

2-iloRestExamples

September 7, 2021

1 iLOrest examples

Version 0.71

1.1 Introduction

This Jupyter Notebook contains iLOrest examples as well as tips and tricks answering recurrent questions from HPE customers.

1.2 Environment preparation

The following cell sets environment variables and checks the connectivity toward the various BMCs used in this notebook.

```
[1]: ##### Environment preparation (Version: 0.30) #####
```

```
# Set Student ID number
export stdid=601
Id=$(id --user --name)

# location and ports variables
ObmcBasePort=44000
iLO5DlBasePort=45000
iLO5SyBasePort=46000
RMCsdfBasePort=47000

let OpenBmcPort=${ObmcBasePort}+${stdid}
let iLO5DlSimulatorPort=${iLO5DlBasePort}+${stdid}
let iLO5SySimulatorPort=${iLO5SyBasePort}+${stdid}
let RMCsdfSimulatorPort=${RMCsdfBasePort}+${stdid}
let ilo5Port=443

CacheDlDir="${PWD}/iLOrestCacheDl"
CacheSyDir="${PWD}/iLOrestCacheSy"
CacheSdfDir="${PWD}/iLOrestCacheSdf"
LogDir="${PWD}"
LogFile="${LogDir}/iLOrest.log"
```

```

iLO5DlSimulatorIP=ilo5simulators
iLO5SySimulatorIP=ilo5simulators
RMCSdfSimulatorIP=ilo5simulators
OpenBmcIP=openbmcsimulators

iLO5DlSimulator=${iLO5SimulatorIP}:${iLO5DlSimulatorPort}
iLO5DlSimulatorURI=https://${iLO5DlSimulator}

iLO5SySimulator=${iLO5SimulatorIP}:${iLO5SySimulatorPort}
iLO5SySimulatorURI=https://${iLO5SySimulator}

RMCSdfSimulator=${RMCSdfSimulatorIP}:${RMCSdfSimulatorPort}
RMCSdfSimulatorURI=https://${RMCSdfSimulator}

OpenBmc="${OpenBmcIP}:${OpenBmcPort}"
OpenBmcURI="https://${OpenBmc}"

ilo5IP="ilo5"
ilo5="${ilo5IP}:${ilo5Port}"
ilo5URI="https://${ilo5}"

# Credentials
User="student"
Password=P@ssw0rd!

# Miscellaneous
w=$(basename $PWD)
alias ResetSimulators="../create-globalbmc.shc.x &>/dev/null ; sleep 1"

# Verify we can reach the remote Bmcs on the right HTTPS ports.
for bmc in OpenBmc ilo5 iLO5DlSimulator iLO5SySimulator RMCSdfSimulator ; do
    ip="${bmc}IP" ; port="${bmc}Port"
    nc -vz $(eval echo "\${ip}") $(eval echo "\${port}") &> /dev/null &&
    echo "$bmc is reachable" \
    || echo "WARNING: Problem reaching $bmc"
done

echo
# Retrieve iLO firmware versions from ServiceRoot (no credentials needed)
for bmc in ilo5 iLO5DlSimulator iLO5SySimulator ; do
    ip="${bmc}IP" ; port="${bmc}Port"
    echo -n "$bmc firmware version: "
    curl --silent --insecure -X GET https://$(eval echo "\${ip}"):$(eval echo \
→\${port})/redfish/v1 | \
    jq '.Oem.Hpe.Manager[] | .[] | .ManagerFirmwareVersion' 2>/dev/null
done

```

```
echo
# Print ilorest version
ilorest --version
```

```
OpenBmc is reachable
ilo5 is reachable
iLO5DlSimulator is reachable
iLO5SySimulator is reachable
RMCsdfSimulator is reachable
```

```
ilo5 firmware version: "2.44"
iLO5DlSimulator firmware version: "2.44"
iLO5SySimulator firmware version: "2.47"
```

RESTful Interface Tool 3.2.2

1.2.1 Cache preparation

The following examples can be run against either an HPE ProLiant DL360 Gen10 simulator or an HPE Sy480 Gen10 simulators. To allow these simultaneous tests, we need to populate a cache directory for each simulator with the corresponding mockup and then create aliases of the `ilorest` command with the ad'hoc cache directory supplied on the command line.

Mockup files have been created with the [DMTF Mockup creator utility](#).

```
[2]: # iLOrest cache preparation
iLO5SyMockup="iLO5Sy480g10.tgz"
iLO5DlMockup="iLO5Dl360g10.tgz"
RMCsdfMockup="RMCsdf.tgz"

CacheLocation="${PWD}"
echo
for s in iLO5Dl iLO5Sy RMCsdf ; do
    case $s in
        RMC*) ServerType=$(echo $s | cut -dC -f2); ;;
        iLO*) ServerType=$(echo $s | cut -d5 -f2); ;;
        *) echo "Server $s not known" ; break; ;;
    esac

    BasePort="${s}BasePort" ; SimulIP="${s}SimulatorIP";
    →SimulPort="${s}SimulatorPort"
    exp="s?$(eval echo \$$BasePort)?$(eval echo \$$SimulPort)?"

    CacheDir="${CacheLocation}/$(eval echo \$$SimulIP):$(eval echo \$$SimulPort)"
    Mockup=$(eval echo "\${s}Mockup")

    [ -d ${CacheDir} ] && rm -rf ${CacheDir} &>/dev/null
```

```

mkdir -p ${CacheDir}/cache/                                &>/dev/null
tar -C ${CacheDir}/cache -x -f ${Mockup}
alias ilorest${ServerType}="ilorest --nologo --cache-dir ${CacheDir} --logdir_
→${LogDir}"
echo -e "\tPopulated $s cache directory under $(basename ${CacheDir}) and_
→created ilorest${ServerType} alias"
done

```

Populated iLO5Dl cache directory under ilo5simulators:45601 and created ilorestDl alias

Populated iLO5Sy cache directory under ilo5simulators:46601 and created ilorestSy alias

Populated RMCSdf cache directory under ilo5simulators:47601 and created ilorestSdf alias

1.2.2 Restart iLO 5 simulator

If you need or desire to restart your iLO 5 simulator to start this workshop from scratch or for other reasons, run the following cell at any time.

```

[3]: # Simulators restart
ResetSimulators

# Verify we can reach the remote Bmcs on the right HTTPS ports.
for bmc in ilo5 OpenBmc iLO5DlSimulator iLO5SySimulator RMCSdfSimulator ; do
    ip="${bmc}IP" ; port="${bmc}Port"
    nc -vz $(eval echo "\${ip}") $(eval echo "\${port}") &>/dev/null &&
    echo "$bmc is reachable" \
    || echo "WARNING: Problem reaching $bmc"
done

```

ilo5 is reachable

OpenBmc is reachable

iLO5DlSimulator is reachable

iLO5SySimulator is reachable

RMCSdfSimulator is reachable

1.3 List of examples

Note that the list of examples present in this Jupyter Notebook is changing over time. The classification is not following any strict model; it is a mix of Root service and data type locations.

Account Service - iLO users: Enforce password complexity and minimum password length

Chassis - Chassis Serial Number - Power State - Indicator LED - Chassis Asset Tag

Thermal - Server thermal information - Thermal configuration

Managers (iLO) - Retrieve iLO firmware version - Enable/Disable SNMPv1 in iLO

CompuSystem - PostState and DiscoveryComplete - AMS and HostOS - Bios/RBSU Login password - Computer System Asset Tag

Resource Directory - Introduction

Security - Security Dashboard - iLO Security Mode/State - Login Security Banner

Bios - Server Serial Number

Storage - SSD Wear Status

Logout - Logout

1.4 Account Service

This root service models the iLO User Accounts.

1.4.1 iLO users: Enforce password complexity and minimum password length

[API Reference document](#) keyword: EnforcePasswordComplexity

iLO GUI location: Security - Access Settings

```
[4]: # Select type of server: Dl (ProLiant DL360g10) or Sy (Synergy)
s="Dl"

# Select DMTF data type
eval ilorest$s select AccountService

# List properties of the HPE Oem extension in json format
eval ilorest$s list Oem/Hpe --json

# Modify two AccountService parameters
eval ilorest$s set Oem/Hpe/EnforcePasswordComplexity=True
eval ilorest$s set Oem/Hpe/MinPasswordLength=5

# Retrieve the list of modified parameters
echo -e "\n\nList of modified parameters:"
eval ilorest$s status

# Uncomment the following command to commit changes in the iLO 5 simulator.
# Note that you will not get the same answer as if done on a real iLO 5.
#eval ilorest$s commit

# Undo changes (clear cache)
eval ilorest$s select AccountService --refresh
```

```
{
  "Oem": {
```

```

    "Hpe": {
      "@odata.context":
"/redfish/v1/$metadata#HpeILOAccountService.HpeILOAccountService",
      "@odata.id": "/redfish/v1/AccountService",
      "@odata.type": "#HpeILOAccountService.v2_3_0.HpeILOAccountService",
      "Actions": {
        "#HpeILOAccountService.ImportKerberosKeytab": {
          "target": "/redfish/v1/AccountService/Actions/Oem/Hpe/HpeILOAccountService.ImportKerberosKeytab"
        }
      },
      "AuthFailureDelayTimeSeconds": 10,
      "AuthFailureLoggingThreshold": 3,
      "AuthFailuresBeforeDelay": 1,
      "DefaultPassword": null,
      "DefaultUserName": null,
      "DirectorySettings": {
        "LdapAuthenticationMode": "Disabled",
        "LdapCaCertificateLoaded": false,
        "LdapCaCertificates": {
          "@odata.id":
"/redfish/v1/AccountService/ExternalAccountProviders/LDAP/Certificates"
        },
        "LdapServerPort": 636
      },
      "DirectoryTest": {
        "@odata.id": "/redfish/v1/AccountService/DirectoryTest"
      },
      "EnforcePasswordComplexity": false,
      "Id": "AccountService",
      "KerberosSettings": {
        "KDCServerPort": 88,
        "KerberosRealm": ""
      },
      "MinPasswordLength": 8
    }
  }
}

```

Added the following patch:

```

{
  "Oem/Hpe/EnforcePasswordComplexity": true
}

```

Added the following patch:

```

{
  "Oem/Hpe/MinPasswordLength": 5
}

```

List of modified parameters:

Current changes found:

AccountService.v1_5_0(/redfish/v1/AccountService/) (Currently selected)

Oem/Hpe/EnforcePasswordComplexity=True

Oem/Hpe/MinPasswordLength=5

<- Environment Preparation ; List of examples ; Restart iLO 5 Simulator ; Logout ->

1.5 Chassis

This section contains parameters related to the Chassis [data type](#). A Chassis represents a physical or virtual container of compute resources with attributes such as Field Replacement Units (FRU) information, power supplies, temperature, etc.

Complex chassis subsystems may have their own data type (i.e. Thermal)

Chassis examples: - Rack - Blade enclosure - HPE Synergy frame - Compute node chassis

1.5.1 Chassis Serial Number

The chassis Serial Number (SN) of an HPE compute node mentioned on an external label/tag with code bar/QRcode is identical to its embedded server serial number (Bios parameter). Note that the serial number of a C7000 enclosure chassis or Synergy frame chassis is unique and different to every SN of embedded compute nodes.

The following cell retrieves the serial number of an HPE compute node chassis and shows explicitly that it is a read-only property. The server serial number is presented here in the documentation.

This chassis serial number is also present in the Bios data type as a writable resource. See the example below.

[API Reference document](#) keyword: SerialNumber

iLO GUI location: Information - iLO Overview

```
[5]: # Select type of server: Dl (ProLiant DL360g10) or Sy (Synergy)
s="Dl"

# Select data type
eval ilorest$s select Chassis

# Get SerialNumber from the Chassis data type
eval ilorest$s get SerialNumber

# Note that this property is READ-ONLY
echo -e "\n\nGet information about resource:"
eval ilorest$s info SerialNumber
```

SerialNumber=CZJ93402YV

Get information about resource:

NAME

SerialNumber

DESCRIPTION

The chassis serial number.

TYPE

string

null

READ-ONLY

True

1.5.2 Power State

The following cell retrieves the power state and information related to this parameter. Note that this parameter is read-only. If you want a more precise server state, read the PostState and DiscoveryComplete and AMS and HostOS paragraphs.

As mentioned in the [API reference documentation](#) this property is part of the Chassis data type.

```
[6]: # Select type of server: Dl (ProLiant DL360g10) or Sy (Synergy)
s="Dl"

# Select data type
eval ilorest$s select Chassis

# Get property value
eval ilorest$s get PowerState

# Retrieve property info with possible values.
echo -e "\n\nGet information about resource:"
eval ilorest$s info PowerState
```

PowerState=On

Get information about resource:

NAME

PowerState

DESCRIPTION

This is the current power state of the chassis.

