DELL Technologies

Technical Whitepaper

Simulate iDRAC Redfish Environment Using a Redfish Mockup Server

Abstract

This technical whitepaper describes the steps to leverage a Redfish mockup server on GitHub that enables the user to simulate an Integrated Dell Remote Access Controller (iDRAC) Redfish environment without the requirement of a physical server or iDRAC.

August 2024

Revisions

Date	Description
February 2024	Initial release
August 2024	Added multiple setups note updates.

Acknowledgments

Author:

- Texas Roemer—Software Quality Senior Principal Engineer

Support:

- Nikita Joshi—Software Engineer 2
- Rich Schnur—Consultant Product Management
- Ajay Shenoy—Software Senior Principal Engineer

Other:

- Roselin Louis—Technical Content Developer, Content Engineering and Translation (CE&T)

Contents

Executive Summary

A Redfish mockup server simulates the Redfish iDRAC environment without the requirement of a physical server or iDRAC.

Leverage the Redfish mockup to resolve issues in the following scenarios:

- When you cannot access to a physical server or iDRAC.
- When you cannot access to a server and want to evaluate by Redfish PATCH, or POST calls without being destructive.
- When you do not have iDRAC user privileges to perform Redfish calls.
- To train team members on Redfish without the requirement of physical servers of iDRAC.

1 iDRAC Redfish mockup client

Before you start the Redfish mockup server, you require a Redfish mockup client from iDRAC. You can either leverage the mockup client from iDRAC Redfish GitHub site, or you can create your own.

1.1 Leverage iDRAC mockup example from iDRAC GitHub repository

Multiple iDRAC Redfish mockup clients that are available on the iDRAC Redfish GitHub repository are available at: iDRAC-Redfish-Scripting. Extract the mockup file to expose the content which is leveraged by the Redfish mockup server to run.

Note—When you use Windows OS native file archiver to extract the content, the following message is displayed: `Destination Path Too Long`. To resolve this issue, do either of the following:
- You may either skip these errors to complete the extraction process.
- It is recommended to use an open-source file archiver (Example: 7-zip) to avoid this issue.

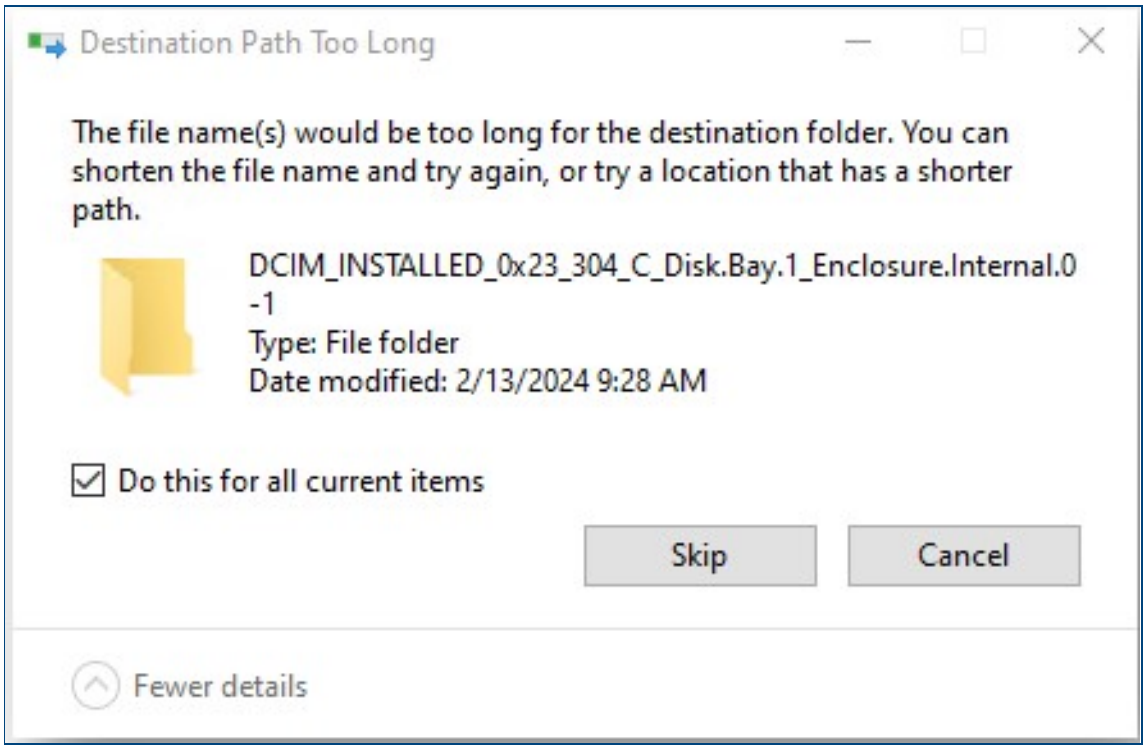

Figure 1 Windows error example

1.2 Create an iDRAC mockup client

When you create your own iDRAC mockup client, you must use Redfish Mockup Creator that is available at Redfish-Mockup-Creator. To run the Redfish Mockup Creator Python script, you must have access to iDRAC (iDRAC IP and user credentials).

The following is an example to run Redfish Mockup Creator Python script at the Windows command line:

```
Python redfishMockupCreate.py -u root -p calvin -r 192.168.0.120 -D
C:\Python310\iDRAC_Redfish_mockup_client -S
```

The following is an example to run Redfish Mockup Creator Python script at a Linux terminal:

```
[root@localhost opt]# python3 redfishMockupCreate.py -u root -p calvin -r
192.168.0.120 -D /junk/iDRAC_Redfish_mockup_client_R650 -S
```

Note—To run the Redfish mockup client, you must require requests and redfish modules. By default, these modules are not installed in Python but can be installed using pip3.

Note—To complete running the script, it may consume 5 to 10 minutes based on the server configuration content and Lifecycle Controller log data size.

To create the Redfish mockup client, run the following Python sample script in Windows OS:

```
C:\Python310>python redfishMockupCreate.py -u root -p calvin -r 192.168.0.120 -D
C:\Python310\iDRAC_mockup_client -S
Redfish Mockup Creator, Version 1.2.0
Address: https://192.168.0.120
Full Output Path: C:\Python310\iDRAC_mockup_client
Description:
Starting mockup creation...
Getting /redfish...
Getting /redfish/v1/odata...
Getting /redfish/v1/$metadata...
Getting /redfish/v1/Schemas/AccelerationFunction_v1.xml...
Getting /redfish/v1/Schemas/RedfishExtensions_v1.xml...
Getting /redfish/v1/Schemas/Settings_v1.xml...
Getting /redfish/v1/Schemas/Resource_v1.xml...
Getting /redfish/v1/Schemas/LogEntry_v1.xml...
Getting /redfish/v1/Schemas/Event_v1.xml...
Getting /redfish/v1/Schemas/Message_v1.xml...
Getting /redfish/v1/Schemas/CollectionCapabilities_v1.xml...
Getting /redfish/v1/Schemas/Certificate_v1.xml...
Getting /redfish/v1/Schemas/ResourceBlock_v1.xml...
Getting /redfish/v1/Schemas/Chassis_v1.xml...
Getting /redfish/v1/Schemas/Thermal_v1.xml...
Getting /redfish/v1/Schemas/PhysicalContext_v1.xml...
Getting /redfish/v1/Schemas/Redundancy_v1.xml...
Getting /redfish/v1/Schemas/Assembly_v1.xml...
```

```
Getting /redfish/v1/Schemas/Power_v1.xml...
Getting /redfish/v1/Schemas/Manager_v1.xml...
Getting /redfish/v1/Schemas/ComputerSystem_v1.xml...
Getting /redfish/v1/Schemas/LogServiceCollection_v1.xml...
Getting /redfish/v1/Schemas/LogService_v1.xml...
Getting /redfish/v1/Schemas/LogEntryCollection_v1.xml...
Getting /redfish/v1/Schemas/EthernetInterfaceCollection_v1.xml...
Getting /redfish/v1/Schemas/EthernetInterface_v1.xml...
Getting /redfish/v1/Schemas/IPAddresses_v1.xml...
Getting /redfish/v1/Schemas/VLanNetworkInterface_v1.xml...
Getting /redfish/v1/Schemas/VLanNetworkInterfaceCollection_v1.xml...
Getting /redfish/v1/Schemas/Endpoint_v1.xml...
Getting /redfish/v1/Schemas/Port_v1.xml...
Getting /redfish/v1/Schemas/Switch_v1.xml...
Getting /redfish/v1/Schemas/PortCollection_v1.xml...
Getting /redfish/v1/Schemas/Protocol_v1.xml...
Getting /redfish/v1/Schemas/PCIeDevice_v1.xml...
Getting /redfish/v1/Schemas/PCIeFunction_v1.xml...
```

