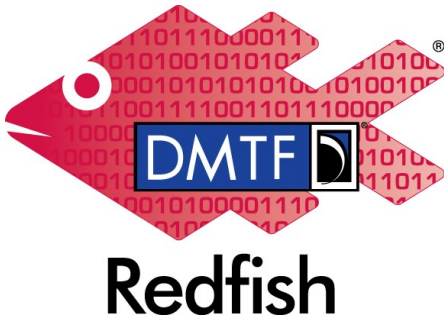

Version 0.17

OData operations

Introduction

This Jupyter notebook demonstrates the `$expand` and `only` Odata queries using Bash (https://www.gnu.org/software/bash/) and the cURL (https://curl.haxx.se/) tool against an HPE iLO 5. For didactic reasons, commands presented in this notebook may be not optimized and don't follow the recommended best practises (https://developer.hpe.com/blog/getting-started-with-the-redfish-api-part-2).

OData (https://www.odata.org/):

An open protocol to allow the creation and consumption of queryable and interoperable RESTful APIs in a simple and standard way.

More details are in the HPE Redfish API Reference document (https://hewlettpackard.github.io/ilo-rest-api-docs/ilo5/#introduction).

`$expand` query parameter

iLO 5 $expand Support

Using the rules above, iLO 5 supports $expand in this way:

`$expand` is applicable to HTTP GET only.

`$expand=.`, `$expand=*`, `$expand=($levels=n)` result in the same behavior: * Expands all links in both root and Oem/Hpe sections not inside the `Links` sections. * Levels is always interpreted as 1, regardless of n. This is to avoid the potential for expanding recursively for interlinked resources. * The `Links` section is never expanded. This is to avoid expanding the Chassis and Manager related links on GET operations to System.

NOTES: * The root resource at `/redfish/v1/` is available without authentication and has navigational links that can be expanded. An $expand request does not result in expansion unless valid authentication credentials are supplied. * There might be other links that do not support expand.

`only` Query parameter

> **iLO 5 “only” Query Parameter**
>
> iLO 5 1.40 and later supports the `only` query parameter documented in the Redfish API specification. This query parameter is ignored except on collections with only one member. Examples include the `ComputerSystemCollection`, `ChassisCollection`, and `ManagerCollection`.

Create environment variables and session

The following `bash` code defines environment variables (i.e. IP address, username, password....) depending on your student ID number stored in variable `$Stud`. It creates as well several `.json` files containing various HTTP workloads required to POST or PATCH the managed iLO.

In [1]:
```
# Create BMC related variables
iLO5_IP=172.16.50.99
iLO5_URI="https://${iLO5_IP}"
RemoteHost_IP=172.16.50.100

# iLO 5 Administrator credentials
iLO5_User="student"
iLO5_Passwd='P@ssw0rd!'

# EventReceiver
#EventReceiverIP=192.168.0.99
EventReceiverIP=balt

# Minimum required Redfish headers
HeaderODataVersion="OData-Version: 4.0"
HeaderContentType="Content-Type: application/json"

# Data files
ResponseHeaders="ResponseHeaders.txt"              # Used to hold HTTP response head
ers
SessionData="./CreateSession-data.json"            # Body/Workload used to create th
e Redfish session

cat > ${SessionData} << __EOF__
{
        "UserName": "$iLO5_User",
        "Password": "$iLO5_Passwd"
}
__EOF__
```

Create the Redfish session

Redfish allows basic authentication and session authentication. With basic authentication you need to supply the required credentials at each and every HTTP request. Session oriented authentication is achieved by requesting a `Token` that will be sent in the headers of all requests until the removal of the session.

To get this `Token` , POST a session request with the remote BMC credentials in its body. The `Token` as well as the `session location` will be in the headers of the response.

In [2]:
```
echo 'Create iLO 5 Session'

curl --dump-header  $ResponseHeaders \
     --insecure --noproxy "localhost, 127.0.0.1" --silent \
     --header "$HeaderContentType" --header "$HeaderODataVersion" \
     --request POST --data "@$SessionData" \
     ${iLO5_URI}/redfish/v1/SessionService/Sessions > /dev/null 2>&1

Token=$(awk '/X-Auth-Token/ {print $NF}' $ResponseHeaders | tr -d '\r')
SessionLocation="$iLO5_URI"$(awk '/^Loca.*Se/ {gsub("https://.*/red", "/red", $NF);
print $NF}' $ResponseHeaders | tr -d '\r')

echo "Token: $Token"
echo -e "Session Location: $SessionLocation\n"
```

```
Create iLO 5 Session
Token: 23f1292c002ebef68e5b7976d0a44858
Session Location: https://172.16.50.99/redfish/v1/SessionService/Sessions/studen
t000000005e661416e72b020c
```

Retrieve iLO 5 supported OData protocol features

The following command retrieves the OData supported protocol features.