TYPE

string
null

READ-ONLY

True

POSSIBLE VALUES

None
On
Off
Unknown
Reset

1.5.3 Indicator LED

The chassis indicator LED corresponds to the blue LED that is used to identify the chassis of a compute node or of a Synergy frame in the datacenter.

The following cell prints the status of the IndicatorLED property of the compute node only (RackMount or Blade). It filters out the IndicatorLED status of the Synergy frame if any.

```
[7]: # Select type of server: Dl (ProLiant DL360g10) or Sy (Synergy)
#s="Dl"
s="Dl"

# Set filter depending on Chassis type
case $s in Dl ) Filter="RackMount"; ;; Sy ) Filter="Blade"; ;; * ) echo_
→"Chassis Type not recognized"; ;; esac

# Select data type
eval ilorest$s select Chassis

# Get property value
IndicatorLED=$(eval ilorest$s get --json IndicatorLED --filter_
→ChassisType="$Filter")
echo "IndicatorLED=$IndicatorLED"

# Retrieve property info
eval ilorest$s info IndicatorLED
```

```
IndicatorLED={  
    "IndicatorLED": "Off"  
}
```

NAME
IndicatorLED

DESCRIPTION
The chassis indicator LED that is used to identify the chassis.
The user can manipulate this LED.

TYPE
string
null

READ-ONLY
False

POSSIBLE VALUES
None
Unknown
Lit
Blinking
Off

```
[8]: # Select type of server: Dl (ProLiant DL360g10) or Sy (Synergy)  
s="Dl"  
  
# Set filter depending on Chassis type  
case $s in Dl ) Filter="RackMount"; ; Sy ) Filter="Blade"; ; * ) echo_  
    ↪ "Chassis Type not recognized"; ; esac  
  
# Refresh cache  
eval ilorest$s select Chassis --refresh  
  
# Change the indicator LED state  
LEDState=$(eval ilorest$s get IndicatorLED --filter ChassisType="$Filter" | cut_  
    ↪ -d= --field=2)  
echo -e "LED State:$LEDState\n"  
  
if [ "$LEDState" = "Off" ]; then  
    LEDState="Lit"
```

```

else
    LEDState="Off"
fi

# Change property and commit
echo "Changing IndicatorLED to $LEDState"
eval ilorest$s set IndicatorLED="$LEDState" --filter ChassisType="$Filter" ␣
→#--commit
echo

# Validate property change
#eval ilorest$s get IndicatorLED

```

LED State:Off

Changing IndicatorLED to Lit

Added the following patch:

```

{
  "IndicatorLED": "Lit"
}

```

1.5.4 Chassis Asset Tag

The Chassis AssetTag is different from the ComputerSystem Asset Tag.

```

[9]: # Select type of server: Dl (ProLiant DL360g10) or Sy (Synergy)
s="Dl"

# Select data type
eval ilorest$s select Chassis --refresh

# Get property value
eval ilorest$s get AssetTag

```

AssetTag=""

```

[10]: # Select type of server: Dl (ProLiant DL360g10) or Sy (Synergy)
s="Dl"

# Select data type
eval ilorest$s select Chassis --refresh

# Change the Chassis Asset Tag property. Uncomment the --commit when performing␣
→this action against a real BMC
eval ilorest$s set AssetTag="New\ Chassis\ Asset\ Tag" #--commit

```

```
# Verify change
echo
eval ilorest$s get AssetTag
```

Added the following patch:

```
{
  "AssetTag": "New Chassis Asset Tag"
}
```

AssetTag=New Chassis Asset Tag

<- Environment Preparation ; List of examples ; Restart iLO 5 Simulator ; Logout ->

1.6 Thermal

The [Thermal](#) data type describes the system thermal metrics. It provides the temperatures in different locations in the chassis as well as fan properties. It also offers the possibility to configure the fans with a specific cooling directive (Optimal, Enhanced, etc ...).

Thermal information belongs to the Thermal data type. However, the thermal configuration (ThermalConfig) is part of the Bios data type. Hence, to modify the ThermalConfig property, you will need to reboot the entire system to effectively apply this modification.

Starting with iLO firmware **2.30**, it is theoretically possible to modify the thermal configuration using the `Oem/Hpe/ThermalConfiguration` property of the Thermal data type without any system reset.

NOTE: As of the writing of this workshop, `Oem/Hpe/ThermalConfiguration` property cannot be modified (Read-Only property error). Hence you need to use a `rawpatch` command to modify it. Possible values for the `ThermalConfiguration` property are: `OptimalCooling`, `IncreasedCooling`, `MaximumCooling` and `EnhancedCooling`. The last two values differ from the Bios `ThermalConfig` property (`MaxCooling`, `EnhancedCPUCooling`)

1.6.1 Server Thermal Information

The following cell retrieves both fan and sensor properties, including temperatures.

```
[11]: # Select type of server: Dl (ProLiant DL360g10) or Sy (Synergy)
s="Dl"

# Retrieve server thermal properties
eval ilorest$s list --json --select Thermal
```

```
{
  "@odata.context": "/redfish/v1/$metadata#Thermal.Thermal",
  "@odata.etag": "W/\"5039E327\"",
  "@odata.id": "/redfish/v1/Chassis/1/Thermal",
  "@odata.type": "#Thermal.v1_6_2.Thermal",
  "Fans": [
```

```

{
  "@odata.id": "/redfish/v1/Chassis/1/Thermal#Fans/0",
  "MemberId": "0",
  "Name": "Fan 1",
  "Oem": {
    "Hpe": {
      "@odata.context": "/redfish/v1/$metadata#HpeServerFan.HpeServerFan",
      "@odata.type": "#HpeServerFan.v2_0_0.HpeServerFan",
      "HotPluggable": true,
      "Location": "System",
      "Redundant": true
    }
  },
  "Reading": 11,
  "ReadingUnits": "Percent",
  "Status": {
    "Health": "OK",
    "State": "Enabled"
  }
},
{
  "@odata.id": "/redfish/v1/Chassis/1/Thermal#Fans/1",
  "MemberId": "1",
  "Name": "Fan 2",
  "Oem": {
    "Hpe": {
      "@odata.context": "/redfish/v1/$metadata#HpeServerFan.HpeServerFan",
      "@odata.type": "#HpeServerFan.v2_0_0.HpeServerFan",
      "HotPluggable": true,
      "Location": "System",
      "Redundant": true
    }
  },
  "Reading": 11,
  "ReadingUnits": "Percent",
  "Status": {
    "Health": "OK",
    "State": "Enabled"
  }
},
{
  "@odata.id": "/redfish/v1/Chassis/1/Thermal#Fans/2",
  "MemberId": "2",
  "Name": "Fan 3",
  "Oem": {
    "Hpe": {
      "@odata.context": "/redfish/v1/$metadata#HpeServerFan.HpeServerFan",
      "@odata.type": "#HpeServerFan.v2_0_0.HpeServerFan",

```

```

        "HotPluggable": true,
        "Location": "System",
        "Redundant": true
    }
},
"Reading": 11,
"ReadingUnits": "Percent",
"Status": {
    "Health": "OK",
    "State": "Enabled"
}
},
{
    "@odata.id": "/redfish/v1/Chassis/1/Thermal#Fans/3",
    "MemberId": "3",
    "Name": "Fan 4",
    "Oem": {
        "Hpe": {
            "@odata.context": "/redfish/v1/$metadata#HpeServerFan.HpeServerFan",
            "@odata.type": "#HpeServerFan.v2_0_0.HpeServerFan",
            "HotPluggable": true,
            "Location": "System",
            "Redundant": true
        }
    },
    "Reading": 11,
    "ReadingUnits": "Percent",
    "Status": {
        "Health": "OK",
        "State": "Enabled"
    }
},
{
    "@odata.id": "/redfish/v1/Chassis/1/Thermal#Fans/4",
    "MemberId": "4",
    "Name": "Fan 5",
    "Oem": {
        "Hpe": {
            "@odata.context": "/redfish/v1/$metadata#HpeServerFan.HpeServerFan",
            "@odata.type": "#HpeServerFan.v2_0_0.HpeServerFan",
            "HotPluggable": true,
            "Location": "System",
            "Redundant": true
        }
    },
    "Reading": 11,
    "ReadingUnits": "Percent",
    "Status": {

```

```

        "Health": "OK",
        "State": "Enabled"
    }
},
{
    "@odata.id": "/redfish/v1/Chassis/1/Thermal#Fans/5",
    "MemberId": "5",
    "Name": "Fan 6",
    "Oem": {
        "Hpe": {
            "@odata.context": "/redfish/v1/$metadata#HpeServerFan.HpeServerFan",
            "@odata.type": "#HpeServerFan.v2_0_0.HpeServerFan",
            "HotPluggable": true,
            "Location": "System",
            "Redundant": true
        }
    },
    "Reading": 11,
    "ReadingUnits": "Percent",
    "Status": {
        "Health": "OK",
        "State": "Enabled"
    }
},
{
    "@odata.id": "/redfish/v1/Chassis/1/Thermal#Fans/6",
    "MemberId": "6",
    "Name": "Fan 7",
    "Oem": {
        "Hpe": {
            "@odata.context": "/redfish/v1/$metadata#HpeServerFan.HpeServerFan",
            "@odata.type": "#HpeServerFan.v2_0_0.HpeServerFan",
            "HotPluggable": true,
            "Location": "System",
            "Redundant": true
        }
    },
    "Reading": 11,
    "ReadingUnits": "Percent",
    "Status": {
        "Health": "OK",
        "State": "Enabled"
    }
}
],
"Id": "Thermal",
"Name": "Thermal",
"Oem": {

```

```

    "Hpe": {
      "@odata.context": "/redfish/v1/$metadata#HpeThermalExt.HpeThermalExt",
      "@odata.type": "#HpeThermalExt.v2_0_0.HpeThermalExt",
      "FanPercentMinimum": 0,
      "ThermalConfiguration": "OptimalCooling"
    }
  },
  "Temperatures": [
    {
      "@odata.id": "/redfish/v1/Chassis/1/Thermal#Temperatures/0",
      "MemberId": "0",
      "Name": "01-Inlet Ambient",
      "Oem": {
        "Hpe": {
          "@odata.context":
"/redfish/v1/$metadata#HpeSeaOfSensors.HpeSeaOfSensors",
          "@odata.type": "#HpeSeaOfSensors.v2_0_0.HpeSeaOfSensors",
          "LocationXmm": 15,
          "LocationYmm": 0
        }
      },
      "PhysicalContext": "Intake",
      "ReadingCelsius": 20,
      "SensorNumber": 1,
      "Status": {
        "Health": "OK",
        "State": "Enabled"
      },
      "UpperThresholdCritical": 42,
      "UpperThresholdFatal": 47,
      "UpperThresholdUser": 0
    },
    {
      "@odata.id": "/redfish/v1/Chassis/1/Thermal#Temperatures/1",
      "MemberId": "1",
      "Name": "02-CPU 1",
      "Oem": {
        "Hpe": {
          "@odata.context":
"/redfish/v1/$metadata#HpeSeaOfSensors.HpeSeaOfSensors",
          "@odata.type": "#HpeSeaOfSensors.v2_0_0.HpeSeaOfSensors",
          "LocationXmm": 11,
          "LocationYmm": 5
        }
      },
      "PhysicalContext": "CPU",
      "ReadingCelsius": 40,
      "SensorNumber": 2,

```



```

    "Status": {
      "Health": "OK",
      "State": "Enabled"
    },
    "UpperThresholdCritical": 70,
    "UpperThresholdFatal": null
  },
  {
    "@odata.id": "/redfish/v1/Chassis/1/Thermal#Temperatures/2",
    "MemberId": "2",
    "Name": "03-CPU 2",
    "Oem": {
      "Hpe": {
        "@odata.context":
"/redfish/v1/$metadata#HpeSeaOfSensors.HpeSeaOfSensors",
        "@odata.type": "#HpeSeaOfSensors.v2_0_0.HpeSeaOfSensors",
        "LocationXmm": 4,
        "LocationYmm": 5
      }
    },
    "PhysicalContext": "CPU",
    "ReadingCelsius": 40,
    "SensorNumber": 3,
    "Status": {
      "Health": "OK",
      "State": "Enabled"
    },
    "UpperThresholdCritical": 70,
    "UpperThresholdFatal": null
  },
  {
    "@odata.id": "/redfish/v1/Chassis/1/Thermal#Temperatures/3",
    "MemberId": "3",
    "Name": "04-P1 DIMM 1-6",
    "Oem": {
      "Hpe": {
        "@odata.context":
"/redfish/v1/$metadata#HpeSeaOfSensors.HpeSeaOfSensors",
        "@odata.type": "#HpeSeaOfSensors.v2_0_0.HpeSeaOfSensors",
        "LocationXmm": 8,
        "LocationYmm": 4
      }
    },
    "PhysicalContext": "SystemBoard",
    "ReadingCelsius": 0,
    "SensorNumber": 4,
    "Status": {
      "State": "Absent"
    }
  }