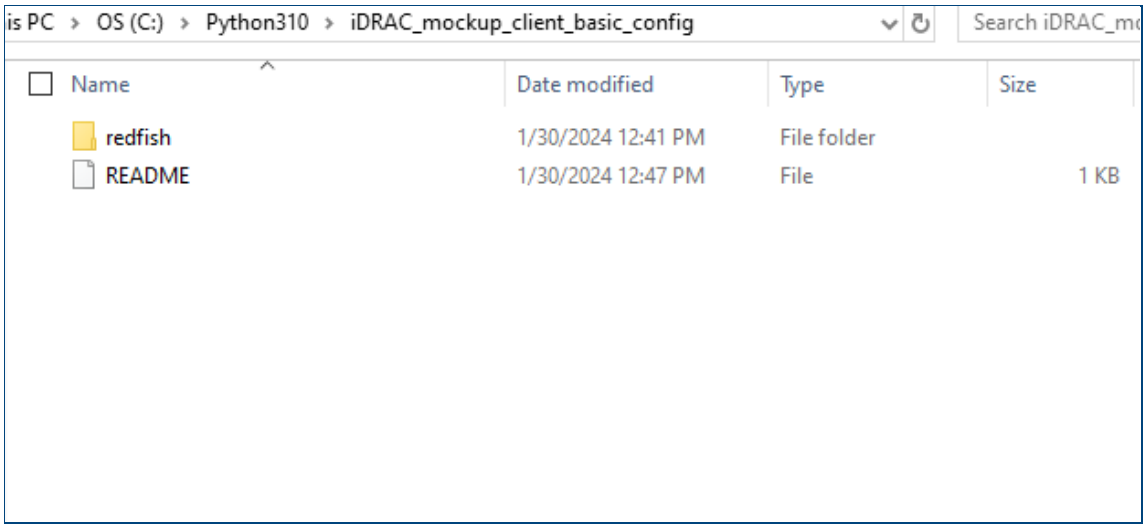

Figure 2 An example of Redfish mockup client content

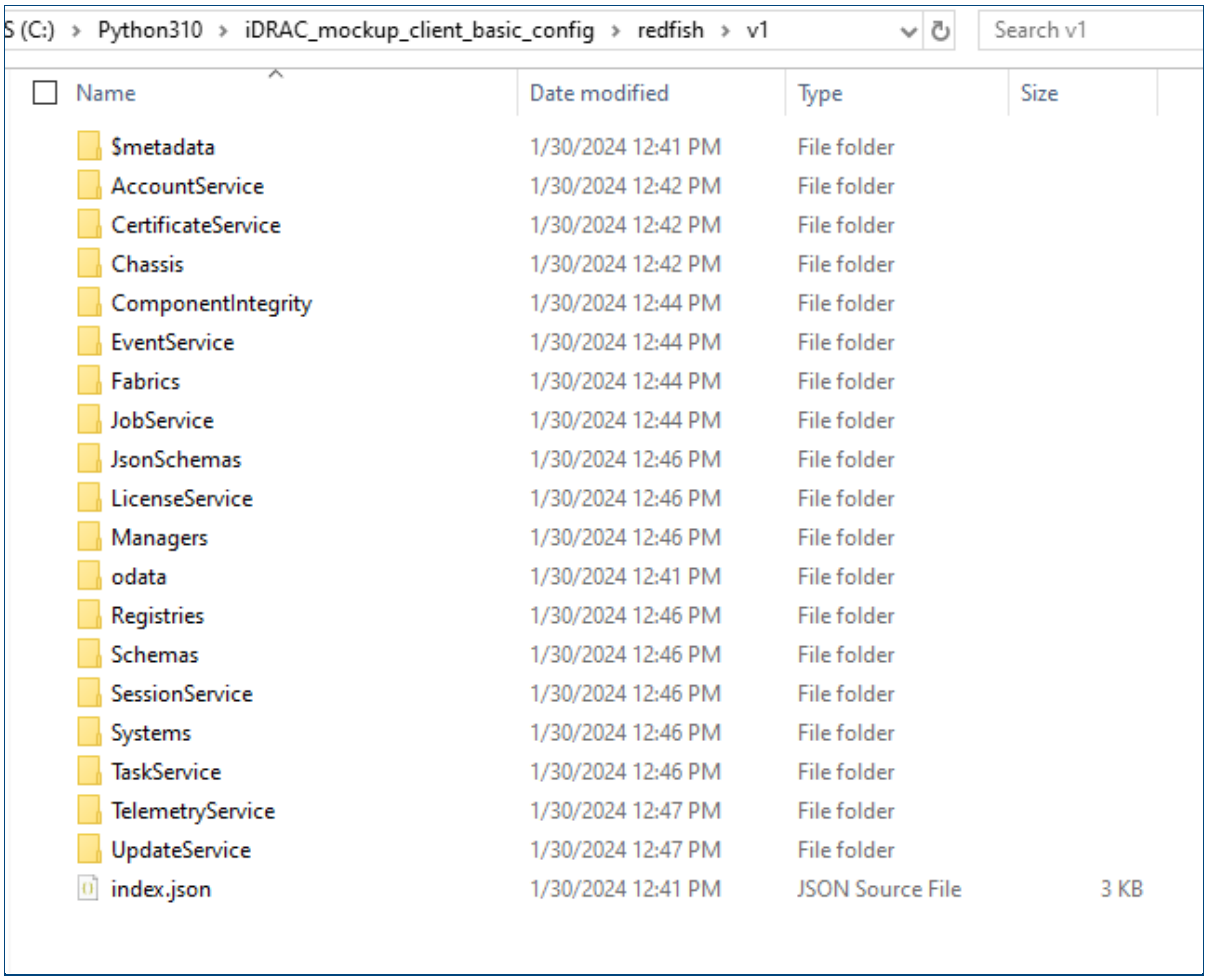

Figure 3 An example of Redfish mockup client content

To create the Redfish mockup client, run the following Python sample script in Linux OS:

```
[root@localhost opt]# python3 redfishMockupCreate.py -u root -p calvin -r
192.168.0.120 -D /junk/iDRAC_Redfish_mockup_client_R650 -S
Redfish Mockup Creator, Version 1.2.0
Address: https://192.168.0.120
Full Output Path: /junk/iDRAC_Redfish_mockup_client_R650
Description:
Starting mockup creation...
Getting /redfish...
Getting /redfish/v1/odata...
Getting /redfish/v1/$metadata...
Getting /redfish/v1/Schemas/AccelerationFunction_v1.xml...
Getting /redfish/v1/Schemas/RedfishExtensions_v1.xml...
Getting /redfish/v1/Schemas/Settings_v1.xml...
Getting /redfish/v1/Schemas/Resource_v1.xml...
Getting /redfish/v1/Schemas/LogEntry_v1.xml...
Getting /redfish/v1/Schemas/Event_v1.xml...
Getting /redfish/v1/Schemas/ResolutionStep_v1.xml...
```

```
Getting /redfish/v1/Schemas/ActionInfo_v1.xml...
Getting /redfish/v1/Schemas/Message_v1.xml...
Getting /redfish/v1/Schemas/CollectionCapabilities_v1.xml...
Getting /redfish/v1/Schemas/Certificate_v1.xml...
Getting /redfish/v1/Schemas/ResourceBlock_v1.xml...
Getting /redfish/v1/Schemas/Chassis_v1.xml...
Getting /redfish/v1/Schemas/Thermal_v1.xml...
Getting /redfish/v1/Schemas/PhysicalContext_v1.xml...
Getting /redfish/v1/Schemas/Redundancy_v1.xml...
Getting /redfish/v1/Schemas/Assembly_v1.xml...
```

2 Redfish mockup server

The Redfish mockup server serves Redfish requests against a Redfish mockup client. After you initiate the Redfish mockup server, you can run Redfish calls that simulates real time behavior.

Note—Redfish mockup server runs at either a specified IP address, port, the default localhost IP address, or 127.0.0.1:8000.

2.1 Start a Redfish mockup server

Before you start the Redfish simulation calls, you must start the Redfish mockup server leveraging the Python script at Redfish-Mockup-Server.

Note—To run the Redfish mockup server, you must require grequests and multipart modules. By default, these modules are not installed in Python but can be installed using pip3.

Note—Ensure to also download the `rfSsdpServer.py` script from GitHub and copy to your Python directory.

The following is an example to start the Redfish mockup server in Windows OS that points to the Redfish mockup client directory:

```
C:\Python310>python redfishMockupServer.py -D C:\Python310\iDRAC_mockup_client -
X
Redfish Mockup Server, version 1.2.4
Hostname: 127.0.0.1
Port: 8000
Mockup directory path specified: C:\Python310\iDRAC_mockup_client
Response time: 0 seconds
Serving Mockup in absolute path: C:\Python310\iDRAC_mockup_client
Serving Redfish mockup on port: 8000
running Server...
```

Note—Ensure that you do not exit the command line that is running the Redfish mockup server. If you exit, it stops the Redfish mockup server session.

The following is an example script to start the Redfish mockup server in Linux OS that points to the Redfish mockup client directory:

```
[root@localhost opt]# python redfishMockupServer.py -D
/junk/iDRAC_Redfish_mockup_client_R650 -X

Redfish Mockup Server, version 1.2.9
Hostname: 127.0.0.1
Port: 8000
Mockup directory path specified: /junk/iDRAC_Redfish_mockup_client_R650
Response time: 0 seconds
```

```
Serving Mockup in absolute path: /junk/iDRAC_Redfish_mockup_client_R650
Serving Redfish mockup on port: 8000
running Server...
```

2.2 Run simulated iDRAC Redfish calls

After the Redfish mockup server is run, you can run the Redfish calls using any supported method. For examples: Python, Curl, Postman, and PowerShell. To start the Redfish calls, use the specified IP address, port, or the default localhost IP address or port. For example: 127.0.0.1:8000. For more information about Redfish workflow examples, see the Redfish FAQ Whitepaper.

