response :

```

{
    "totalResultsCount": 6,
    "totalItemsCount": 6,
    "offset": 0,
    "description": "Protocol top nodes by score",
    "entries": [
        {
            "nodeName": "leaf-103",
            "entries": [
                {
                    "counterName": "protocol|utilization",
                    "stats": [
                        {
                            "ts": "2019-02-08T13:50:00.000Z",
                            "value": 62.33333333333336
                        },
                        {
                            "ts": "2019-02-08T13:45:00.000Z",
                            "value": 62.83333333333336
                        }
                    ]
                },
                {
                    "value": 62.33333333333336,
                    "trending": "down"
                }
            ]
        },
        ....
    ]
}

```

health_diagnostics()

```

Get health dianostics.
REST URL   :
            GET /api/telemetry/health/collectionStats.json
Parameters :
            None
Example    :
            curl -k -i -XGET
'https://<ip:port>/appcenter/Cisco/NIR/apiserver-api/api/telemetry/health/collectionStats.json'
Response   :
            {
              "totalItemsCount": 11,
              "entries": [
                {
                  "nodeName": "pod20-leaf3",
                  "stats": [
                    {
                      "resource": "sysStats",
                      "totalItemsCount": 9600,
                      "lastUpdatedTs": "2018-06-13T10:25:52.468Z",
                      "state": "HEALTHY"
                    }
                  ]
                },
                <---snip-->
              ]
            }

```

service_health()

```

Get the health of the services
REST URL   :
            GET /api/telemetry/health/serviceHealth.json
Parameters :
            None
Example    :
            curl -k -i -XGET
'https://<ip:port>/appcenter/Cisco/NIR/apiserver-api/api/telemetry/health/serviceHealth.json'
Response   :
            {
              "entries": [
                {
                  "serviceType": "THIRD_PARTY_SERVICE",
                  "serviceName": "elastic",
                  "state": "HEALTHY",
                  "displayName": "Data Store"
                },
                {
                  "serviceType": "CISCO_SERVICE",
                  "serviceName": "correlator",
                  "state": "HEALTHY",
                  "displayName": "Correlator"
                },
                <---snip-->
              ]
            }

```

utilization_node_details()

```

Get node details .
REST URL      :
    GET /api/telemetry/utilization/nodeDetails.json
Parameters   :
    None
Example      :
    curl -k -i -XGET
'https://<ip:port>/appcenter/Cisco/NIR/apiserver-api/api/telemetry/utilizationnodeDetails.json'
Response     :
    {
        "totalResultsCount": 157,
        "totalItemsCount":157,
        "entries": [
            {
                "nodeName": "node-1",
                "entries": [
                    {
                        "resourceName":"cpu",
                        "latestValue":"85",
                        "maxValue":"100",
                        "resourceCategory":"",
                        "trending":"down",
                        "values":[
                            { "value":"85", "ts":"2018-02-21T20:21:03.109Z" },
                            {},
                            <--snip-->
                            {}
                        ]
                    },
                    {
                        "resourceName":"memory",
                        "latestValue":"84",
                        "maxValue":"100",
                        "resourceCategory":"",
                        "trending":"up",
                        "values":[
                            { "value":"84", "ts":"2018-02-21T20:21:03.109Z" },
                            {},
                            <--snip-->
                            {}
                        ]
                    },
                    <-- snip , LIST OF ALL OTHER RESOURCES -->
                    {
                        "resourceName":"ports",
                        "latestValue":"83",
                        "maxValue":"100",
                        "resourceCategory":"",
                        "trending":"up",
                        "values":[
                            { "value":"83", "ts":"2018-02-21T20:21:03.109Z" },
                            {},
                            <--snip-->
                            {}
                        ]
                    }
                ]
            },
            {
                "nodeName": "node-2"
            }
        ]
    }

```



```

    <-- same as in node-1 -->
  }
  <----snip LIST OF ALL OTHER NODES ---->
  {
    "nodeName": "node-10"
    <-- same as in node-1 -->
  }
]
}

```

utilization_top_nodes()

Get top nodes by utilization .

REST URL :

```
GET /api/telemetry/utilization/topNodes.json
```

Parameters :

None

Example :

```
curl -k -i -XGET
```

'https://<ip:port>/appcenter/Cisco/NIR/apiserver-api/api/telemetry/utilization/topNodes.json'

Response :

```

{
  "totalResultsCount": 10,
  "totalItemsCount":10,
  "entries": [
    {
      "nodeName": "node-1",
      "entries": [
        {
          "resourceName":"cpu",
          "latestValue":"85",
          "maxValue":"100",
          "resourceCategory":"",
          "trending":"down",
          "values":[
            { "value":"85", "ts":"2018-02-21T20:21:03.109Z" },
            {},
            <--snip-->
            {}
          ]
        },
        {
          "resourceName":"memory",
          "latestValue":"84",
          "maxValue":"100",
          "resourceCategory":"",
          "trending":"up",
          "values":[
            { "value":"84", "ts":"2018-02-21T20:21:03.109Z" },
            {},
            <--snip-->
            {}
          ]
        },
        {
          "resourceName":"ports",
          "latestValue":"83",
          "maxValue":"100",
          "resourceCategory":"",
          "trending":"up",
          "values":[
            { "value":"83", "ts":"2018-02-21T20:21:03.109Z" },

```

```
        {},
        <---snip-->
        {}
    ]
}
},
{
  "nodeName": "node-2"
  <-- same as in node-1 -->
}
<----snip---->
{
  "nodeName": "node-10"
  <-- same as in node-1 -->
}
]
}
```