```

```

    },
    "UpperThresholdCritical": null,
    "UpperThresholdFatal": null
  },
  {
    "@odata.id": "/redfish/v1/Chassis/1/Thermal#Temperatures/4",
    "MemberId": "4",
    "Name": "05-PMM 1-6",
    "Oem": {
      "Hpe": {
        "@odata.context":
"/redfish/v1/$metadata#HpeSeaOfSensors.HpeSeaOfSensors",
        "@odata.type": "#HpeSeaOfSensors.v2_0_0.HpeSeaOfSensors",
        "LocationXmm": 8,
        "LocationYmm": 4
      }
    },
    "PhysicalContext": "SystemBoard",
    "ReadingCelsius": 0,
    "SensorNumber": 5,
    "Status": {
      "State": "Absent"
    },
    "UpperThresholdCritical": null,
    "UpperThresholdFatal": null
  },
  {
    "@odata.id": "/redfish/v1/Chassis/1/Thermal#Temperatures/5",
    "MemberId": "5",
    "Name": "06-P1 DIMM 7-12",
    "Oem": {
      "Hpe": {
        "@odata.context":
"/redfish/v1/$metadata#HpeSeaOfSensors.HpeSeaOfSensors",
        "@odata.type": "#HpeSeaOfSensors.v2_0_0.HpeSeaOfSensors",
        "LocationXmm": 13,
        "LocationYmm": 4
      }
    },
    "PhysicalContext": "SystemBoard",
    "ReadingCelsius": 28,
    "SensorNumber": 6,
    "Status": {
      "Health": "OK",
      "State": "Enabled"
    },
    "UpperThresholdCritical": 90,
    "UpperThresholdFatal": null
  }

```

```

    },
    {
      "@odata.id": "/redfish/v1/Chassis/1/Thermal#Temperatures/6",
      "MemberId": "6",
      "Name": "07-PMM 7-12",
      "Oem": {
        "Hpe": {
          "@odata.context":
"/redfish/v1/$metadata#HpeSeaOfSensors.HpeSeaOfSensors",
          "@odata.type": "#HpeSeaOfSensors.v2_0_0.HpeSeaOfSensors",
          "LocationXmm": 13,
          "LocationYmm": 4
        }
      },
      "PhysicalContext": "SystemBoard",
      "ReadingCelsius": 0,
      "SensorNumber": 7,
      "Status": {
        "State": "Absent"
      },
      "UpperThresholdCritical": null,
      "UpperThresholdFatal": null
    },
    {
      "@odata.id": "/redfish/v1/Chassis/1/Thermal#Temperatures/7",
      "MemberId": "7",
      "Name": "08-P2 DIMM 1-6",
      "Oem": {
        "Hpe": {
          "@odata.context":
"/redfish/v1/$metadata#HpeSeaOfSensors.HpeSeaOfSensors",
          "@odata.type": "#HpeSeaOfSensors.v2_0_0.HpeSeaOfSensors",
          "LocationXmm": 1,
          "LocationYmm": 4
        }
      },
      "PhysicalContext": "SystemBoard",
      "ReadingCelsius": 0,
      "SensorNumber": 8,
      "Status": {
        "State": "Absent"
      },
      "UpperThresholdCritical": null,
      "UpperThresholdFatal": null
    },
    {
      "@odata.id": "/redfish/v1/Chassis/1/Thermal#Temperatures/8",
      "MemberId": "8",

```

```

    "Name": "09-PMM 1-6",
    "Oem": {
      "Hpe": {
        "@odata.context":
"/redfish/v1/$metadata#HpeSeaOfSensors.HpeSeaOfSensors",
        "@odata.type": "#HpeSeaOfSensors.v2_0_0.HpeSeaOfSensors",
        "LocationXmm": 1,
        "LocationYmm": 4
      }
    },
    "PhysicalContext": "SystemBoard",
    "ReadingCelsius": 0,
    "SensorNumber": 9,
    "Status": {
      "State": "Absent"
    },
    "UpperThresholdCritical": null,
    "UpperThresholdFatal": null
  },
  {
    "@odata.id": "/redfish/v1/Chassis/1/Thermal#Temperatures/9",
    "MemberId": "9",
    "Name": "10-P2 DIMM 7-12",
    "Oem": {
      "Hpe": {
        "@odata.context":
"/redfish/v1/$metadata#HpeSeaOfSensors.HpeSeaOfSensors",
        "@odata.type": "#HpeSeaOfSensors.v2_0_0.HpeSeaOfSensors",
        "LocationXmm": 6,
        "LocationYmm": 4
      }
    },
    "PhysicalContext": "SystemBoard",
    "ReadingCelsius": 28,
    "SensorNumber": 10,
    "Status": {
      "Health": "OK",
      "State": "Enabled"
    },
    "UpperThresholdCritical": 90,
    "UpperThresholdFatal": null
  },
  {
    "@odata.id": "/redfish/v1/Chassis/1/Thermal#Temperatures/10",
    "MemberId": "10",
    "Name": "11-PMM 7-12",
    "Oem": {
      "Hpe": {

```

```

        "@odata.context":
"/redfish/v1/$metadata#HpeSeaOfSensors.HpeSeaOfSensors",
        "@odata.type": "#HpeSeaOfSensors.v2_0_0.HpeSeaOfSensors",
        "LocationXmm": 6,
        "LocationYmm": 4
    }
},
"PhysicalContext": "SystemBoard",
"ReadingCelsius": 0,
"SensorNumber": 11,
"Status": {
    "State": "Absent"
},
"UpperThresholdCritical": null,
"UpperThresholdFatal": null
},
{
    "@odata.id": "/redfish/v1/Chassis/1/Thermal#Temperatures/11",
    "MemberId": "11",
    "Name": "12-HD Max",
    "Oem": {
        "Hpe": {
            "@odata.context":
"/redfish/v1/$metadata#HpeSeaOfSensors.HpeSeaOfSensors",
            "@odata.type": "#HpeSeaOfSensors.v2_0_0.HpeSeaOfSensors",
            "LocationXmm": 11,
            "LocationYmm": 0
        }
    },
    "PhysicalContext": "SystemBoard",
    "ReadingCelsius": 35,
    "SensorNumber": 12,
    "Status": {
        "Health": "OK",
        "State": "Enabled"
    },
    "UpperThresholdCritical": 60,
    "UpperThresholdFatal": null
},
{
    "@odata.id": "/redfish/v1/Chassis/1/Thermal#Temperatures/12",
    "MemberId": "12",
    "Name": "13-Exp Bay Drive",
    "Oem": {
        "Hpe": {
            "@odata.context":
"/redfish/v1/$metadata#HpeSeaOfSensors.HpeSeaOfSensors",
            "@odata.type": "#HpeSeaOfSensors.v2_0_0.HpeSeaOfSensors",

```

```

        "LocationXmm": 14,
        "LocationYmm": 0
    }
},
"PhysicalContext": "SystemBoard",
"ReadingCelsius": 0,
"SensorNumber": 13,
"Status": {
    "State": "Absent"
},
"UpperThresholdCritical": null,
"UpperThresholdFatal": null
},
{
    "@odata.id": "/redfish/v1/Chassis/1/Thermal#Temperatures/13",
    "MemberId": "13",
    "Name": "14-Stor Batt 1",
    "Oem": {
        "Hpe": {
            "@odata.context":
"/redfish/v1/$metadata#HpeSeaOfSensors.HpeSeaOfSensors",
            "@odata.type": "#HpeSeaOfSensors.v2_0_0.HpeSeaOfSensors",
            "LocationXmm": 5,
            "LocationYmm": 0
        }
    },
    "PhysicalContext": "SystemBoard",
    "ReadingCelsius": 28,
    "SensorNumber": 14,
    "Status": {
        "Health": "OK",
        "State": "Enabled"
    },
    "UpperThresholdCritical": 60,
    "UpperThresholdFatal": null
},
{
    "@odata.id": "/redfish/v1/Chassis/1/Thermal#Temperatures/14",
    "MemberId": "14",
    "Name": "15-Front Ambient",
    "Oem": {
        "Hpe": {
            "@odata.context":
"/redfish/v1/$metadata#HpeSeaOfSensors.HpeSeaOfSensors",
            "@odata.type": "#HpeSeaOfSensors.v2_0_0.HpeSeaOfSensors",
            "LocationXmm": 9,
            "LocationYmm": 1
        }
    }
}

```

```

    },
    "PhysicalContext": "Intake",
    "ReadingCelsius": 23,
    "SensorNumber": 15,
    "Status": {
        "Health": "OK",
        "State": "Enabled"
    },
    "UpperThresholdCritical": 60,
    "UpperThresholdFatal": null
},
{
    "@odata.id": "/redfish/v1/Chassis/1/Thermal#Temperatures/15",
    "MemberId": "15",
    "Name": "16-VR P1",
    "Oem": {
        "Hpe": {
            "@odata.context":
"/redfish/v1/$metadata#HpeSeaOfSensors.HpeSeaOfSensors",
            "@odata.type": "#HpeSeaOfSensors.v2_0_0.HpeSeaOfSensors",
            "LocationXmm": 11,
            "LocationYmm": 3
        }
    },
    "PhysicalContext": "SystemBoard",
    "ReadingCelsius": 31,
    "SensorNumber": 16,
    "Status": {
        "Health": "OK",
        "State": "Enabled"
    },
    "UpperThresholdCritical": 115,
    "UpperThresholdFatal": 120
},
{
    "@odata.id": "/redfish/v1/Chassis/1/Thermal#Temperatures/16",
    "MemberId": "16",
    "Name": "17-VR P2",
    "Oem": {
        "Hpe": {
            "@odata.context":
"/redfish/v1/$metadata#HpeSeaOfSensors.HpeSeaOfSensors",
            "@odata.type": "#HpeSeaOfSensors.v2_0_0.HpeSeaOfSensors",
            "LocationXmm": 4,
            "LocationYmm": 3
        }
    },
    "PhysicalContext": "SystemBoard",

```

```

    "ReadingCelsius": 37,
    "SensorNumber": 17,
    "Status": {
        "Health": "OK",
        "State": "Enabled"
    },
    "UpperThresholdCritical": 115,
    "UpperThresholdFatal": 120
},
{
    "@odata.id": "/redfish/v1/Chassis/1/Thermal#Temperatures/17",
    "MemberId": "17",
    "Name": "18-VR P1 Mem 1",
    "Oem": {
        "Hpe": {
            "@odata.context":
"/redfish/v1/$metadata#HpeSeaOfSensors.HpeSeaOfSensors",
            "@odata.type": "#HpeSeaOfSensors.v2_0_0.HpeSeaOfSensors",
            "LocationXmm": 8,
            "LocationYmm": 2
        }
    },
    "PhysicalContext": "SystemBoard",
    "ReadingCelsius": 27,
    "SensorNumber": 18,
    "Status": {
        "Health": "OK",
        "State": "Enabled"
    },
    "UpperThresholdCritical": 115,
    "UpperThresholdFatal": 120
},
{
    "@odata.id": "/redfish/v1/Chassis/1/Thermal#Temperatures/18",
    "MemberId": "18",
    "Name": "19-VR P1 Mem 2",
    "Oem": {
        "Hpe": {
            "@odata.context":
"/redfish/v1/$metadata#HpeSeaOfSensors.HpeSeaOfSensors",
            "@odata.type": "#HpeSeaOfSensors.v2_0_0.HpeSeaOfSensors",
            "LocationXmm": 13,
            "LocationYmm": 2
        }
    },
    "PhysicalContext": "SystemBoard",
    "ReadingCelsius": 28,
    "SensorNumber": 19,