2.2.1 Example of a get request on chassis schema

The following is an example to GET request on chassis schema using Postman and mockup server logs:

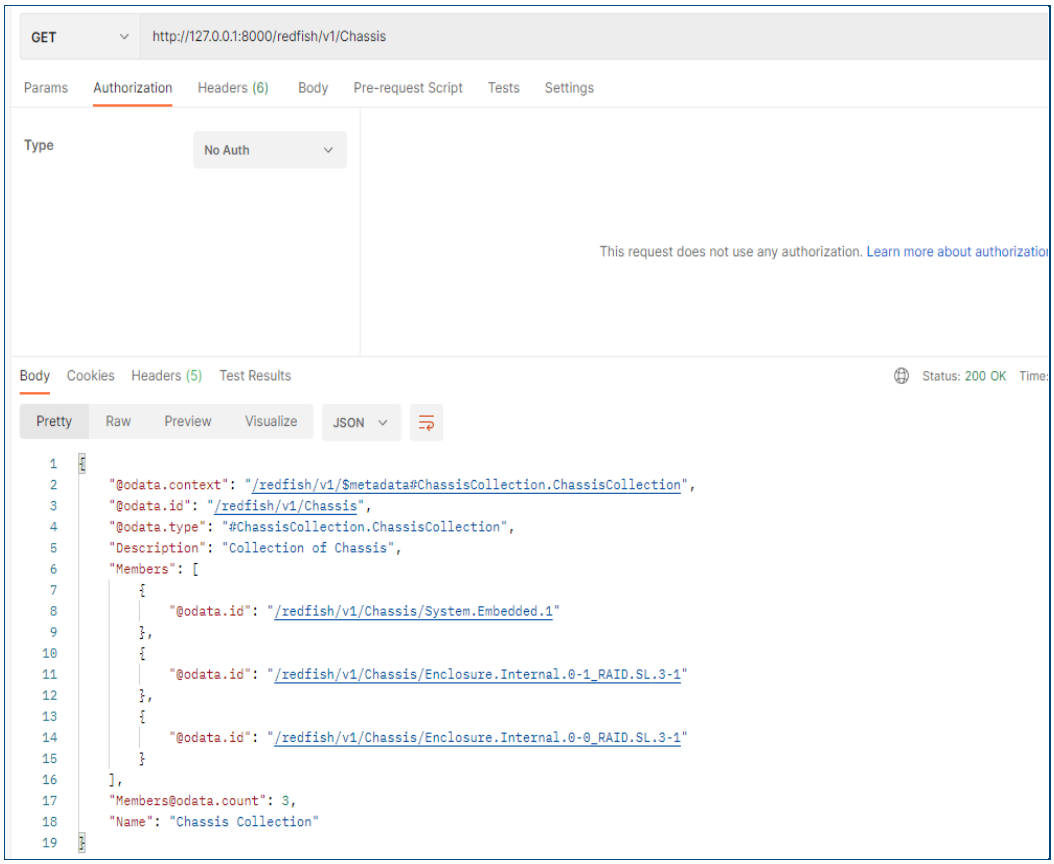

Figure 4 Example of a get request on chassis schema using Postman and mockup server log

```
C:\Python310>python redfishMockupServer.py -D C:\Python310\iDRAC_mockup_client -
X
Redfish Mockup Server, version 1.2.4
Hostname: 127.0.0.1
Port: 8000
Mockup directory path specified: C:\Python310\iDRAC_mockup_client
Response time: 0 seconds
Serving Mockup in absolute path: C:\Python310\iDRAC_mockup_client
Serving Redfish mockup on port: 8000
```

```
running Server...
('GET', '/redfish/v1/Chassis')
   GET: Headers: User-Agent: PostmanRuntime/7.29.2
Accept: */*
Postman-Token: 90a5c701-ae39-4e05-8fe5-369691c342ff
Host: 127.0.0.1:8000
Accept-Encoding: gzip, deflate, br
Connection: keep-alive


127.0.0.1 - - [30/Jan/2024 13:36:16] "GET /redfish/v1/Chassis HTTP/1.1" 200 -
('GET', '/redfish/v1/Chassi')
   GET: Headers: User-Agent: PostmanRuntime/7.29.2
Accept: */*
Postman-Token: c890fba0-756e-463b-95f2-9331d4fd36fe
Host: 127.0.0.1:8000
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
```

2.2.2 Example of a get request on the `UpdateService` schema using `$expand`

The following is an example to GET request about `UpdateService` schema using the `$expand` query parameter and mockup server logs:

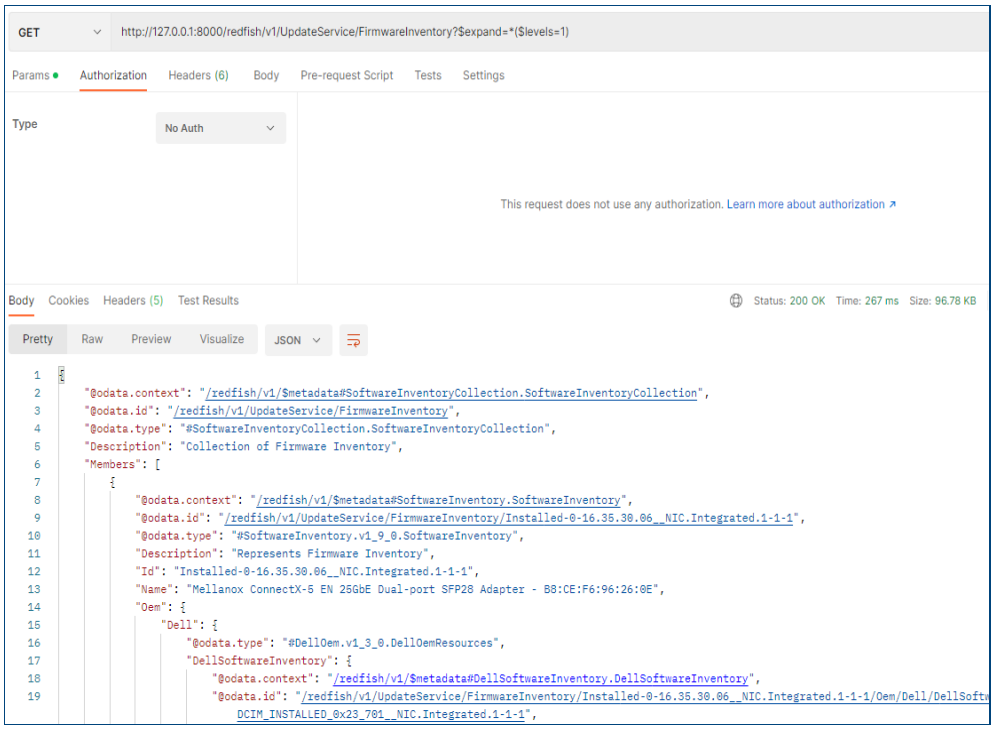

Figure 5 Example of a get request on the `UpdateService` schema using $`expand` query parameter and mockup server logs

```
127.0.0.1 - - [06/Feb/2024 12:54:56] "GET
/redfish/v1/UpdateService/FirmwareInventory HTTP/1.1" 200 -
('GET', '/redfish/v1/UpdateService/FirmwareInventory?$expand=*($levels=1)')
   GET: Headers: User-Agent: PostmanRuntime/7.29.2
Accept: */*
Postman-Token: 6e0a137d-f117-47f8-92b8-aee7f2c1eed1
Host: 127.0.0.1:8000
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
```

2.2.3 Example of a patch request on the `BootSourceOverrideTarget` property

The following is an example that displays a PATCH request to change the `BootSourceOverrideTarget` value from **None** to **Pxe**. After the PATCH request returns success, run the get command to see the updated property value (mockup server logs are provided in the following example:)