In [3]:
```
echo "Retrieve supported OData features:"
curl --insecure --silent --noproxy "localhost, 127.0.0.1"  \
     --header  "$HeaderContentType" --header "$HeaderODataVersion" \
     --header "X-Auth-Token: $Token" \
     --request GET ${iLO5_URI}/redfish/v1 | jq '.ProtocolFeaturesSupported'
```

```
Retrieve supported OData features:
{
  "ExpandQuery": {
    "ExpandAll": false,
    "Levels": true,
    "Links": false,
    "MaxLevels": 1,
    "NoLinks": true
  },
  "FilterQuery": true,
  "OnlyMemberQuery": true,
  "SelectQuery": false
}
```

GET collection without `$expand`

In [4]:

```
echo "GET FirmwareInventory collection"

curl --insecure --noproxy "localhost, 127.0.0.1" --silent \
     --header "$HeaderContentType" --header "$HeaderODataVersion" \
     --header "X-Auth-Token: $Token" \
     --request GET ${iLO5_URI}/redfish/v1/UpdateService/FirmwareInventory | jq
```

GET FirmwareInventory collection


```
{
  "@odata.context": "/redfish/v1/$metadata#SoftwareInventoryCollection.SoftwareI
nventoryCollection",
  "@odata.etag": "W/\"D821FF8B\"",
  "@odata.id": "/redfish/v1/UpdateService/FirmwareInventory",
  "@odata.type": "#SoftwareInventoryCollection.SoftwareInventoryCollection",
  "Description": "Firmware Inventory Collection",
  "Name": "Firmware Inventory Collection",
  "Members": [
    {
      "@odata.id": "/redfish/v1/UpdateService/FirmwareInventory/1"
    },
    {
      "@odata.id": "/redfish/v1/UpdateService/FirmwareInventory/2"
    },
    {
      "@odata.id": "/redfish/v1/UpdateService/FirmwareInventory/3"
    },
    {
      "@odata.id": "/redfish/v1/UpdateService/FirmwareInventory/4"
    },
    {
      "@odata.id": "/redfish/v1/UpdateService/FirmwareInventory/5"
    },
    {
      "@odata.id": "/redfish/v1/UpdateService/FirmwareInventory/6"
    },
    {
      "@odata.id": "/redfish/v1/UpdateService/FirmwareInventory/7"
    },
    {
      "@odata.id": "/redfish/v1/UpdateService/FirmwareInventory/8"
    },
    {
      "@odata.id": "/redfish/v1/UpdateService/FirmwareInventory/9"
    },
    {
      "@odata.id": "/redfish/v1/UpdateService/FirmwareInventory/10"
    },
    {
      "@odata.id": "/redfish/v1/UpdateService/FirmwareInventory/11"
    },
    {
      "@odata.id": "/redfish/v1/UpdateService/FirmwareInventory/12"
    },
    {
      "@odata.id": "/redfish/v1/UpdateService/FirmwareInventory/13"
    },
    {
      "@odata.id": "/redfish/v1/UpdateService/FirmwareInventory/14"
    },
    {
      "@odata.id": "/redfish/v1/UpdateService/FirmwareInventory/15"
    },
    {
      "@odata.id": "/redfish/v1/UpdateService/FirmwareInventory/16"
    },
    {
      "@odata.id": "/redfish/v1/UpdateService/FirmwareInventory/17"
    },
    {
      "@odata.id": "/redfish/v1/UpdateService/FirmwareInventory/18"
    },
```

Expand collection with `$expand`

In [5]:

```
echo "Retrieve ALL Firmware versions with a single GET"

curl --insecure --noproxy "localhost, 127.0.0.1" --silent \
     --header "$HeaderContentType" --header "$HeaderODataVersion" \
     --header "X-Auth-Token: $Token" \
     --request GET ${iLO5_URI}'/redfish/v1/UpdateService/FirmwareInventory?$expand
=.' | \
       jq '.Members | .[] | {Name: .Name, Description: .Description, Version: .Vers
ion}'
```

Retrieve ALL Firmware versions with a single GET


```
{
  "Name": "iLO 5",
  "Description": "SystemBMC",
  "Version": "2.10 Oct 30 2019"
}
{
  "Name": "System ROM",
  "Description": "SystemRomActive",
  "Version": "U32 v2.22 (11/13/2019)"
}
{
  "Name": "Intelligent Platform Abstraction Data",
  "Description": "PlatformDefinitionTable",
  "Version": "9.8.0 Build 15"
}
{
  "Name": "System Programmable Logic Device",
  "Description": "SystemProgrammableLogicDevice",
  "Version": "0x2A"
}
{
  "Name": "Power Management Controller Firmware",
  "Description": "PowerManagementController",
  "Version": "1.0.7"
}
{
  "Name": "NVMe Backplane Firmware",
  "Description": "NVMeBackplane",
  "Version": "1.20"
}
{
  "Name": "Power Supply Firmware",
  "Description": "PowerSupplies",
  "Version": "1.00"
}
{
  "Name": "Innovation Engine (IE) Firmware",
  "Description": "InnovationEngineFirmware",
  "Version": "0.2.1.2"
}
{
  "Name": "Server Platform Services (SPS) Firmware",
  "Description": "SPSFirmwareVersionData",
  "Version": "4.1.4.339"
}
{
  "Name": "Smart Storage Energy Pack",
  "Description": "SmartStorageEnergyPack",
  "Version": "2.1"
}
{
  "Name": "Redundant System ROM",
  "Description": "SystemRomBackup",
  "Version": "U32 v2.22 (11/13/2019)"
}
{
  "Name": "Intelligent Provisioning",
  "Description": "Intelligent Provisioning",
  "Version": "3.30.213"
}
{
  "Name": "Power Management Controller FW Bootloader",
  "Description": "PowerManagementControllerBootloader",
```