```



```

    "Status": {
      "Health": "OK",
      "State": "Enabled"
    },
    "UpperThresholdCritical": 115,
    "UpperThresholdFatal": 120
  },
  {
    "@odata.id": "/redfish/v1/Chassis/1/Thermal#Temperatures/19",
    "MemberId": "19",
    "Name": "20-VR P2 Mem 1",
    "Oem": {
      "Hpe": {
        "@odata.context":
"/redfish/v1/$metadata#HpeSeaOfSensors.HpeSeaOfSensors",
        "@odata.type": "#HpeSeaOfSensors.v2_0_0.HpeSeaOfSensors",
        "LocationXmm": 1,
        "LocationYmm": 2
      }
    },
    "PhysicalContext": "SystemBoard",
    "ReadingCelsius": 29,
    "SensorNumber": 20,
    "Status": {
      "Health": "OK",
      "State": "Enabled"
    },
    "UpperThresholdCritical": 115,
    "UpperThresholdFatal": 120
  },
  {
    "@odata.id": "/redfish/v1/Chassis/1/Thermal#Temperatures/20",
    "MemberId": "20",
    "Name": "21-VR P2 Mem 2",
    "Oem": {
      "Hpe": {
        "@odata.context":
"/redfish/v1/$metadata#HpeSeaOfSensors.HpeSeaOfSensors",
        "@odata.type": "#HpeSeaOfSensors.v2_0_0.HpeSeaOfSensors",
        "LocationXmm": 6,
        "LocationYmm": 2
      }
    },
    "PhysicalContext": "SystemBoard",
    "ReadingCelsius": 34,
    "SensorNumber": 21,
    "Status": {
      "Health": "OK",

```



```

    "UpperThresholdCritical": 110,
    "UpperThresholdFatal": 115
  },
  {
    "@odata.id": "/redfish/v1/Chassis/1/Thermal#Temperatures/23",
    "MemberId": "23",
    "Name": "24-BMC Zone",
    "Oem": {
      "Hpe": {
        "@odata.context":
"/redfish/v1/$metadata#HpeSeaOfSensors.HpeSeaOfSensors",
        "@odata.type": "#HpeSeaOfSensors.v2_0_0.HpeSeaOfSensors",
        "LocationXmm": 9,
        "LocationYmm": 13
      }
    },
    "PhysicalContext": "SystemBoard",
    "ReadingCelsius": 41,
    "SensorNumber": 24,
    "Status": {
      "Health": "OK",
      "State": "Enabled"
    },
    "UpperThresholdCritical": 90,
    "UpperThresholdFatal": 95
  },
  {
    "@odata.id": "/redfish/v1/Chassis/1/Thermal#Temperatures/24",
    "MemberId": "24",
    "Name": "25-HD Controller",
    "Oem": {
      "Hpe": {
        "@odata.context":
"/redfish/v1/$metadata#HpeSeaOfSensors.HpeSeaOfSensors",
        "@odata.type": "#HpeSeaOfSensors.v2_0_0.HpeSeaOfSensors",
        "LocationXmm": 8,
        "LocationYmm": 9
      }
    },
    "PhysicalContext": "SystemBoard",
    "ReadingCelsius": 58,
    "SensorNumber": 25,
    "Status": {
      "Health": "OK",
      "State": "Enabled"
    },
    "UpperThresholdCritical": 100,
    "UpperThresholdFatal": null
  }

```

```

    },
    {
      "@odata.id": "/redfish/v1/Chassis/1/Thermal#Temperatures/25",
      "MemberId": "25",
      "Name": "26-HD Cntlr Zone",
      "Oem": {
        "Hpe": {
          "@odata.context":
"/redfish/v1/$metadata#HpeSeaOfSensors.HpeSeaOfSensors",
          "@odata.type": "#HpeSeaOfSensors.v2_0_0.HpeSeaOfSensors",
          "LocationXmm": 8,
          "LocationYmm": 8
        }
      },
      "PhysicalContext": "SystemBoard",
      "ReadingCelsius": 32,
      "SensorNumber": 26,
      "Status": {
        "Health": "OK",
        "State": "Enabled"
      },
      "UpperThresholdCritical": 85,
      "UpperThresholdFatal": 90
    },
    {
      "@odata.id": "/redfish/v1/Chassis/1/Thermal#Temperatures/26",
      "MemberId": "26",
      "Name": "27-LOM",
      "Oem": {
        "Hpe": {
          "@odata.context":
"/redfish/v1/$metadata#HpeSeaOfSensors.HpeSeaOfSensors",
          "@odata.type": "#HpeSeaOfSensors.v2_0_0.HpeSeaOfSensors",
          "LocationXmm": 7,
          "LocationYmm": 13
        }
      },
      "PhysicalContext": "SystemBoard",
      "ReadingCelsius": 49,
      "SensorNumber": 27,
      "Status": {
        "Health": "OK",
        "State": "Enabled"
      },
      "UpperThresholdCritical": 100,
      "UpperThresholdFatal": null
    },
  ],
  {

```

```

"@odata.id": "/redfish/v1/Chassis/1/Thermal#Temperatures/27",
"MemberId": "27",
"Name": "28-LOM Card",
"Oem": {
  "Hpe": {
    "@odata.context":
"/redfish/v1/$metadata#HpeSeaOfSensors.HpeSeaOfSensors",
    "@odata.type": "#HpeSeaOfSensors.v2_0_0.HpeSeaOfSensors",
    "LocationXmm": 14,
    "LocationYmm": 14
  }
},
"PhysicalContext": "SystemBoard",
"ReadingCelsius": 0,
"SensorNumber": 28,
"Status": {
  "State": "Absent"
},
"UpperThresholdCritical": null,
"UpperThresholdFatal": null
},
{
  "@odata.id": "/redfish/v1/Chassis/1/Thermal#Temperatures/28",
"MemberId": "28",
"Name": "29-I/O Zone",
"Oem": {
  "Hpe": {
    "@odata.context":
"/redfish/v1/$metadata#HpeSeaOfSensors.HpeSeaOfSensors",
    "@odata.type": "#HpeSeaOfSensors.v2_0_0.HpeSeaOfSensors",
    "LocationXmm": 14,
    "LocationYmm": 11
  }
},
"PhysicalContext": "SystemBoard",
"ReadingCelsius": 31,
"SensorNumber": 29,
"Status": {
  "Health": "OK",
  "State": "Enabled"
},
"UpperThresholdCritical": 90,
"UpperThresholdFatal": 95
},
{
  "@odata.id": "/redfish/v1/Chassis/1/Thermal#Temperatures/29",
"MemberId": "29",
"Name": "30-PCI 1",

```

```

    "Oem": {
      "Hpe": {
        "@odata.context":
"/redfish/v1/$metadata#HpeSeaOfSensors.HpeSeaOfSensors",
        "@odata.type": "#HpeSeaOfSensors.v2_0_0.HpeSeaOfSensors",
        "LocationXmm": 11,
        "LocationYmm": 8
      }
    },
    "PhysicalContext": "SystemBoard",
    "ReadingCelsius": 0,
    "SensorNumber": 30,
    "Status": {
      "State": "Absent"
    },
    "UpperThresholdCritical": null,
    "UpperThresholdFatal": null
  },
  {
    "@odata.id": "/redfish/v1/Chassis/1/Thermal#Temperatures/30",
    "MemberId": "30",
    "Name": "31-PCI 1 Zone",
    "Oem": {
      "Hpe": {
        "@odata.context":
"/redfish/v1/$metadata#HpeSeaOfSensors.HpeSeaOfSensors",
        "@odata.type": "#HpeSeaOfSensors.v2_0_0.HpeSeaOfSensors",
        "LocationXmm": 11,
        "LocationYmm": 9
      }
    },
    "PhysicalContext": "SystemBoard",
    "ReadingCelsius": 31,
    "SensorNumber": 31,
    "Status": {
      "Health": "OK",
      "State": "Enabled"
    },
    "UpperThresholdCritical": 90,
    "UpperThresholdFatal": 95
  },
  {
    "@odata.id": "/redfish/v1/Chassis/1/Thermal#Temperatures/31",
    "MemberId": "31",
    "Name": "32-PCI 2",
    "Oem": {
      "Hpe": {
        "@odata.context":

```

```

"/redfish/v1/$metadata#HpeSeaOfSensors.HpeSeaOfSensors",
  "@odata.type": "#HpeSeaOfSensors.v2_0_0.HpeSeaOfSensors",
  "LocationXmm": 11,
  "LocationYmm": 8
},
{
  "PhysicalContext": "SystemBoard",
  "ReadingCelsius": 0,
  "SensorNumber": 32,
  "Status": {
    "State": "Absent"
  },
  "UpperThresholdCritical": null,
  "UpperThresholdFatal": null
},
{
  "@odata.id": "/redfish/v1/Chassis/1/Thermal#Temperatures/32",
  "MemberId": "32",
  "Name": "33-PCI 2 Zone",
  "Oem": {
    "Hpe": {
      "@odata.context":
"/redfish/v1/$metadata#HpeSeaOfSensors.HpeSeaOfSensors",
      "@odata.type": "#HpeSeaOfSensors.v2_0_0.HpeSeaOfSensors",
      "LocationXmm": 11,
      "LocationYmm": 9
    }
  },
  "PhysicalContext": "SystemBoard",
  "ReadingCelsius": 30,
  "SensorNumber": 33,
  "Status": {
    "Health": "OK",
    "State": "Enabled"
  },
  "UpperThresholdCritical": 90,
  "UpperThresholdFatal": 95
},
{
  "@odata.id": "/redfish/v1/Chassis/1/Thermal#Temperatures/33",
  "MemberId": "33",
  "Name": "34-PCI 3",
  "Oem": {
    "Hpe": {
      "@odata.context":
"/redfish/v1/$metadata#HpeSeaOfSensors.HpeSeaOfSensors",
      "@odata.type": "#HpeSeaOfSensors.v2_0_0.HpeSeaOfSensors",
      "LocationXmm": 5,

```

```

        "LocationYmm": 11
    },
    "PhysicalContext": "SystemBoard",
    "ReadingCelsius": 0,
    "SensorNumber": 34,
    "Status": {
        "State": "Absent"
    },
    "UpperThresholdCritical": null,
    "UpperThresholdFatal": null
},
{
    "@odata.id": "/redfish/v1/Chassis/1/Thermal#Temperatures/34",
    "MemberId": "34",
    "Name": "35-PCI 3 Zone",
    "Oem": {
        "Hpe": {
            "@odata.context":
"/redfish/v1/$metadata#HpeSeaOfSensors.HpeSeaOfSensors",
            "@odata.type": "#HpeSeaOfSensors.v2_0_0.HpeSeaOfSensors",
            "LocationXmm": 5,
            "LocationYmm": 13
        }
    },
    "PhysicalContext": "SystemBoard",
    "ReadingCelsius": 0,
    "SensorNumber": 35,
    "Status": {
        "State": "Absent"
    },
    "UpperThresholdCritical": null,
    "UpperThresholdFatal": null
},
{
    "@odata.id": "/redfish/v1/Chassis/1/Thermal#Temperatures/35",
    "MemberId": "35",
    "Name": "37-Rear HD Max",
    "Oem": {
        "Hpe": {
            "@odata.context":
"/redfish/v1/$metadata#HpeSeaOfSensors.HpeSeaOfSensors",
            "@odata.type": "#HpeSeaOfSensors.v2_0_0.HpeSeaOfSensors",
            "LocationXmm": 14,
            "LocationYmm": 14
        }
    },
    "PhysicalContext": "SystemBoard",

```



```

    "ReadingCelsius": 0,
    "SensorNumber": 36,
    "Status": {
        "State": "Absent"
    },
    "UpperThresholdCritical": null,
    "UpperThresholdFatal": null
},
{
    "@odata.id": "/redfish/v1/Chassis/1/Thermal#Temperatures/36",
    "MemberId": "36",
    "Name": "38-Battery Zone",
    "Oem": {
        "Hpe": {
            "@odata.context":
"/redfish/v1/$metadata#HpeSeaOfSensors.HpeSeaOfSensors",
            "@odata.type": "#HpeSeaOfSensors.v2_0_0.HpeSeaOfSensors",
            "LocationXmm": 7,
            "LocationYmm": 10
        }
    },
    "PhysicalContext": "SystemBoard",
    "ReadingCelsius": 36,
    "SensorNumber": 37,
    "Status": {
        "Health": "OK",
        "State": "Enabled"
    },
    "UpperThresholdCritical": 75,
    "UpperThresholdFatal": 80
},
{
    "@odata.id": "/redfish/v1/Chassis/1/Thermal#Temperatures/37",
    "MemberId": "37",
    "Name": "39-P/S 1 Inlet",
    "Oem": {
        "Hpe": {
            "@odata.context":
"/redfish/v1/$metadata#HpeSeaOfSensors.HpeSeaOfSensors",
            "@odata.type": "#HpeSeaOfSensors.v2_0_0.HpeSeaOfSensors",
            "LocationXmm": 1,
            "LocationYmm": 10
        }
    },
    "PhysicalContext": "PowerSupply",
    "ReadingCelsius": 30,
    "SensorNumber": 38,
    "Status": {