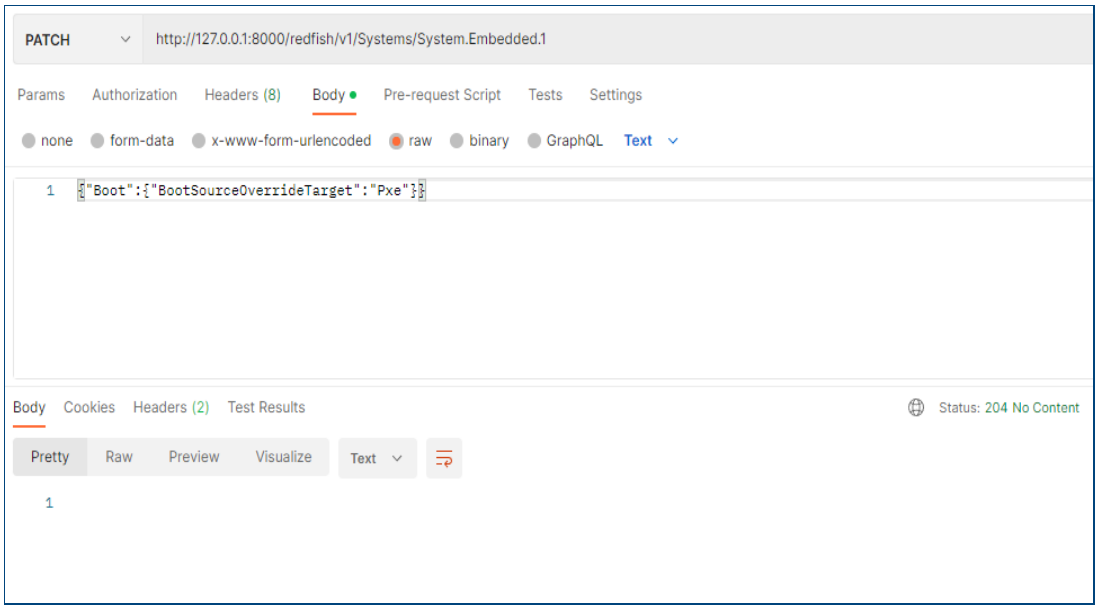

Figure 6 Example of a patch request to change the `BootSourceOverrideTarget` value from **None** to **Pxe**

```
PATCH: Data: {'Boot': {'BootSourceOverrideTarget': 'Pxe'}}
text/plain
{'Boot': {'BootSourceOverrideTarget': 'Pxe'}}
{'@Redfish.Settings': {'@odata.context':
'/redfish/v1/$metadata#Settings.Settings', '@odata.type':
'#Settings.v1_3_5.Settings', 'SettingsObject': {'@odata.id':
'/redfish/v1/Systems/System.Embedded.1/Settings'}, 'SupportedApplyTimes':
['OnReset']}, '@odata.context':
'/redfish/v1/$metadata#ComputerSystem.ComputerSystem', '@odata.id':
'/redfish/v1/Systems/System.Embedded.1', '@odata.type':
'#ComputerSystem.v1_20_1.ComputerSystem', 'Actions': {'#ComputerSystem.Reset':
{'target': '/redfish/v1/Systems/System.Embedded.1/Actions/ComputerSystem.Reset',
'ResetType@Redfish.AllowableValues': ['On', 'ForceOff', 'ForceRestart',
'GracefulRestart', 'GracefulShutdown', 'PushPowerButton', 'Nmi',
'PowerCycle']}}, 'AssetTag': '123456', 'Bios': {'@odata.id':
'/redfish/v1/Systems/System.Embedded.1/Bios'}, 'BiosVersion': '1.12.1',
'BootProgress': {'LastState': 'OSRunning'}, 'Boot': {'BootOptions':
{'@odata.id': '/redfish/v1/Systems/System.Embedded.1/BootOptions'},
'Certificates': {'@odata.id':
'/redfish/v1/Systems/System.Embedded.1/Boot/Certificates'}, 'BootOrder':
['Boot0010', 'Boot000E', 'Boot000D', 'Boot0001', 'Boot0003'],
'BootOrder@odata.count': 5, 'BootSourceOverrideEnabled': 'Disabled',
'BootSourceOverrideMode': 'UEFI', 'BootSourceOverrideTarget': 'Pxe',
'UefiTargetBootSourceOverride': '3A191845-5F86-4E78-8FCE-C4CFF59F9DAA',
'BootSourceOverrideTarget@Redfish.AllowableValues': ['None', 'Pxe', 'Floppy',
'Cd', 'Hdd', 'BiosSetup', 'Utilities', 'UefiTarget', 'SDCard', 'UefiHttp'],
'StopBootOnFault': 'Never'}, 'Description': 'Computer System which represents a
machine (physical or virtual) and the local resources such as memory, cpu, and
other devices that can be accessed from that machine.', 'EthernetInterfaces':
{'@odata.id': '/redfish/v1/Systems/System.Embedded.1/EthernetInterfaces'},
'GraphicalConsole': {'ConnectTypesSupported': ['KVMIP'],
'ConnectTypesSupported@odata.count': 1, 'MaxConcurrentSessions': 6,
```

```
'ServiceEnabled': True}, 'HostName': 'localhost.ept.adc.delllabs.net',
'HostWatchdogTimer': {'FunctionEnabled': False, 'Status': {'State': 'Disabled'},
'TimeoutAction': 'None'}, 'HostingRoles': [], 'HostingRoles@odata.count': 0,
'Id': 'System.Embedded.1', 'IndicatorLED': 'Lit',
'IndicatorLED@Redfish.Deprecated': 'Please migrate to use
LocationIndicatorActive property', 'Links': {'Chassis': [{'@odata.id':
'/redfish/v1/Chassis/System.Embedded.1'}], 'Chassis@odata.count': 1, 'CooledBy':
[{'@odata.id': '/redfish/v1/Chassis/System.Embedded.1/Thermal#/Fans/0'}]}
127.0.0.1 - - [01/Feb/2024 10:11:17] "PATCH
/redfish/v1/Systems/System.Embedded.1 HTTP/1.1" 204 -
```

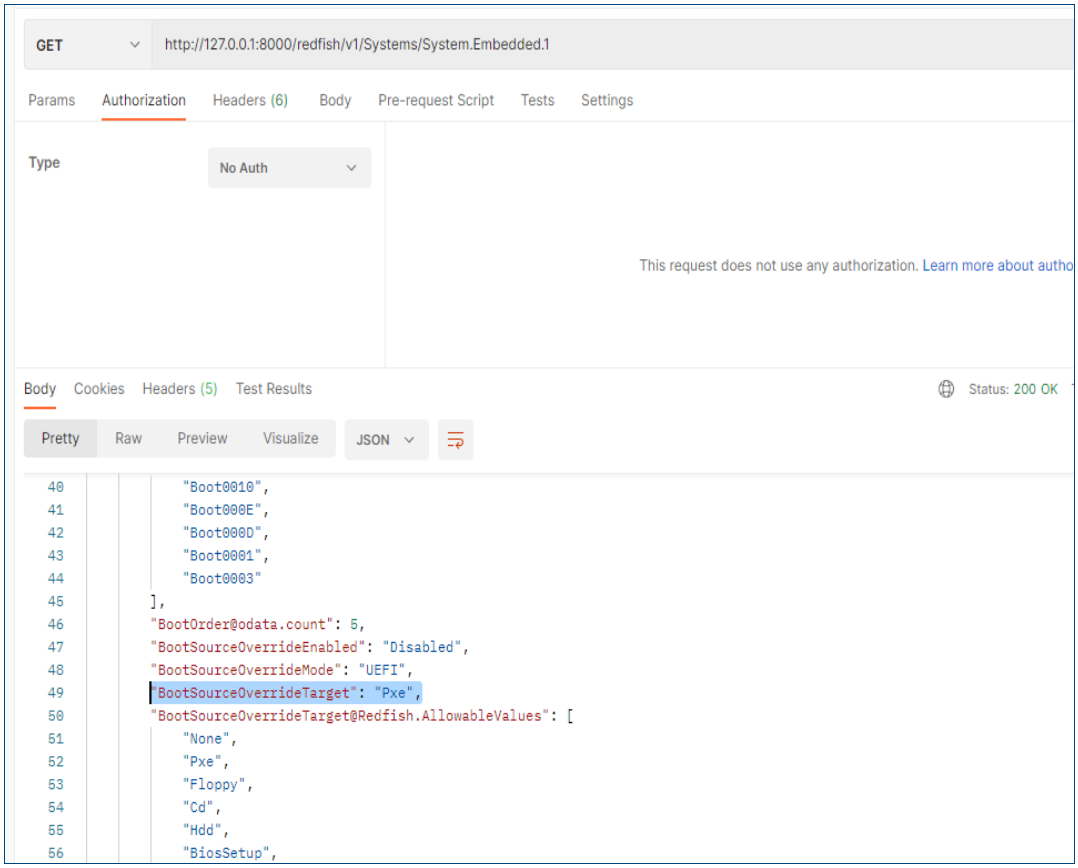

Figure 7 Example of a patch request to change the `BootSourceOverrideTarget` value from **None** to **Pxe**

```
127.0.0.1 - - [06/Feb/2024 12:55:06] "GET
/redfish/v1/UpdateService/FirmwareInventory?$expand=*($levels=1) HTTP/1.1" 200 -
('GET', '/redfish/v1/Systems/System.Embedded.1')
   GET: Headers: User-Agent: PostmanRuntime/7.29.2
Accept: */*
Postman-Token: b9674463-ad22-47e8-9743-7901f355e0a7
Host: 127.0.0.1:8000
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
```

2.2.4 Example of a patch request using OEM system attribute

An example that displays the following:

- Gets the current iDRAC system power cap settings.
- Changes values using PATCH operation.
- Confirms new values along with mockup server logs.

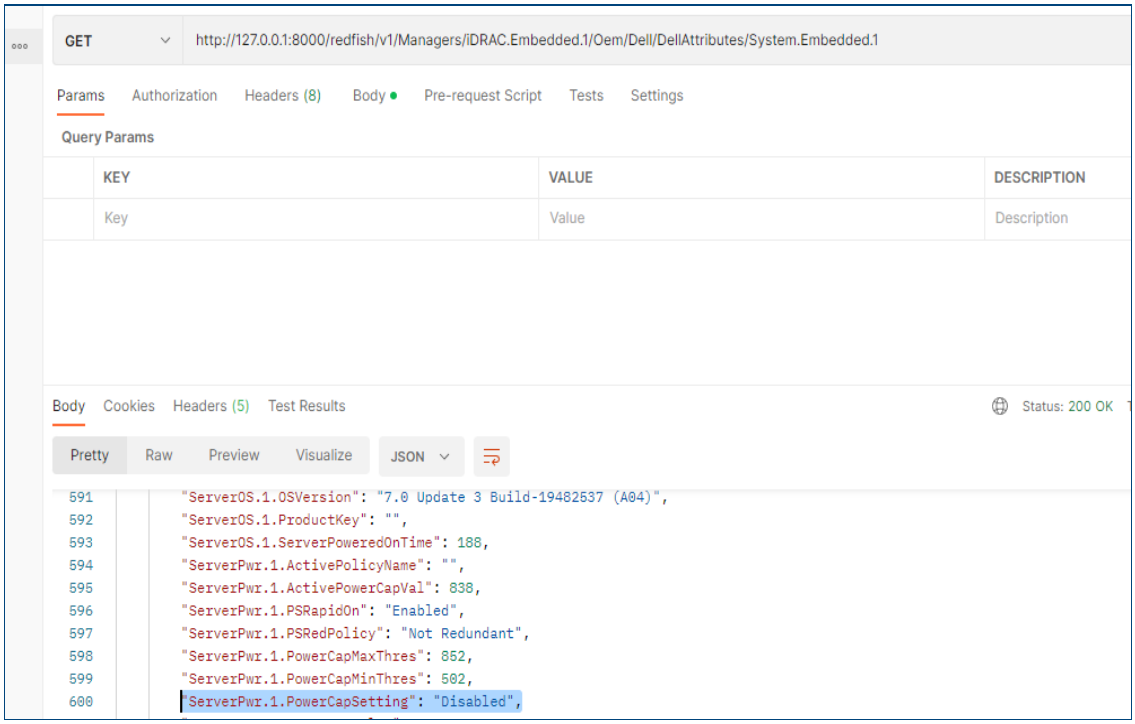

Figure 8 Example of a patch request

```
('GET',
'/redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/System.Embedded.1
')
   GET: Headers: Content-Type: text/plain
User-Agent: PostmanRuntime/7.29.2
Accept: */*
Postman-Token: 12c4eb52-6e37-4a2f-85a1-c6ddcd0e3ed5
Host: 127.0.0.1:8000
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Content-Length: 53
```