Expand collection only if it contains one element

Without the `only` query parameter

In [6]:
```
echo "Chassis collection"

curl --insecure --noproxy "localhost, 127.0.0.1" --silent \
     --header "$HeaderContentType" --header "$HeaderODataVersion" \
     --header "X-Auth-Token: $Token" \
     --request GET ${iLO5_URI}'/redfish/v1/Chassis' | jq
```

```
Chassis collection
{
  "@odata.context": "/redfish/v1/$metadata#ChassisCollection.ChassisCollection",
  "@odata.etag": "W/\"AA6D42B0\"",
  "@odata.id": "/redfish/v1/Chassis",
  "@odata.type": "#ChassisCollection.ChassisCollection",
  "Description": "Computer System Chassis View",
  "Name": "Computer System Chassis",
  "Members": [
    {
      "@odata.id": "/redfish/v1/Chassis/1"
    }
  ],
  "Members@odata.count": 1
}
```

With the `only` query parameter

In [7]:

```
echo "Chassis collection"

curl --insecure --noproxy "localhost, 127.0.0.1" --silent \
     --header "$HeaderContentType" --header "$HeaderODataVersion" \
     --header "X-Auth-Token: $Token" \
     --request GET ${iLO5_URI}'/redfish/v1/Chassis?only' | jq
```

Chassis collection


```
{
  "@odata.context": "/redfish/v1/$metadata#Chassis.Chassis",
  "@odata.etag": "W/\"0942D38B\"",
  "@odata.id": "/redfish/v1/Chassis/1",
  "@odata.type": "#Chassis.v1_6_0.Chassis",
  "Id": "1",
  "AssetTag": "",
  "ChassisType": "RackMount",
  "IndicatorLED": "Off",
  "Links": {
    "ManagedBy": [
      {
        "@odata.id": "/redfish/v1/Managers/1"
      }
    ],
    "ComputerSystems": [
      {
        "@odata.id": "/redfish/v1/Systems/1"
      }
    ]
  },
  "Manufacturer": "HPE",
  "Model": "ProLiant DL360 Gen10",
  "Name": "Computer System Chassis",
  "NetworkAdapters": {
    "@odata.id": "/redfish/v1/Chassis/1/NetworkAdapters"
  },
  "Oem": {
    "Hpe": {
      "@odata.context": "/redfish/v1/$metadata#HpeServerChassis.HpeServerChassi
s",
      "@odata.type": "#HpeServerChassis.v2_3_1.HpeServerChassis",
      "Actions": {
        "#HpeServerChassis.DisableMCTPOnServer": {
          "target": "/redfish/v1/Chassis/1/Actions/Oem/Hpe/HpeServerChassis.Disa
bleMCTPOnServer"
        },
        "#HpeServerChassis.FactoryResetMCTP": {
          "target": "/redfish/v1/Chassis/1/Actions/Oem/Hpe/HpeServerChassis.Fact
oryResetMCTP"
        }
      },
      "Firmware": {
        "PlatformDefinitionTable": {
          "Current": {
            "VersionString": "9.8.0 Build 15"
          }
        },
        "PowerManagementController": {
          "Current": {
            "VersionString": "1.0.7"
          }
        },
        "PowerManagementControllerBootloader": {
          "Current": {
            "Family": "25",
            "VersionString": "1.1"
          }
        },
        "SPSFirmwareVersionData": {
          "Current": {
            "VersionString": "4.1.4.339"
          }
```

Delete sessions

It is extremely important to delete Redfish sessions to avoid reaching the maximum number of opened sessions in a BMC, preventing any access to it. Read this article (https://developer.hpe.com/blog/managing-ilo-sessions-with-redfish) for more detail.

In [8]:
```
echo "Body response of a session deletion:"

curl --insecure --noproxy "localhost, 127.0.0.1" --silent \
     --header "$HeaderContentType" --header "$HeaderODataVersion" \
     --header "X-Auth-Token: $Token" \
     --request DELETE $SessionLocation | jq
```

```
Body response of a session deletion:
{
  "error": {
    "code": "iLO.0.10.ExtendedInfo",
    "message": "See @Message.ExtendedInfo for more information.",
    "@Message.ExtendedInfo": [
      {
        "MessageId": "Base.1.4.Success"
      }
    ]
  }
}
```

Wrap up

In this notebook you performed the following actions:

- Create and delete a todken based Redfish session
- Retrieved iLO 5 supported OData protocol features
- Expand a collection with OData `$expand` query operator
- Expand a collection with OData `only` query operator