```

```

        "Health": "OK",
        "State": "Enabled"
    },
    "UpperThresholdCritical": null,
    "UpperThresholdFatal": null
},
{
    "@odata.id": "/redfish/v1/Chassis/1/Thermal#Temperatures/38",
    "MemberId": "38",
    "Name": "40-P/S 2 Inlet",
    "Oem": {
        "Hpe": {
            "@odata.context":
"/redfish/v1/$metadata#HpeSeaOfSensors.HpeSeaOfSensors",
            "@odata.type": "#HpeSeaOfSensors.v2_0_0.HpeSeaOfSensors",
            "LocationXmm": 4,
            "LocationYmm": 10
        }
    },
    "PhysicalContext": "PowerSupply",
    "ReadingCelsius": 34,
    "SensorNumber": 39,
    "Status": {
        "Health": "OK",
        "State": "Enabled"
    },
    "UpperThresholdCritical": null,
    "UpperThresholdFatal": null
},
{
    "@odata.id": "/redfish/v1/Chassis/1/Thermal#Temperatures/39",
    "MemberId": "39",
    "Name": "41-P/S 1",
    "Oem": {
        "Hpe": {
            "@odata.context":
"/redfish/v1/$metadata#HpeSeaOfSensors.HpeSeaOfSensors",
            "@odata.type": "#HpeSeaOfSensors.v2_0_0.HpeSeaOfSensors",
            "LocationXmm": 1,
            "LocationYmm": 13
        }
    },
    "PhysicalContext": "PowerSupply",
    "ReadingCelsius": 40,
    "SensorNumber": 40,
    "Status": {
        "Health": "OK",
        "State": "Enabled"
    }
}

```

```

    },
    "UpperThresholdCritical": null,
    "UpperThresholdFatal": null
  },
  {
    "@odata.id": "/redfish/v1/Chassis/1/Thermal#Temperatures/40",
    "MemberId": "40",
    "Name": "42-P/S 2",
    "Oem": {
      "Hpe": {
        "@odata.context":
"/redfish/v1/$metadata#HpeSeaOfSensors.HpeSeaOfSensors",
        "@odata.type": "#HpeSeaOfSensors.v2_0_0.HpeSeaOfSensors",
        "LocationXmm": 3,
        "LocationYmm": 13
      }
    },
    "PhysicalContext": "PowerSupply",
    "ReadingCelsius": 40,
    "SensorNumber": 41,
    "Status": {
      "Health": "OK",
      "State": "Enabled"
    },
    "UpperThresholdCritical": null,
    "UpperThresholdFatal": null
  },
  {
    "@odata.id": "/redfish/v1/Chassis/1/Thermal#Temperatures/41",
    "MemberId": "41",
    "Name": "43-E-Fuse",
    "Oem": {
      "Hpe": {
        "@odata.context":
"/redfish/v1/$metadata#HpeSeaOfSensors.HpeSeaOfSensors",
        "@odata.type": "#HpeSeaOfSensors.v2_0_0.HpeSeaOfSensors",
        "LocationXmm": 4,
        "LocationYmm": 9
      }
    },
    "PhysicalContext": "PowerSupply",
    "ReadingCelsius": 27,
    "SensorNumber": 42,
    "Status": {
      "Health": "OK",
      "State": "Enabled"
    },
    "UpperThresholdCritical": 100,

```

```

    "UpperThresholdFatal": null
  },
  {
    "@odata.id": "/redfish/v1/Chassis/1/Thermal#Temperatures/42",
    "MemberId": "42",
    "Name": "44-P/S 2 Zone",
    "Oem": {
      "Hpe": {
        "@odata.context":
"/redfish/v1/$metadata#HpeSeaOfSensors.HpeSeaOfSensors",
        "@odata.type": "#HpeSeaOfSensors.v2_0_0.HpeSeaOfSensors",
        "LocationXmm": 4,
        "LocationYmm": 7
      }
    },
    "PhysicalContext": "PowerSupply",
    "ReadingCelsius": 34,
    "SensorNumber": 43,
    "Status": {
      "Health": "OK",
      "State": "Enabled"
    },
    "UpperThresholdCritical": 75,
    "UpperThresholdFatal": 80
  },
  {
    "@odata.id": "/redfish/v1/Chassis/1/Thermal#Temperatures/43",
    "MemberId": "43",
    "Name": "61-AHCI HD Max",
    "Oem": {
      "Hpe": {
        "@odata.context":
"/redfish/v1/$metadata#HpeSeaOfSensors.HpeSeaOfSensors",
        "@odata.type": "#HpeSeaOfSensors.v2_0_0.HpeSeaOfSensors",
        "LocationXmm": 11,
        "LocationYmm": 0
      }
    },
    "PhysicalContext": "SystemBoard",
    "ReadingCelsius": 0,
    "SensorNumber": 44,
    "Status": {
      "State": "Absent"
    },
    "UpperThresholdCritical": null,
    "UpperThresholdFatal": null
  },
  {

```

```

    "@odata.id": "/redfish/v1/Chassis/1/Thermal#Temperatures/44",
    "MemberId": "44",
    "Name": "69-PCI 1 M2",
    "Oem": {
      "Hpe": {
        "@odata.context":
"/redfish/v1/$metadata#HpeSeaOfSensors.HpeSeaOfSensors",
        "@odata.type": "#HpeSeaOfSensors.v2_0_0.HpeSeaOfSensors",
        "LocationXmm": 11,
        "LocationYmm": 8
      }
    },
    "PhysicalContext": "SystemBoard",
    "ReadingCelsius": 0,
    "SensorNumber": 45,
    "Status": {
      "State": "Absent"
    },
    "UpperThresholdCritical": null,
    "UpperThresholdFatal": null
  },
  {
    "@odata.id": "/redfish/v1/Chassis/1/Thermal#Temperatures/45",
    "MemberId": "45",
    "Name": "70-PCI 1 M2 Zn",
    "Oem": {
      "Hpe": {
        "@odata.context":
"/redfish/v1/$metadata#HpeSeaOfSensors.HpeSeaOfSensors",
        "@odata.type": "#HpeSeaOfSensors.v2_0_0.HpeSeaOfSensors",
        "LocationXmm": 11,
        "LocationYmm": 9
      }
    },
    "PhysicalContext": "SystemBoard",
    "ReadingCelsius": 0,
    "SensorNumber": 46,
    "Status": {
      "State": "Absent"
    },
    "UpperThresholdCritical": null,
    "UpperThresholdFatal": null
  },
  {
    "@odata.id": "/redfish/v1/Chassis/1/Thermal#Temperatures/46",
    "MemberId": "46",
    "Name": "71-PCI 2 M2",
    "Oem": {

```

```

        "Hpe": {
            "@odata.context":
"/redfish/v1/$metadata#HpeSeaOfSensors.HpeSeaOfSensors",
            "@odata.type": "#HpeSeaOfSensors.v2_0_0.HpeSeaOfSensors",
            "LocationXmm": 11,
            "LocationYmm": 8
        }
    },
    "PhysicalContext": "SystemBoard",
    "ReadingCelsius": 0,
    "SensorNumber": 47,
    "Status": {
        "State": "Absent"
    },
    "UpperThresholdCritical": null,
    "UpperThresholdFatal": null
},
{
    "@odata.id": "/redfish/v1/Chassis/1/Thermal#Temperatures/47",
    "MemberId": "47",
    "Name": "72-PCI 2 M2 Zn",
    "Oem": {
        "Hpe": {
            "@odata.context":
"/redfish/v1/$metadata#HpeSeaOfSensors.HpeSeaOfSensors",
            "@odata.type": "#HpeSeaOfSensors.v2_0_0.HpeSeaOfSensors",
            "LocationXmm": 11,
            "LocationYmm": 9
        }
    },
    "PhysicalContext": "SystemBoard",
    "ReadingCelsius": 0,
    "SensorNumber": 48,
    "Status": {
        "State": "Absent"
    },
    "UpperThresholdCritical": null,
    "UpperThresholdFatal": null
},
{
    "@odata.id": "/redfish/v1/Chassis/1/Thermal#Temperatures/48",
    "MemberId": "48",
    "Name": "73-PCI 3 M2",
    "Oem": {
        "Hpe": {
            "@odata.context":
"/redfish/v1/$metadata#HpeSeaOfSensors.HpeSeaOfSensors",
            "@odata.type": "#HpeSeaOfSensors.v2_0_0.HpeSeaOfSensors",

```

```

        "LocationXmm": 5,
        "LocationYmm": 11
    }
},
"PhysicalContext": "SystemBoard",
"ReadingCelsius": 0,
"SensorNumber": 49,
"Status": {
    "State": "Absent"
},
"UpperThresholdCritical": null,
"UpperThresholdFatal": null
},
{
    "@odata.id": "/redfish/v1/Chassis/1/Thermal#Temperatures/49",
    "MemberId": "49",
    "Name": "74-PCI 3 M2 Zn",
    "Oem": {
        "Hpe": {
            "@odata.context":
"/redfish/v1/$metadata#HpeSeaOfSensors.HpeSeaOfSensors",
            "@odata.type": "#HpeSeaOfSensors.v2_0_0.HpeSeaOfSensors",
            "LocationXmm": 5,
            "LocationYmm": 13
        }
    },
    "PhysicalContext": "SystemBoard",
    "ReadingCelsius": 0,
    "SensorNumber": 50,
    "Status": {
        "State": "Absent"
    },
    "UpperThresholdCritical": null,
    "UpperThresholdFatal": null
}
]
}

```

If you want only sensor information, you can use the `serverinfo` macro command like that found in the following cell

```

[12]: # Select type of server: Dl (ProLiant DL360g10) or Sy (Synergy)
s="Dl"

# Retrieve data type used by the server info macro command
eval ilorest$s serverinfo --thermal

```

Thermal Information:

Sensor #1:

Location: Intake
Current Temp: 20 C
Critical Threshold: 42 C
Fatal Threshold: 47 C
Health: OK

Sensor #2:

Location: CPU
Current Temp: 40 C
Critical Threshold: 70 C
Fatal Threshold: -
Health: OK

Sensor #3:

Location: CPU
Current Temp: 40 C
Critical Threshold: 70 C
Fatal Threshold: -
Health: OK

Sensor #6:

Location: SystemBoard
Current Temp: 28 C
Critical Threshold: 90 C
Fatal Threshold: -
Health: OK

Sensor #10:

Location: SystemBoard
Current Temp: 28 C
Critical Threshold: 90 C
Fatal Threshold: -
Health: OK

Sensor #12:

Location: SystemBoard
Current Temp: 35 C
Critical Threshold: 60 C
Fatal Threshold: -
Health: OK

Sensor #14:

Location: SystemBoard
Current Temp: 28 C
Critical Threshold: 60 C
Fatal Threshold: -
Health: OK

Sensor #15:

Location: Intake
Current Temp: 23 C
Critical Threshold: 60 C

Fatal Threshold: -
Health: OK

Sensor #16:
Location: SystemBoard
Current Temp: 31 C
Critical Threshold: 115 C
Fatal Threshold: 120 C
Health: OK

Sensor #17:
Location: SystemBoard
Current Temp: 37 C
Critical Threshold: 115 C
Fatal Threshold: 120 C
Health: OK

Sensor #18:
Location: SystemBoard
Current Temp: 27 C
Critical Threshold: 115 C
Fatal Threshold: 120 C
Health: OK

Sensor #19:
Location: SystemBoard
Current Temp: 28 C
Critical Threshold: 115 C
Fatal Threshold: 120 C
Health: OK

Sensor #20:
Location: SystemBoard
Current Temp: 29 C
Critical Threshold: 115 C
Fatal Threshold: 120 C
Health: OK

Sensor #21:
Location: SystemBoard
Current Temp: 34 C
Critical Threshold: 115 C
Fatal Threshold: 120 C
Health: OK

Sensor #22:
Location: SystemBoard
Current Temp: 36 C
Critical Threshold: 100 C
Fatal Threshold: -
Health: OK

Sensor #23:
Location: SystemBoard
Current Temp: 72 C
Critical Threshold: 110 C

Fatal Threshold: 115 C
Health: OK

Sensor #24:
Location: SystemBoard
Current Temp: 41 C
Critical Threshold: 90 C
Fatal Threshold: 95 C
Health: OK

Sensor #25:
Location: SystemBoard
Current Temp: 58 C
Critical Threshold: 100 C
Fatal Threshold: -
Health: OK

Sensor #26:
Location: SystemBoard
Current Temp: 32 C
Critical Threshold: 85 C
Fatal Threshold: 90 C
Health: OK

Sensor #27:
Location: SystemBoard
Current Temp: 49 C
Critical Threshold: 100 C
Fatal Threshold: -
Health: OK