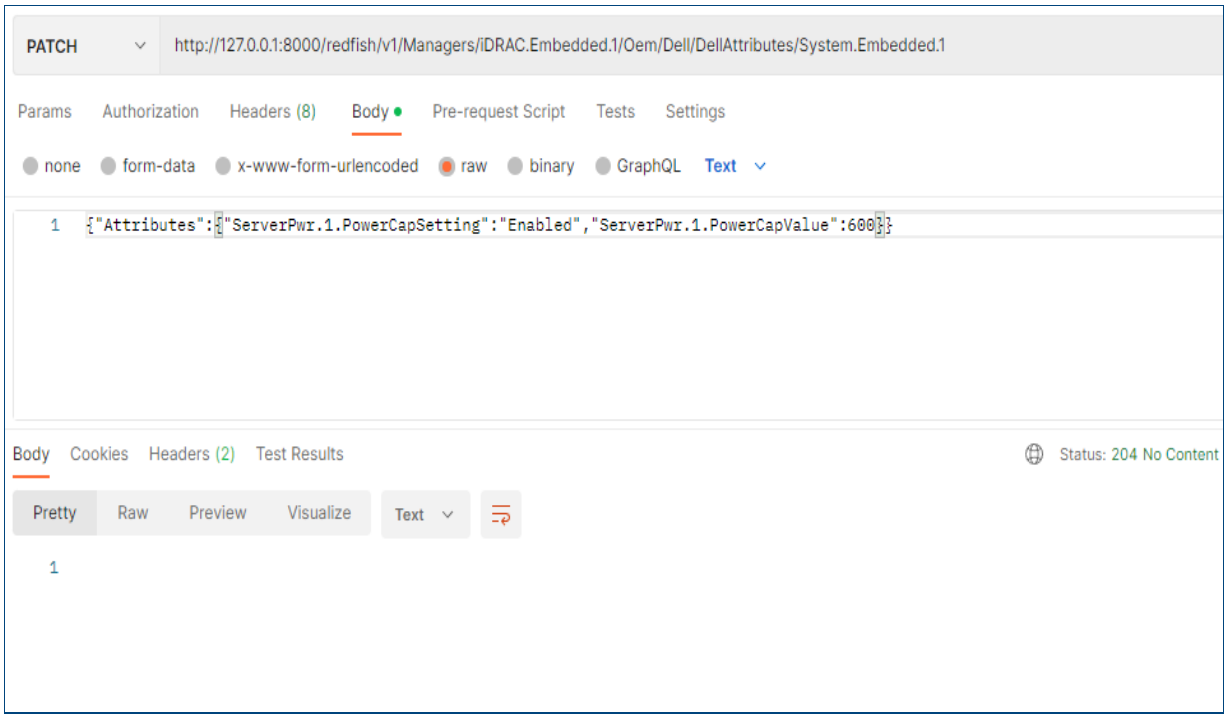

Figure 9 Example of a patch request

```
127.0.0.1 - - [06/Feb/2024 13:34:53] "GET
/redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/System.Embedded.1
HTTP/1.1" 200 -
   PATCH: Headers: Content-Type: text/plain
User-Agent: PostmanRuntime/7.29.2
Accept: */*
Postman-Token: c3843b5d-78a0-4ee9-aa60-10448c8a3a16
Host: 127.0.0.1:8000
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Content-Length: 88


   PATCH: Data: {'Attributes': {'ServerPwr.1.PowerCapSetting': 'Enabled',
'ServerPwr.1.PowerCapValue': 600}}
text/plain
{'Attributes': {'ServerPwr.1.PowerCapSetting': 'Enabled',
'ServerPwr.1.PowerCapValue': 600}}
```

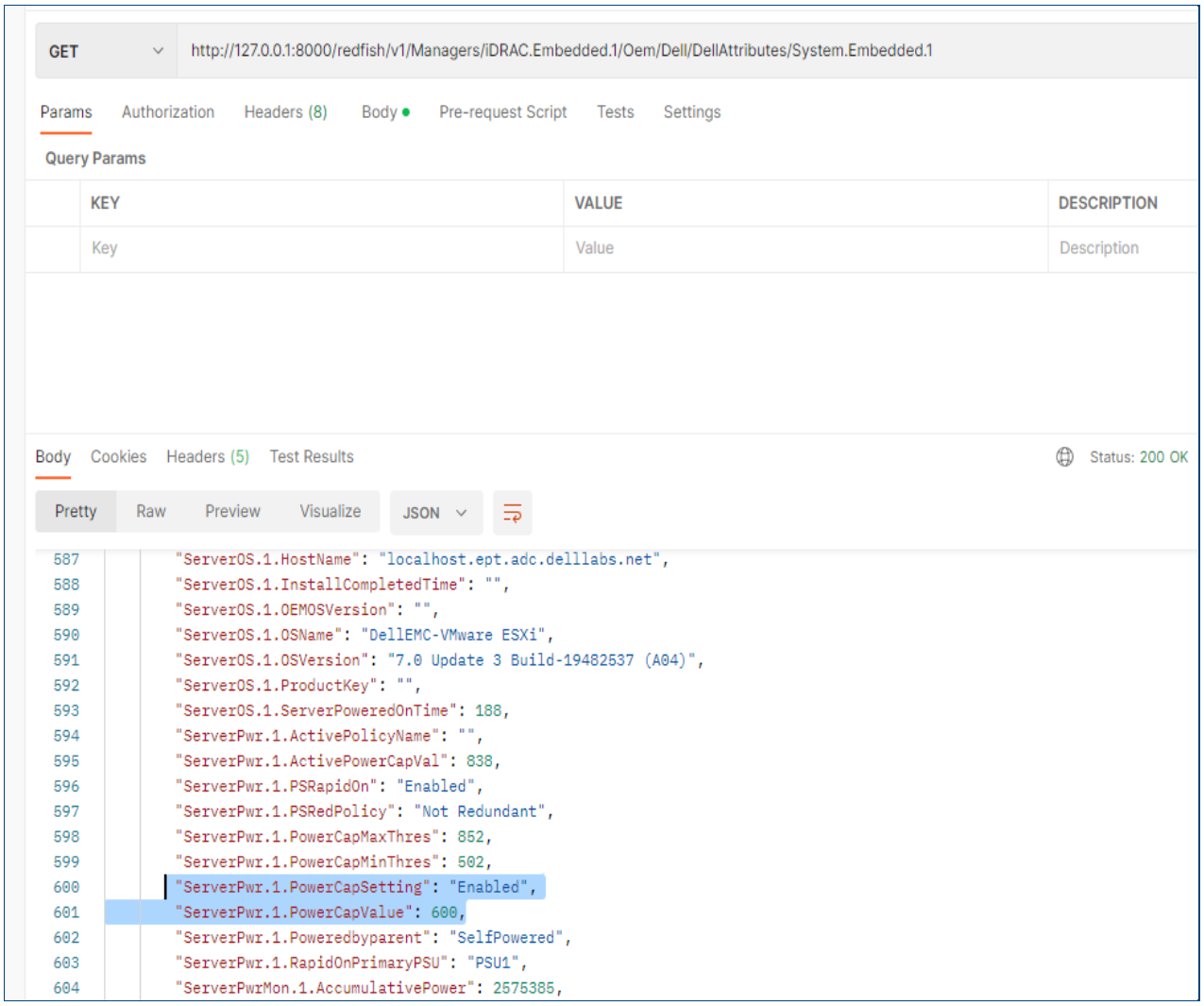

Figure 10 Example of a patch request

```
('GET',
'/redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/System.Embedded.1
')
   GET: Headers: Content-Type: text/plain
User-Agent: PostmanRuntime/7.29.2
Accept: */*
Postman-Token: c124e851-b34e-4b43-b9e4-db46fe812f8d
Host: 127.0.0.1:8000
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Content-Length: 53
```

2.2.5 Example of a patch request to create an iDRAC user

The following is an example to create an iDRAC user (ID 4). The GET command is leveraged to validate a new user that is created and enabled.

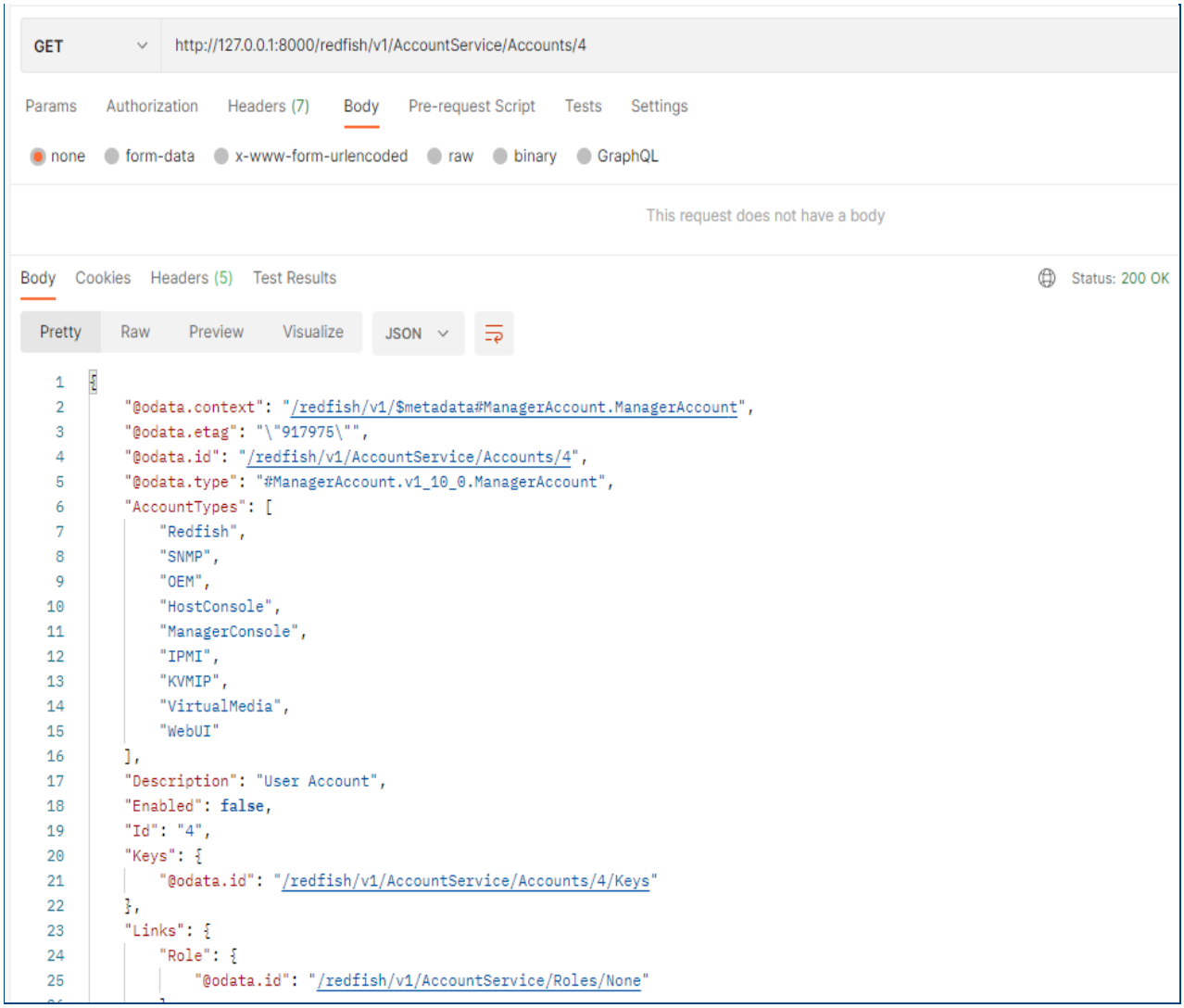

Figure 11 Example of a patch request to create an iDRAC user

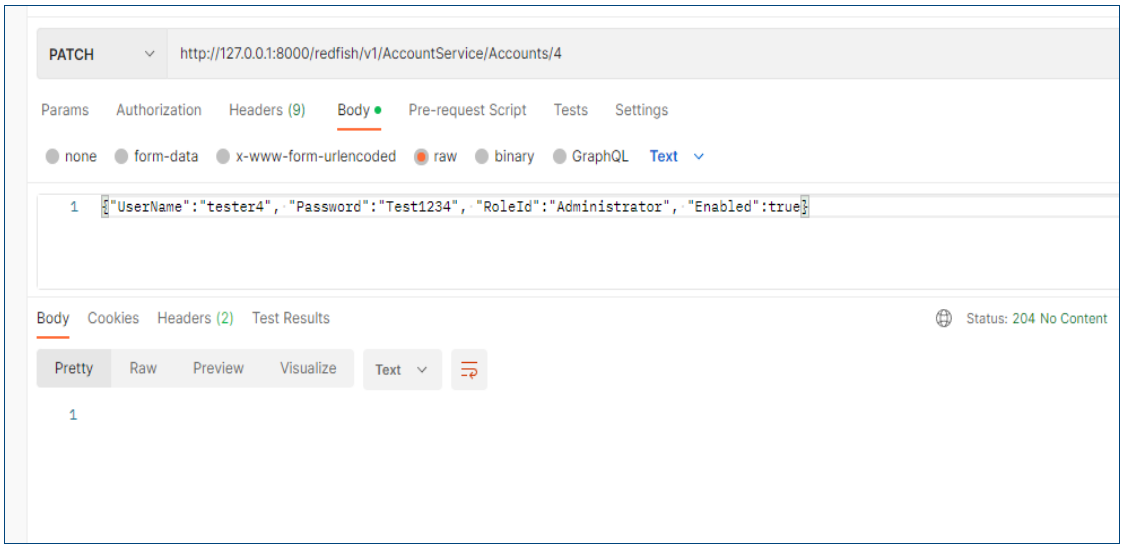

Figure 12 Example of a patch request to create an iDRAC user

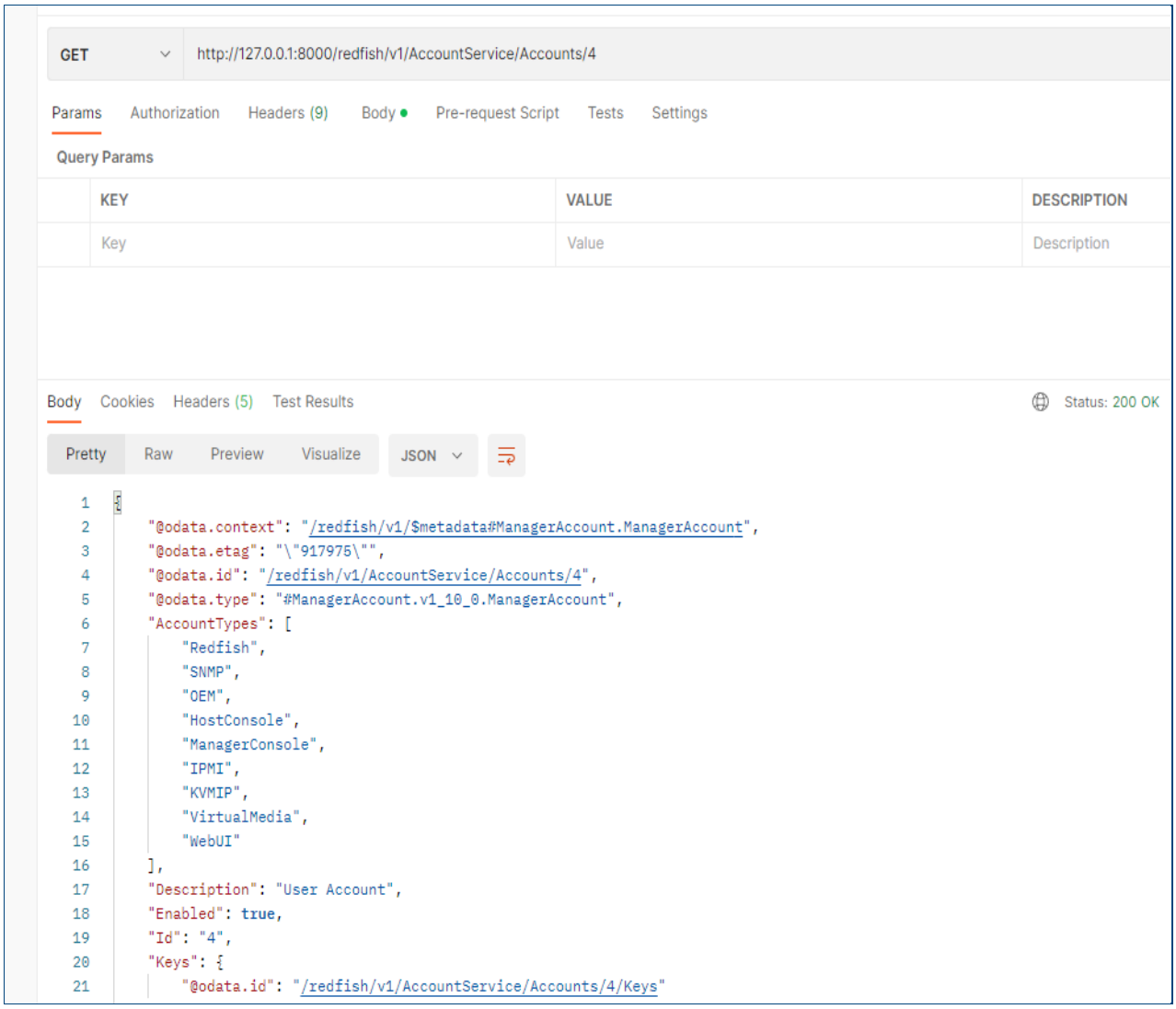

Figure 13 Example of a patch request to create an iDRAC user

```
    "Oem": {
        "Dell": {
            "@odata.type": "#DellManagerAccount.v1_0_0.DellManagerAccount",
            "SNMPv3PassphraseEnabled": "Disabled"
        }
    },
    "Password": "Test1234",
    "PasswordChangeRequired": false,
    "PasswordExpiration": null,
    "RoleId": "Administrator",
    "SNMP": {
        "AuthenticationKey": null,
        "AuthenticationKeySet": false,
        "AuthenticationProtocol": "HMAC_SHA96",
        "EncryptionKey": null,
        "EncryptionKeySet": false,
        "EncryptionProtocol": "CFB128_AES128"
    },
    "StrictAccountTypes": false,
    "UserName": "tester4"
}
```

Figure 14 Example of a patch request to create an iDRAC user

Note—In the above example, the Password property that is displayed in clear text occurs only on the Redfish mockup server. However, the iDRAC does not display passwords in clear text.

2.2.6 Example of a patch request to change the iDRAC user password

The following is an example to change iDRAC user (ID 4) password using PATCH operation.

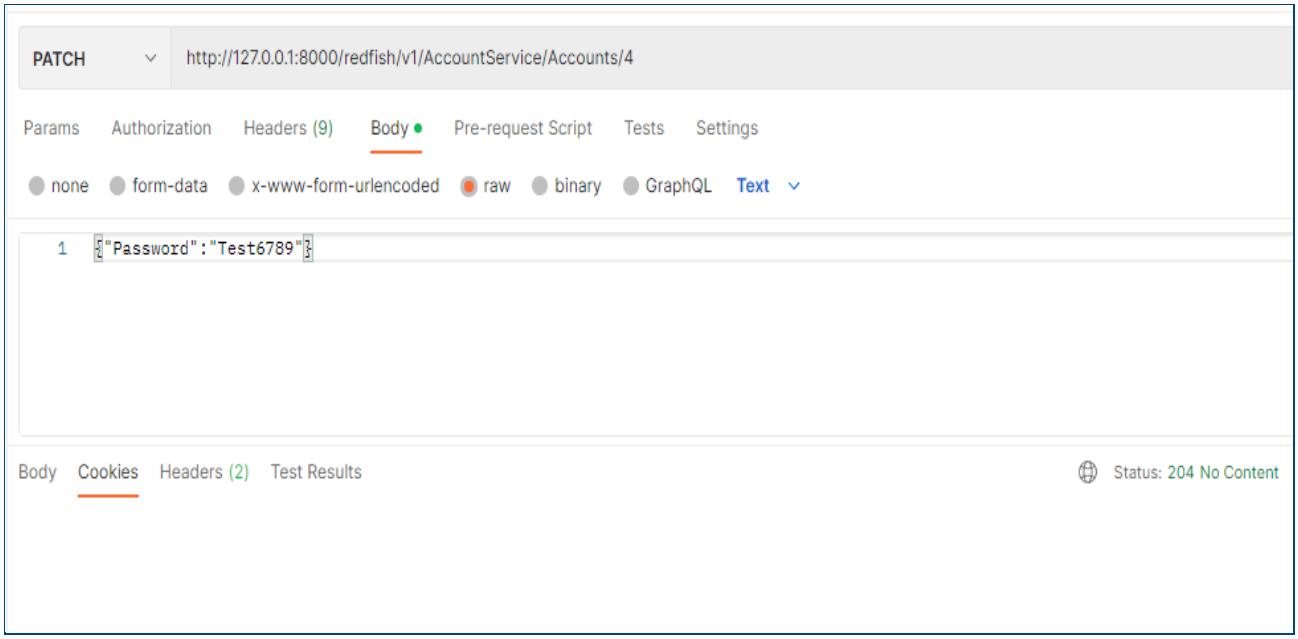

Figure 15 Example of a patch request to change the iDRAC user password

2.2.7 Example of a post request using the `InsertMedia` command

The following is an example that displays a POST request using the `InsertMedia` command to attach virtual media International Organization for Standardization (ISO).