Sensor #29:
Location: SystemBoard
Current Temp: 31 C
Critical Threshold: 90 C
Fatal Threshold: 95 C
Health: OK

Sensor #31:
Location: SystemBoard
Current Temp: 31 C
Critical Threshold: 90 C
Fatal Threshold: 95 C
Health: OK

Sensor #33:
Location: SystemBoard
Current Temp: 30 C
Critical Threshold: 90 C
Fatal Threshold: 95 C
Health: OK

Sensor #37:
Location: SystemBoard
Current Temp: 36 C
Critical Threshold: 75 C

```

        Fatal Threshold: 80 C
        Health: OK
Sensor #38:
    Location: PowerSupply
    Current Temp: 30 C
    Critical Threshold: -
    Fatal Threshold: -
    Health: OK
Sensor #39:
    Location: PowerSupply
    Current Temp: 34 C
    Critical Threshold: -
    Fatal Threshold: -
    Health: OK
Sensor #40:
    Location: PowerSupply
    Current Temp: 40 C
    Critical Threshold: -
    Fatal Threshold: -
    Health: OK
Sensor #41:
    Location: PowerSupply
    Current Temp: 40 C
    Critical Threshold: -
    Fatal Threshold: -
    Health: OK
Sensor #42:
    Location: PowerSupply
    Current Temp: 27 C
    Critical Threshold: 100 C
    Fatal Threshold: -
    Health: OK
Sensor #43:
    Location: PowerSupply
    Current Temp: 34 C
    Critical Threshold: 75 C
    Fatal Threshold: 80 C
    Health: OK

```

1.6.2 ThermalConfiguration

The following cells retrieve the thermal configuration from the Bios data type first, and then from the Thermal data type.

```

[13]: # Select type of server: Dl (ProLiant DL360g10) or Sy (Synergy)
      s="Sy"

      # Select data type

```

```
eval ilorest$s select Bios.

# Retrieve property
eval ilorest$s get Attributes/ThermalConfig
echo

# Retrieve possible values for the ThermalConfig property
# NOTE: The following command fails against an iLO 5 simulator because the
eval ilorest$s info Attributes/ThermalConfig

# Change ThermalConfig
echo -e "\n\nChanging ThermalConfig"
eval ilorest$s set Attributes/ThermalConfig="IncreasedCooling"  --commit
```

ThermalConfig=OptimalCooling

NAME

ThermalConfig

DISPLAY NAME

Thermal Configuration

HELP TEXT

Use this option to select the fan cooling solution for the system. Optimal Cooling provides the most efficient solution by configuring fan speeds to the minimum required speed to provide adequate cooling. Increased Cooling runs fans at higher speeds to provide additional cooling. Select Increased Cooling when third-party storage controllers are cabled to the embedded hard drive cage, or if the system is experiencing thermal issues that cannot be resolved. Maximum cooling provides the maximum cooling available on this platform. Enhanced CPU Cooling runs the fans at a higher speed to provide additional cooling to the processors. Selecting Enhanced CPU Cooling may improve system performance with certain processor intensive workloads.

TYPE

Enumeration

READ-ONLY

False

POSSIBLE VALUES

- OptimalCooling
- IncreasedCooling
- MaxCooling
- EnhancedCPUCooling

Changing ThermalConfig

Added the following patch:

```
{  
  "Attributes/ThermalConfig": "IncreasedCooling"  
}
```

```
[14]: # Select type of server: Dl (ProLiant DL360g10) or Sy (Synergy)  
s="Dl"  
  
# Verify iLO firmware is equal or greater than 2.30  
eval ilorest$s get FirmwareVersion --select Manager.  
echo  
  
# Select data type  
eval ilorest$s select Thermal  
  
# Retrieve property  
eval ilorest$s get Oem/Hpe/ThermalConfiguration --json | jq -r '.[0].[]'  
echo  
  
# Retrieve possible values for property  
eval ilorest$s info Oem/Hpe/ThermalConfiguration  
echo  
  
# Modify property  
eval ilorest$s select Thermal --refresh  
eval ilorest$s set Oem/Hpe/ThermalConfiguration="MaximumCooling"
```

FirmwareVersion=iLO 5 v2.44

```
{  
  "ThermalConfiguration":  
    "OptimalCooling"  
}
```

Unable to locate registry model or No data available for entry:
Oem/Hpe/ThermalConfiguration

Entry Oem/Hpe/ThermalConfiguration not found in current selection
ERROR : There are no valid entries for info in the current instance.

Property is read-only skipping 'Oem'
ERROR : Nothing changed for attribute 'oem/hpe/thermalconfiguration'. Please
check if it is READONLY or the value trying to set is same or invalid



<- Environment Preparation ; List of examples ; Restart iLO 5 Simulator ; Logout ->

1.7 Managers

Examples in this section are related to the Manager data type as well as HPE Oem data types located below the /redfish/v1/Managers/ end point like the HpeILOSnmpService.

1.7.1 Retrieve iLO firmware version

The easiest way to retrieve an iLO firmware version is from the Oem.Hpe.ManagerFirmwareVersion property of the ServiceRoot data type at /redfish/v1. See the last loop in the environment preparation cell above. Note that a GET on this location does not require any authentication.

As per the [DMTF Schema documentation](#), the standard location for managers firmware is in the FirmwareVersion property of the Manager data type.

```
[15]: # Select type of server: Dl (ProLiant DL360g10) or Sy (Synergy)
s="Dl"

# Select data type
eval ilorest$s select Manager.

# Get current value
eval ilorest$s get FirmwareVersion
echo
```

FirmwareVersion=iLO 5 v2.44

1.7.2 Enable/Disable SNMPv1 in iLO

NOTE Changing the SNMPv1Enabled setting appeared in iLO 5 FW 2.30.

API Reference document keyword: "SNMPv1Enabled"

iLO GUI location: Management --> SNMP Settings --> SNMP Alerts --> SNMPv1

```
[16]: # Select type of server: Dl (ProLiant DL360g10) or Sy (Synergy)
s="Sy"

# Select data type
```

```
eval ilorest$$ select HpeiLOSnmpService

# Get current value
eval ilorest$$ get SNMPv1Enabled
echo

# Set new value
eval ilorest$$ set SNMPv1Enabled=False ##commit

#eval ilorest$$ status
```

SNMPv1Enabled=True

Added the following patch:

```
{
  "SNMPv1Enabled": false
}
```

<- Environment Preparation ; List of examples ; Restart iLO 5 Simulator ; Logout ->

1.8 HPE iLO Resource Directory

The HPE iLO Resource Directory is an HPE OEM extension that contains the exhaustive list of the data types present in a Redfish service. This directory can be used by Redfish clients to lookup resources faster. Without this extension, Redfish clients must crawl the entire Redfish tree, starting at the root entry point, to find a specific resource.

The iLOrest utility caches the HPE iLO Resource Directory locally during the login process. When you select a data type, iLOrest very quickly retrieves its location from the cached directory. Then, when you submit get or set requests against specific properties, iLOrest starts looking for them starting at the location found in the cached directory.

1.8.1 Introduction

The HpeiLOResourceDirectory data type contains an array called Instances. Each element of this array corresponds to a data type with associated descriptors: @odata.id and HttpMethods. The first descriptor is the entry point of the data type in the Redfish tree and the second lists the valid HTTP methods against the data type.

NOTE: The word “instance” is a synonym of “data type”. You will find it in DMTF [python programs](#) as well as [HPE programs](#).

The following cell retrieves the first data types (aka instances) of the current iLO 5.

```
[17]: # Select type of server: Dl (ProLiant DL360g10) or Sy (Synergy)
s="Dl"

# Select data type
eval ilorest$$ select HpeiLOResourceDirectory
```

```
# List the first elements of array Instances containing data type properties
eval ilorest$s list Instances --json | head -30
```

```
{
  "Instances": [
    {
      "@odata.id": "/redfish/v1/",
      "@odata.type": "#ServiceRoot.v1_5_1.ServiceRoot",
      "ETag": "W/\\"7F921C47\\\"",
      "HttpMethods": [
        "GET",
        "HEAD"
      ]
    },
    {
      "@odata.id": "/redfish/v1/Managers/1/",
      "@odata.type": "#Manager.v1_5_1.Manager",
      "ETag": "W/\\"EDF80840\\\"",
      "HttpMethods": [
        "GET",
        "HEAD",
        "POST",
        "PATCH"
      ]
    },
    {
      "@odata.id": "/redfish/v1/Managers/",
      "@odata.type": "#ManagerCollection.ManagerCollection",
      "ETag": "W/\\"AA6D42B0\\\"",
      "HttpMethods": [
        "GET",
        "HEAD"
      ]
    }
  ]
}
```

```
Exception ignored in: <_io.TextIOWrapper name='<stdout>' mode='w'
encoding='UTF-8'>
```

```
BrokenPipeError: [Errno 32] Broken pipe
```

<- Environment Preparation ; List of examples ; Restart iLO 5 Simulator ; Logout ->

1.9 Computer System

This section contains examples related to the ComputerSystem data type.

1.9.1 PostState and DiscoveryComplete

Use these state parameters to evaluate the current state of a system before modifying or viewing its resources. As explained in this [article](#), certain parameters cannot be modified if the computer is not in a specific state.

Moreover, some parameters cannot be viewed when the entire discovery of all subsystems is not complete. As an example, you will not be able to view or modify SmartArray and logical drive configurations if the SmartArrayDiscovery is not complete.

```
[18]: # Select type of server: Dl (ProLiant DL360g10) or Sy (Synergy)
s="Dl"

# Select data type
eval ilorest$$ select ComputerSystem

# Retrieve and print DeviceDiscoveryComplete and PostState properties
eval ilorest$$ get Oem/Hpe/DeviceDiscoveryComplete | grep -v -E 'Oem|Hpe'; echo
eval ilorest$$ get Oem/Hpe/PostState | grep -v -E 'Oem|Hpe' ; echo
```

```
DeviceDiscoveryComplete=
                                AMSDeviceDiscovery=Complete
                                DeviceDiscovery=vMainDeviceDiscoveryComplete
                                SmartArrayDiscovery=Complete

PostState=FinishedPost
```

<– Environment Preparation ; List of examples ; Restart iLO 5 Simulator ; Logout –>

1.9.2 AMS and HostOS

When the HPE Agentless Management Service (AMS) is installed in the operating system of the host, it is possible to get various information from the HostOS property. This property is present only when AMS is up and running in the host OS.

Use keywords agentless or ams in the [API reference documentation](#) to get the data type related to this topic.

```
[19]: # Select type of server: Dl (ProLiant DL360g10) or Sy (Synergy)
s="Dl"

# Print info on the AgentlessManagementService
echo "*****"
eval ilorest$$ info Oem/Hpe/AggregateHealthStatus/AgentlessManagementService
echo "*****"
echo -e "\n\n"

# Select type
eval ilorest$$ select ComputerSystem.

# Retrieve and print AMS status
AmsStatus=$(eval ilorest$$ get --json Oem/Hpe/AggregateHealthStatus/
→AgentlessManagementService | \
```

```
jq -r '.Oem.Hpe.AggregateHealthStatus.AgentlessManagementService')
echo "AMS Status on $s simulator: $AmsStatus" ; echo

# Retrieve and print HostOS info only if AMS is ready
if [ "$AmsStatus" == "Ready" ] ; then
    echo "Host OS properties:"
    eval ilorest$s get --json Oem/Hpe/HostOS | jq -r '.Oem.Hpe.HostOS'
fi
```

Unable to locate registry model or No data available for entry:

Oem/Hpe/AggregateHealthStatus/AgentlessManagementService

NAME

AgentlessManagementService

DESCRIPTION

This indicates if the Agentless Management Service is available or not.

TYPE

string

READ-ONLY

True

POSSIBLE VALUES

Unavailable

Ready

AMS Status on D1 simulator: Ready

Host OS properties:

```
{
  "OsName": "CentOS Linux",
  "OsSysDescription": "Linux centos82rf2
4.18.0-240.1.1.el8_3.x86_64 #1 SMP Thu Nov 19 17:20:08 UTC 2020
x86_64",
  "OsType": 51,
  "OsVersion": "8"
}
```

<- Environment Preparation ; List of examples ; Restart iLO 5 Simulator ; Logout ->

1.9.3 Bios/RBSU Login password

The following cell uses the setpassword macro command to set the RBSU password for the first time and then removes it.