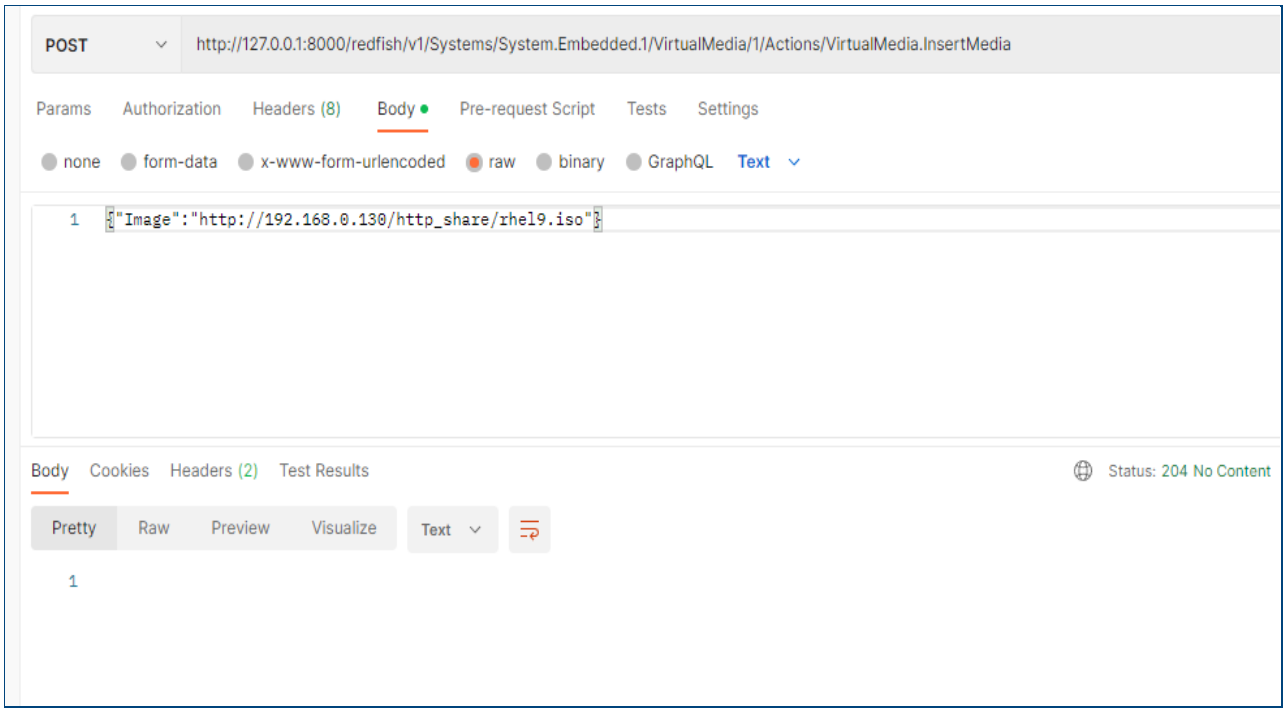

Figure 16 Example of a post request uisg the `InsertMedia` command

```
 POST: Data: {}
127.0.0.1 - - [01/Feb/2024 10:27:18] "POST
/redfish/v1/Systems/System.Embedded.1/VirtualMedia/1/Actions/VirtualMedia.Insert
Media HTTP/1.1" 204 -
   POST: Headers: Content-Type: text/plain
User-Agent: PostmanRuntime/7.29.2
Accept: */*
Postman-Token: 16dd3465-3d42-4ddb-afee-a95da1c30bac
Host: 127.0.0.1:8000
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Content-Length: 53


 POST: Data: {'Image': 'http://192.168.0.130/http_share/rhel9.iso'}
127.0.0.1 - - [01/Feb/2024 10:28:06] "POST
/redfish/v1/Systems/System.Embedded.1/VirtualMedia/1/Actions/VirtualMedia.Insert
Media HTTP/1.1" 204 -
```

2.2.8 Example of a post request using the `ExportSystemConfiguration` OEM command

The following is an example that runs the `ExportSystemConfiguration` OEM command to export iDRAC system configuration information locally along with mockup server logs.

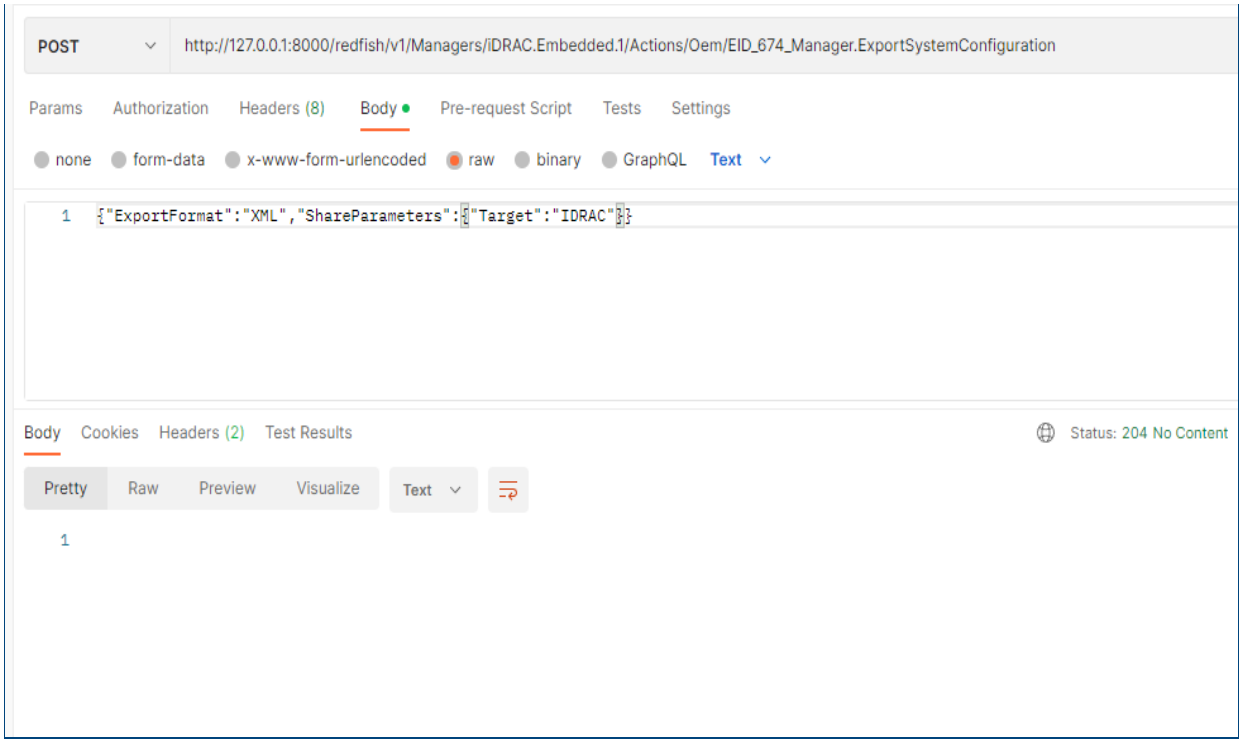

Figure 17 Example of a post request using the `ExportSystemConfiguration` OEM command

```
127.0.0.1 - - [06/Feb/2024 13:26:27] "POST
/redfish/v1/Managers/iDRAC.Embedded.1/Actions/Oem/EID_674_Manager.ExportSystemCo
nfiguration HTTP/1.1" 204 -
   POST: Headers: Content-Type: text/plain
User-Agent: PostmanRuntime/7.29.2
Accept: */*
Postman-Token: 13ef4539-f8b5-4a2e-88d1-9c2742f57bdb
Host: 127.0.0.1:8000
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Content-Length: 59


   POST: Data: {'ExportFormat': 'XML', 'ShareParameters': {'Target': 'IDRAC'}}
127.0.0.1 - - [06/Feb/2024 13:27:11] "POST
/redfish/v1/Managers/iDRAC.Embedded.1/Actions/Oem/EID_674_Manager.ExportSystemCo
nfiguration HTTP/1.1" 204 -
```

2.2.9 Example of a post request to create a session

The following is an example to get info about VDs associated with a storage controller, delete a VD, and identify a VD that no longer is associated with mockup.