NOTE: The setpassword command may return errors when launched against a DMTF iLO 5 simulator.

```
[20]: # Select type of server: Dl (ProLiant DL360g10) or Sy (Synergy)
s="Dl"

# Retrieve help syntax and examples
echo -e "setpassword help syntax:\n"
eval ilorest$s help setpassword

# Setting a password for the first time
echo -e "\nSetting a password for the first time"
eval ilorest$s setpassword --newpassword testnew --currentpassword None

# Removing the Bios/RBSU password
echo -e "\n\nRemoving the BIOS/RBSU password"
eval ilorest$s setpassword --newpassword None --currentpassword testnew

# No commit needed. No iLO nor system reset needed.
eval ilorest$s status
```

setpassword help syntax:

Sets the admin password and power-on password

```
setpassword --newpassword <NEW_PASSWORD> --currentpassword <OLD_PASSWORD>
[OPTIONS]
```

Setting the admin password with no previous password set.

example: setpassword --newpassword testnew --currentpassword None

Setting the admin password back to nothing.

example: `setpassword --newpassword None --currentpassword testnew`

Setting the power on password.

example: `setpassword --newpassword testnew --currentpassword None --poweron`

Note: if it is empty password, send None as above.

Setting a password for the first time

An invalid response body was returned: Expecting value: line 1 column 1 (char 0)No error message returned or unable to parse error response.

Removing the BIOS/RBSU password

An invalid response body was returned: Expecting value: line 1 column 1 (char 0)No error message returned or unable to parse error response.

Current changes found:

Chassis.v1_10_2(/redfish/v1/Chassis/1/)

AssetTag=New Chassis Asset Tag

<- Environment Preparation ; List of examples ; Restart iLO 5 Simulator ; Logout ->

1.9.4 ComputerSystem Asset Tag

The ComputerSystem Asset Tag is different from the Chassis Asset Tag.

```
[21]: # Select type of server: Dl (ProLiant DL360g10) or Sy (Synergy)
s="Dl"

# Select data type and clear cache
eval ilorest$s select ComputerSystem. --refresh

# Retrieve ComputerSystem Asset Tag
eval ilorest$s get AssetTag

# Change ComputerSystem Asset Tag
eval ilorest$s set AssetTag="New\ ComputerSystem\ Asset\ Tag" --commit
echo

# Verify change
#eval ilorest$s get AssetTag
```

AssetTag=""

Added the following patch:

```
{
  "AssetTag": "New ComputerSystem Asset Tag"
}
```

Committing changes...

ERROR: expected string or bytes-like object

<- Environment Preparation ; List of examples ; Restart iLO 5 Simulator ; Logout ->

1.10 Security

1.10.1 Security Dashboard

The following five examples retrieve different security dashboard properties.

Overall security dashboard status Use keyword `Security Dashboard` or `SecurityDashboard` in the [API reference document](#) to find the overall security dashboard data type.

```
[22]: # Select type of server: Dl (ProLiant DL360g10) or Sy (Synergy)
s="Dl"

# Select data type
eval ilorest$s select HpeiLOSecurityDashboard.

# Retrieve overall security dashboard status
eval ilorest$s get OverallSecurityStatus
```

OverallSecurityStatus=Risk

All Security parameters and properties in json format Use keyword `SecurityParam` to find the corresponding data type in the [API Reference Document](#).

```
[23]: # Select type of server: Dl (ProLiant DL360g10) or Sy (Synergy)
s="Dl"

# Select data type
eval ilorest$s select HpeiLOSecurityParam.

# List all security params in JSON format. Discard "@odata" metadata to get
  → cleaner output.
eval ilorest$s list --json | grep -v '@'
```

```
[
  {
    "Id": "2",
    "Ignore": false,
    "Name": "Minimum Password Length",
    "SecurityStatus": "Ok",
    "State": "Ok"
  },
  {
    "Id": "7",
    "Ignore": false,
```

```

    "Name": "Require Host Authentication",
    "SecurityStatus": "Ok",
    "State": "Disabled"
  },
  {
    "Id": "4",
    "Ignore": false,
    "Name": "Authentication Failure Logging",
    "SecurityStatus": "Ok",
    "State": "Enabled"
  },
  {
    "Id": "9",
    "Ignore": false,
    "Name": "Default SSL Certificate In Use",
    "SecurityStatus": "Ok",
    "State": "False"
  },
  {
    "Description": "The Require Login for iLO RBSU setting is disabled. This
configuration allows unauthenticated iLO access through the UEFI System
Utilities.",
    "Id": "3",
    "Ignore": false,
    "Name": "Require Login for iLO RBSU",
    "RecommendedAction": "Enable the Require Login for iLO RBSU setting.",
    "SecurityStatus": "Risk",
    "State": "Disabled"
  },
  {
    "Id": "8",
    "Ignore": false,
    "Name": "Last Firmware Scan Result",
    "SecurityStatus": "Ok",
    "State": "Ok"
  },
  {
    "Id": "1",
    "Ignore": false,
    "Name": "IPMI/DCMI Over LAN",
    "SecurityStatus": "Ok",
    "State": "Disabled"
  },
  {
    "Description": "The Password Complexity setting is disabled. This
configuration increases system vulnerability to attack.",
    "Id": "6",
    "Ignore": true,

```

```

    "IgnoredBy": "demopaq",
    "IgnoredTime": "2020-05-11T12:55:02Z",
    "Name": "Password Complexity",
    "SecurityStatus": "Risk",
    "State": "Disabled"
  },
  {
    "Description": "SNMPv1 is enabled. This configuration increases system
vulnerability to attack.",
    "Id": "10",
    "Ignore": false,
    "Name": "SNMPv1",
    "RecommendedAction": "Disable the SNMPv1 protocol.",
    "SecurityStatus": "Risk",
    "State": "Enabled"
  },
  {
    "Id": "0",
    "Ignore": false,
    "Name": "Security Override Switch",
    "SecurityStatus": "Ok",
    "State": "Off"
  },
  {
    "Description": "The UEFI Secure Boot setting is disabled. In this
configuration, the UEFI system firmware does not validate the boot loader,
Option ROM firmware, and other system software executables for trusted
signatures. This configuration breaks the chain of trust established by iLO from
power-on",
    "Id": "5",
    "Ignore": true,
    "IgnoredBy": "demopaq",
    "IgnoredTime": "2020-05-11T16:53:22Z",
    "Name": "Secure Boot",
    "SecurityStatus": "Risk",
    "State": "Disabled"
  }
]

```

List only security params at risk To retrieve this list you don't need to select the data type as it has already selected in the previous cell.

```

[24]: # Select type of server: Dl (ProLiant DL360g10) or Sy (Synergy)
s="Dl"

# Select data type
eval ilorest$s select HpeiLOSecurityParam.

```

```
eval ilorest$s list --filter SecurityStatus="Risk" | grep -v '@'
echo
```

Description=The UEFI Secure Boot setting is disabled. In this configuration, the UEFI system firmware does not validate the boot loader, Option ROM firmware, and other system software executables for trusted signatures. This configuration breaks the chain of trust established by iLO from power-on

Id=5

Ignore=True

IgnoredBy=demopaq

IgnoredTime=2020-05-11T16:53:22Z

Name=Secure Boot

SecurityStatus=Risk

State=Disabled

Description=The Password Complexity setting is disabled. This configuration increases system vulnerability to attack.

Id=6

Ignore=True

IgnoredBy=demopaq

IgnoredTime=2020-05-11T12:55:02Z

Name=Password Complexity

SecurityStatus=Risk

State=Disabled

Description=The Require Login for iLO RBSU setting is disabled. This configuration allows unauthenticated iLO access through the UEFI System Utilities.

Id=3

Ignore=False

Name=Require Login for iLO RBSU

RecommendedAction=Enable the Require Login for iLO RBSU setting.

SecurityStatus=Risk

State=Disabled

Description=SNMPv1 is enabled. This configuration increases system vulnerability to attack.

Id=10

Ignore=False

Name=SNMPv1

RecommendedAction=Disable the SNMPv1 protocol.

SecurityStatus=Risk

State=Enabled

Get specific security parameter properties

```
[25]: # Select type of server: Dl (ProLiant DL360g10) or Sy (Synergy)
s="Dl"

# Select data type
eval ilorest$s select HpeiLOSecurityParam.

eval ilorest$s get --filter Name="Require\ Login\ for\ iLO\ RBSU" | grep -v '@'
echo
```

Description=The Require Login for iLO RBSU setting is disabled. This configuration allows unauthenticated iLO access through the UEFI System Utilities.

Id=3

Ignore=False

Name=Require Login for iLO RBSU

RecommendedAction=Enable the Require Login for iLO RBSU setting.

SecurityStatus=Risk

State=Disabled

Get specific property of a specific security parameter

```
[26]: # Select type of server: Dl (ProLiant DL360g10) or Sy (Synergy)
s="Dl"

# Select data type
eval ilorest$s select HpeiLOSecurityParam.

# Retrieve the `Ignore` property of the `Minimum Password Length` security_
→parameter
eval ilorest$s get Ignore --filter Name="Minimum\ Password\ Length"
echo
```

Ignore=False

<- Environment Preparation ; List of examples ; Restart iLO 5 Simulator ; Logout ->

1.10.2 Security Mode/State

From the iLO 5 Graphical User Interface, in the Security -> Encryption menu, you can view and modify the SecurityState setting. This setting is part of the HpeSecurityService data type as mentioned in the [API Reference Document](#).

NOTE: Software like the Smart Update Tool (SUT) may refer to this setting as Security Mode instead of Security State.

Retrieve and modify the iLO 5 Security State setting The following cell retrieves all the properties of the HpeSecurityService and modifies the SecurityState. A similar change on a real iLO 5 automatically triggers a reset of the iLO.

```
[27]: # Select type of server: Dl (ProLiant DL360g10) or Sy (Synergy)
s="Dl"

# Select data type and refresh cache
eval ilorest$s select HpeSecurityService. --refresh

# Retrieve all Security Service parameters
eval ilorest$s get --json

# Modify Security State
SecurityState=$(eval ilorest$s get SecurityState | cut --delimiter== --field=2)
echo

if [ "$SecurityState" = "Production" ]; then
    echo "Setting Security State to HighSecurity"
    eval ilorest$s set SecurityState="HighSecurity" #--commit
else
    echo "Setting Security State to Production"
    eval ilorest$s set SecurityState="Production" #--commit
fi

echo
eval ilorest$s get SecurityState
```

```
{
  "@odata.context":
"/redfish/v1/$metadata#HpeSecurityService.HpeSecurityService",
  "@odata.etag": "W/\\"71C6AA88\\\"",
  "@odata.id": "/redfish/v1/Managers/1/SecurityService",
  "@odata.type": "#HpeSecurityService.v2_3_1.HpeSecurityService",
  "CurrentCipher": "ECDHE-RSA-AES256-GCM-SHA384",
  "Id": "SecurityService",
  "Links": {
    "CertAuth": {
      "@odata.id":
"/redfish/v1/Managers/1/SecurityService/CertificateAuthentication"
    },
    "ESKM": {
      "@odata.id": "/redfish/v1/Managers/1/SecurityService/ESKM"
    },
    "HttpsCert": {
      "@odata.id": "/redfish/v1/Managers/1/SecurityService/HttpsCert"
    },
    "SSO": {
```

```

    "@odata.id": "/redfish/v1/Managers/1/SecurityService/SSO"
  },
  "SecurityDashboard": {
    "@odata.id": "/redfish/v1/Managers/1/SecurityService/SecurityDashboard"
  }
},
"LoginSecurityBanner": {
  "IsEnabled": true,
  "SecurityMessage": "*Default HPE SecurityMessage: This is a private system.
It is to be used solely by authorized users and may be monitored for all lawful
purposes. By accessing this system, you are consenting to such monitoring.\n"
},
"PlatformCert": {
  "Certificates": {
    "@odata.id":
"/redfish/v1/Managers/1/SecurityService/PlatformCert/Certificates"
  }
},
"PlatformCertUpdate": {
  "Certificates": {
    "@odata.id":
"/redfish/v1/Managers/1/SecurityService/PlatformCertUpdate/Certificates"
  }
},
"SSHHostKey": "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDD578WZlbtL3xzm4TcxaHosm
OgeEn5kGhvlcAlTSpYNkYJY3WYYTkrWeLj+h4A1ofkZtoiOLUwYGvz7m0JjLgtwKLsEYRP7F50Izzva4
tpw03RPZ9AB9ijgbiVkuHCdnNC0cna3tsDsCmuDjo4yu5DCtL4TABR4eGEjppMH45SNl6wj8Pz8aD1Zy
tZUHu92EmRH4GiToxGWSIukEb/uw1l8PCuqj3iTzZScWhPdHrcCv+k34f0TtDyIqj8iEhjygtjtmc6k
N9M+ebpx2cETD5f53RySUF26tPqg81r0jh24K7Cuxwij1VcHKNDFM7TObEGMwwf1L1kPoVWn/ewFf",
  "SecurityState": "Production",
  "SecurityState@Redfish.AllowableValues": [
    "Production",
    "HighSecurity",
    "FIPS"
  ],
  "SystemIAK": {
    "Certificates": {
      "@odata.id":
"/redfish/v1/Managers/1/SecurityService/SystemIAK/Certificates"
    }
  },
  "SystemIDDevID": {
    "Certificates": {
      "@odata.id":
"/redfish/v1/Managers/1/SecurityService/SystemIDDevID/Certificates"
    }
  },
  "iLOIDevID": {

```

```

    "Certificates": {
      "@odata.id":
"/redfish/v1/Managers/1/SecurityService/iLOIDevID/Certificates"
    }
  },
  "iLODevID": {
    "Certificates": {
      "@odata.id":
"/redfish/v1/Managers/1/SecurityService/iLODevID/Certificates"
    }
  }
}

```

Setting Security State to HighSecurity

Added the following patch:

```

{
  "SecurityState": "HighSecurity"
}

```

SecurityState=HighSecurity

<– Environment Preparation ; List of examples ; Restart iLO 5 Simulator ; Logout –>

1.10.3 Login Security Banner

From the iLO 5 Graphical User Interface, in the Security -> Login Security Banner menu, you can enable/disable and customize the LoginSecurityBanner setting. This setting is part of the HpeSecurityService data type as mentioned in the [API Reference Document](#).

Retrieve and modify the iLO 5 Security State setting The following cell modifies the state of the Security Login Banner and provides a message when enabled. You don't need to reset the iLO 5 to effectively modify this banner.

```

[28]: # Select type of server: Dl (ProLiant DL360g10) or Sy (Synergy)
s="Sy"

# Select data type and refresh cache
eval ilorest$s select HpeSecurityService. --refresh

# Retrieve Login Security Banner settings
eval ilorest$s get LoginSecurityBanner --json

# Modify Login Security Banner settings
LoginSecurityBanner=$(eval ilorest$s get LoginSecurityBanner/IsEnabled | cut_
->--delimiter== --field=2)
echo

if [ $LoginSecurityBanner = False ]; then

```

```

echo "Enabling Login Security Banner with a custom message"
eval ilorest$$ set LoginSecurityBanner/IsEnabled=True ##--commit
eval ilorest$$ set LoginSecurityBanner/SecurityMessage="This is My Custom_
→Security Message" ##--commit
else
    echo "Disabling Login Security Banner"
    # The following set command is not required with a real iLO 5.
    # It's only there to get a "realistic" behavior of the simulator.
    eval ilorest$$ set LoginSecurityBanner/SecurityMessage="" ##--commit
    eval ilorest$$ set LoginSecurityBanner/IsEnabled=False ##--commit
fi

echo
eval ilorest$$ get --json LoginSecurityBanner

```

```

{
  "LoginSecurityBanner": {
    "IsEnabled": false
  }
}

```

Enabling Login Security Banner with a custom message

Added the following patch:

```

{
  "LoginSecurityBanner/IsEnabled": true
}

```

Skipping property securitymessage, not found in current server.

ERROR : Nothing changed for attribute 'loginsecuritybanner/securitymessage'.

Please check if it is READONLY or the value trying to set is same or invalid

```

{
  "LoginSecurityBanner": {
    "IsEnabled": true
  }
}

```

<- Environment Preparation ; List of examples ; Restart iLO 5 Simulator ; Logout ->

1.11 Bios

1.11.1 Server Serial Number stored in BIOS

In a computer node, the Serial Number stored in the Bios. data type corresponds to the server mother board serial number. HPE requires the change of this serial number in case of mother board replacement to match the chassis serial number.

```

[29] : # Select type of server: Dl (ProLiant DL360g10) or Sy (Synergy)
s="Dl"

```

```

# Select data type
eval ilorest$s select Bios. --refresh
echo

# Make sure SN is read-write
eval ilorest$s info Attributes/SerialNumber --latestschema
echo

# Get SerialNumber from Bios data type.
eval ilorest$s get Attributes/SerialNumber

echo "Set new SN"
eval ilorest$s set Attributes/SerialNumber="NEWSN" #--commit

# Reload a fresh simulator cache
eval ilorest$s select Bios. --refresh
echo

# Validate the SN change
eval ilorest$s get Attributes/SerialNumber
echo

```

NAME

SerialNumber

DISPLAY NAME

Serial Number

HELP TEXT

Use this option to set the system serial number. This value must always match the serial number sticker located on the chassis.

WARNING

The serial number is modified by qualified service personnel and must match the serial number located on the chassis.

```
TYPE
    String
```

```
READ-ONLY
    False
```

```
MIN LENGTH
    0
```

```
MAX LENGTH
    16
```

```
SerialNumber=CZJ93402YV
Set new SN
Added the following patch:
{
    "Attributes/SerialNumber": "NEWSN"
}
```

```
SerialNumber=CZJ93402YV
```

<- Environment Preparation ; List of examples ; Restart iLO 5 Simulator ; Logout ->

1.12 Storage

The following cell lists the storage related data types implemented in the current iLO 5 Redfish service.

```
[30]: # Select type of server: Dl (ProLiant DL360g10) or Sy (Synergy)
s="Sy"

eval ilorest$s types | grep -i -E 'storage|drive' | grep -v -i driver
```

```
Drive.v1_1_0
HpeSmartStorage.v2_0_0
HpeSmartStorageArrayController.v2_2_0
HpeSmartStorageArrayControllerCollection
HpeSmartStorageDiskDrive.v2_1_0
HpeSmartStorageDiskDriveCollection
HpeSmartStorageHostBusAdapterCollection
HpeSmartStorageLogicalDrive.v2_3_0
HpeSmartStorageLogicalDriveCollection
HpeSmartStorageStorageEnclosure.v2_0_0
HpeSmartStorageStorageEnclosureCollection
SmartStorageConfig.v2_0_2
```

```
Storage.v1_5_0
StorageCollection
```

1.12.1 SSD Wear Status

The `WearStatus` property is an HPE Oem property part of the Drive data type as mentioned in the [API Reference Document](#). Note that the Redfish standard proposes the `PredictedMediaLifeLeftPercent` property, also in the Drive type, and being somehow equivalent.

The Drive data type may be absent in servers not containing suitable hardware. The physical DL360 Gen10 server available in this infrastructure (and the associated simulator) does not contain any SSD drive with a `WearStatus`. Hence, you will have to use the Synergy iLO 5 simulator to view this property.

The Redfish service of the Synergy iLO 5 simulator contains two SSD drives. The next cell retrieves successively all their properties, the `PredictedMediaLifeLeftPercent` standard property and then the Oem HPE `WearStatus` property.

```
[31]: # Select type of server: DL (ProLiant DL360g10) or Sy (Synergy)
s="Sy"

echo -e "Drive list. Physical and Logical Hpe SmartStorage excluded"
eval ilorest$s list --json --select Drive. --filter "@odata.type"="#Drive.v1_1_0.
→Drive"

echo -e "\n\n'PredictedMediaLifeLeftPercent' standard property of drives. Odata_
→metadata discarded."
eval ilorest$s get --json Id PredictedMediaLifeLeftPercent --filter "@odata.
→type"="#Drive.v1_1_0.Drive"

echo -e "\n\nOem/Hpe specific 'WearStatus' of non-SmartStorage"
eval ilorest$s get Id Oem/Hpe/WearStatus --json --filter "@odata.type"="#Drive.
→v1_1_0.Drive"
```

```
Drive list. Physical and Logical Hpe SmartStorage excluded
[
  {
    "@odata.context": "/redfish/v1/$metadata#Drive.Drive",
    "@odata.etag": "W/\"50B86A00\"",
    "@odata.id": "/redfish/v1/Systems/1/Storage/DA000001/Drives/DA000001/",
    "@odata.type": "#Drive.v1_1_0.Drive",
    "Actions": {
      "#Drive.Reset": {
        "target":
"/redfish/v1/Systems/1/Storage/DA000001/Drives/DA000001/Actions/Drive.Reset/"
      }
    }
  }
]
```



```

    },
    "CapacityBytes": 3200631000000,
    "FailurePredicted": false,
    "Id": "DA000001",
    "Identifiers": {
        "DurableName": "5CD2E447932E0100",
        "DurableNameFormat": "EUI"
    },
    "IndicatorLED": "Off",
    "Location": [
        {
            "Info": "1:2",
            "InfoFormat": "Box:Bay"
        }
    ],
    "MediaType": "SSD",
    "Model": "M0003200KWVUV",
    "Name": "Secondary Storage Device",
    "Oem": {
        "Hpe": {
            "@odata.context": "/redfish/v1/$metadata#HpeILODriveExt.HpeILODriveExt",
            "@odata.type": "#HpeILODriveExt.v2_0_1.HpeILODriveExt",
            "CurrentTemperatureCelsius": 34,
            "DriveStatus": {
                "Health": "OK",
                "State": "Enabled"
            },
            "HealthUpdated": "Dynamic",
            "NVMeId": "8086_M0003200KWVUV_PHLN0070013A3P2BGN",
            "PowerOnHours": 6120,
            "TemperatureStatus": {
                "Health": "OK",
                "State": "Enabled"
            },
            "WearStatus": "OK"
        }
    },
    "PredictedMediaLifeLeftPercent": 100,
    "Protocol": "NVMe",
    "Revision": "4ICRHPK0",
    "SerialNumber": "PHLN0070013A3P2BGN",
    "Status": {
        "Health": "OK"
    }
},
{
    "@odata.context": "/redfish/v1/$metadata#Drive.Drive",
    "@odata.etag": "W/\"50B86A00\"",

```

```

"@odata.id": "/redfish/v1/Systems/1/Storage/DA000000/Drives/DA000000/",
"@odata.type": "#Drive.v1_1_0.Drive",
"Actions": {
  "#Drive.Reset": {
    "target":
"/redfish/v1/Systems/1/Storage/DA000000/Drives/DA000000/Actions/Drive.Reset/"
  }
},
"CapacityBytes": 3200631000000,
"FailurePredicted": false,
"Id": "DA000000",
"Identifiers": {
  "DurableName": "5CD2E454572C0100",
  "DurableNameFormat": "EUI"
},
"IndicatorLED": "Off",
"Location": [
  {
    "Info": "1:1",
    "InfoFormat": "Box:Bay"
  }
],
"MediaType": "SSD",
"Model": "M0003200KWVUV",
"Name": "Secondary Storage Device",
"Oem": {
  "Hpe": {
    "@odata.context": "/redfish/v1/$metadata#HpeILODriveExt.HpeILODriveExt",
    "@odata.type": "#HpeILODriveExt.v2_0_1.HpeILODriveExt",
    "CurrentTemperatureCelsius": 36,
    "DriveStatus": {
      "Health": "OK",
      "State": "Enabled"
    },
    "HealthUpdated": "Dynamic",
    "NVMeId": "8086_M0003200KWVUV_PHLN009201P23P2BGN",
    "PowerOnHours": 6210,
    "TemperatureStatus": {
      "Health": "OK",
      "State": "Enabled"
    },
    "WearStatus": "OK"
  }
},
"PredictedMediaLifeLeftPercent": 100,
"Protocol": "NVMe",
"Revision": "4ICRHPKO",
"SerialNumber": "PHLN009201P23P2BGN",

```

```

    "Status": {
      "Health": "OK"
    }
  }
]

```

'PredictedMediaLifeLeftPercent' standard property of drives. Odata metadata discarded.

```

[
  {
    "Id": "DA000001",
    "PredictedMediaLifeLeftPercent": 100
  },
  {
    "Id": "DA000000",
    "PredictedMediaLifeLeftPercent": 100
  }
]

```

Oem/Hpe specific 'WearStatus' of non-SmartStorage

```

[
  {
    "Id": "DA000001",
    "Oem": {
      "Hpe": {
        "WearStatus": "OK"
      }
    }
  },
  {
    "Id": "DA000000",
    "Oem": {
      "Hpe": {
        "WearStatus": "OK"
      }
    }
  }
]

```

<- Environment Preparation ; List of examples ; Restart iLO 5 Simulator ; Logout ->

1.13 Logout

Always logout when your session is over. You can as well logout and restart the iLO 5 simulator if you want once again use this Jupyter Notebook.

```
[32]: ilorestDl logout  
      ilorestSy logout
```

Logging session out.

Logging session out.

<- Environment Preparation ; List of examples ; Restart iLO 5 Simulator ; Logout ->

1.14 Summary

In this workshop, you discovered several HPE eval iLOrest\$s examples. Go to the [Conclusion Notebook](#) if you are finished.