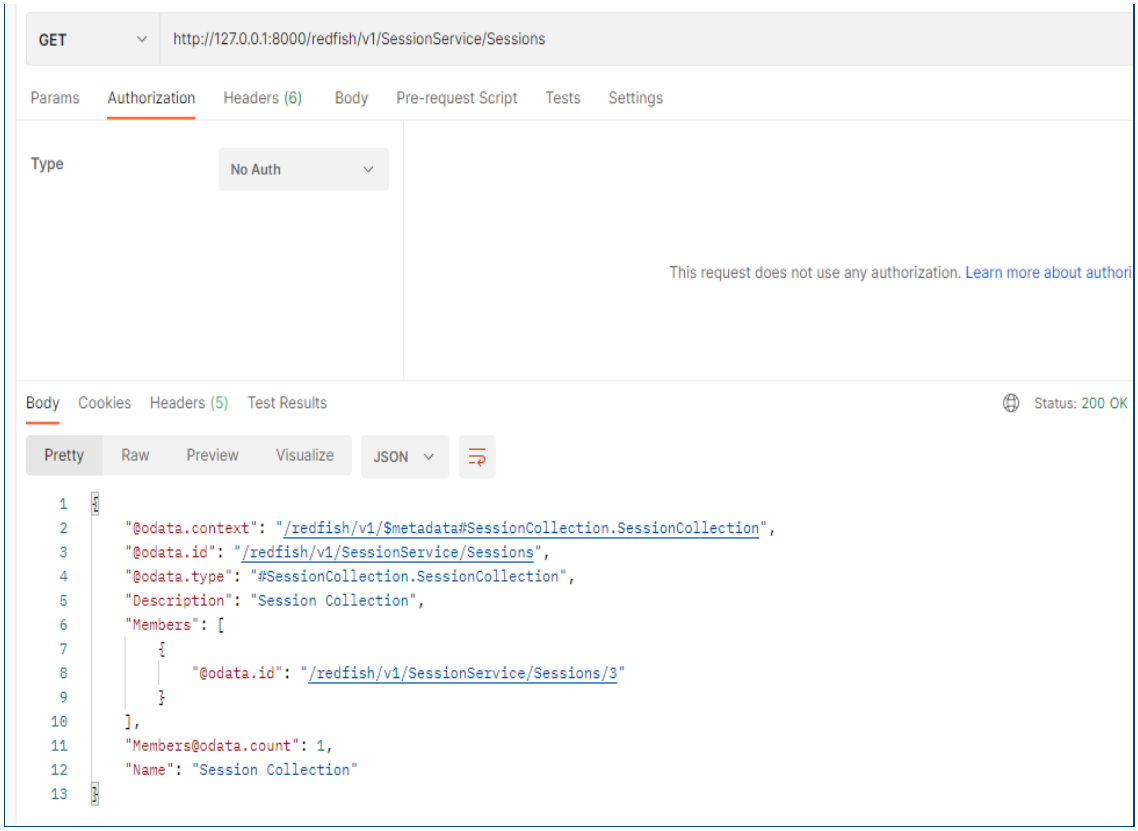

Figure 18 Example of a post request to create a session

```
('GET', '/redfish/v1/SessionService/Sessions')
   GET: Headers: User-Agent: PostmanRuntime/7.29.2
Accept: */*
Postman-Token: f581274f-f930-491a-b758-bdb5360716ec
Host: 127.0.0.1:8000
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
```

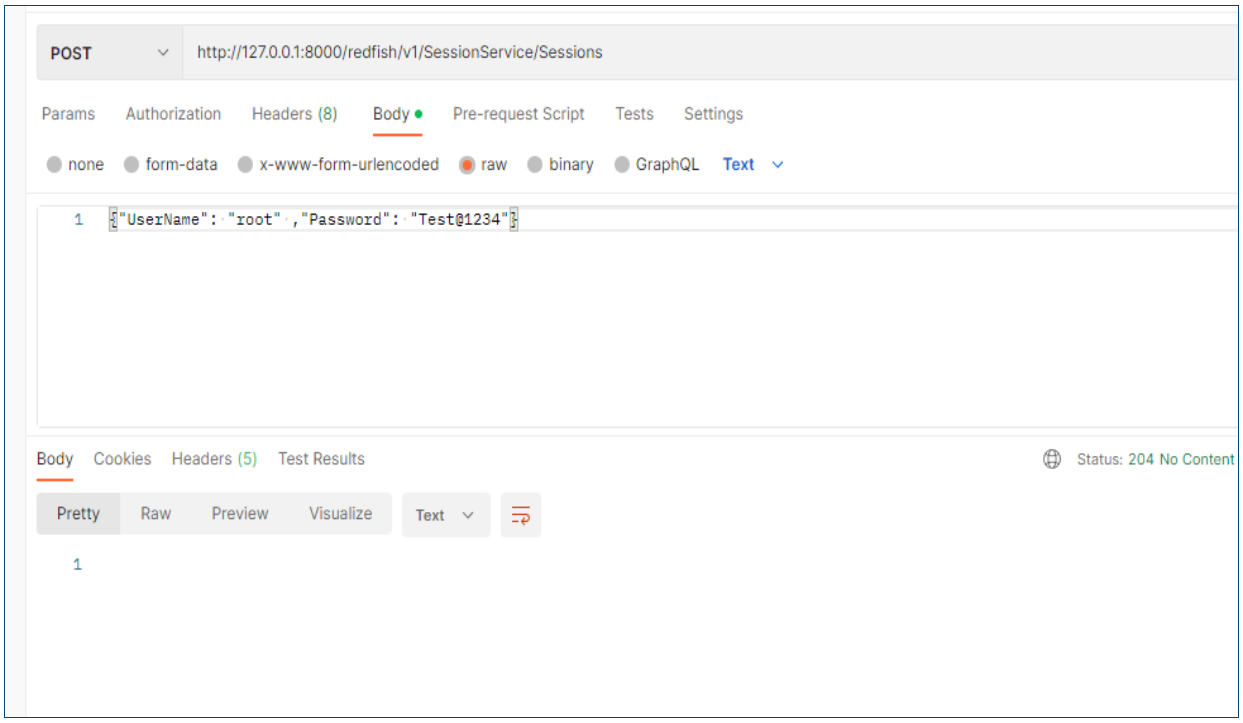

Figure 19 Example of a post request to create a session

```
127.0.0.1 - - [06/Feb/2024 14:11:45] "GET /redfish/v1/SessionService/Sessions
HTTP/1.1" 200 -
   POST: Headers: Content-Type: text/plain
User-Agent: PostmanRuntime/7.29.2
Accept: */*
Postman-Token: 64806302-13e0-499a-819b-603e03edfaf2
Host: 127.0.0.1:8000
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Content-Length: 45


   POST: Data: {'UserName': 'root', 'Password': 'Test@1234'}
{'UserName': 'root', 'Password': 'Test@1234'}
<class 'dict'>
C:\Python310\iDRAC_mockup_client/redfish/v1/SessionService/Sessions/1/index.json
127.0.0.1 - - [06/Feb/2024 14:16:01] "POST /redfish/v1/SessionService/Sessions
HTTP/1.1" 204 -
```

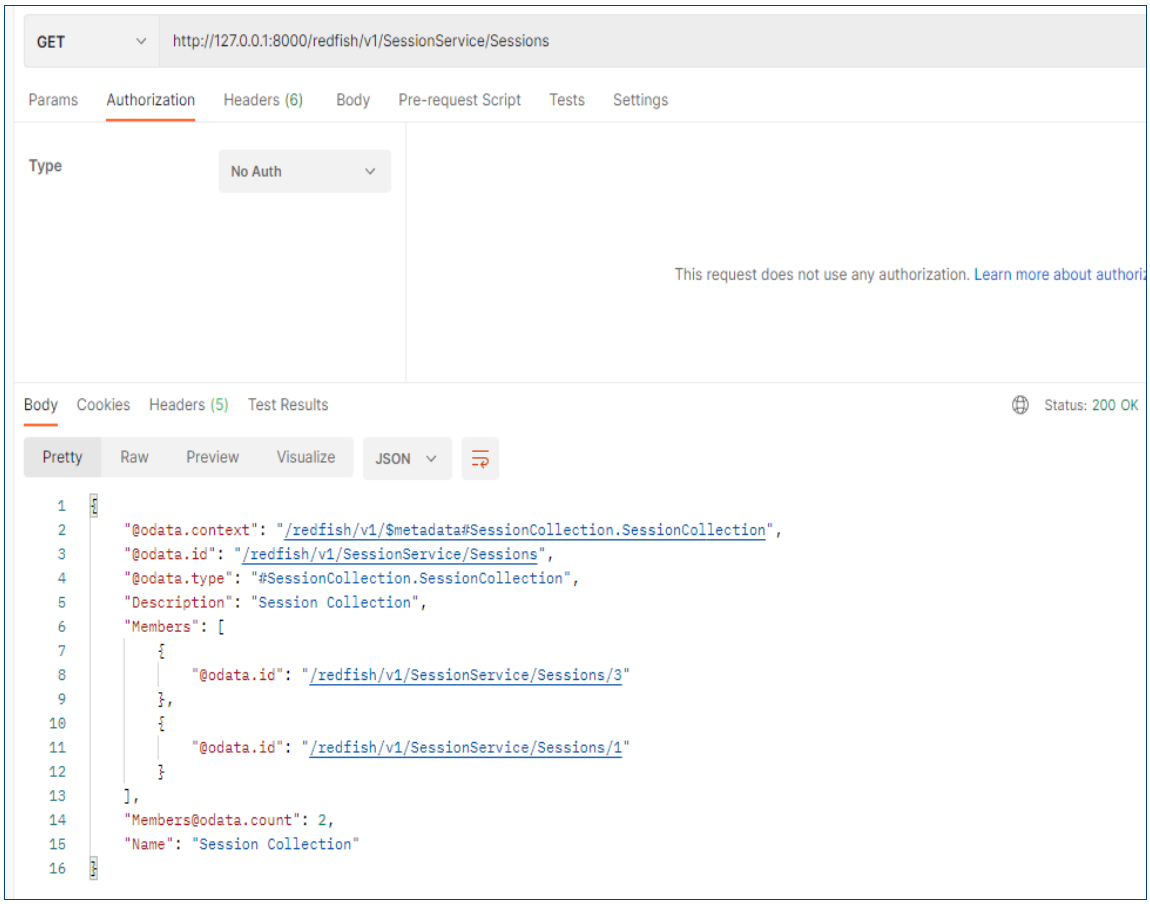

Figure 20 Example of a post request to create a session

```
127.0.0.1 - - [06/Feb/2024 14:16:44] "GET /redfish/v1/SessionService/Sessions
HTTP/1.1" 200 -
('GET', '/redfish/v1/SessionService/Sessions')
   GET: Headers: User-Agent: PostmanRuntime/7.29.2
Accept: */*
Postman-Token: b273a0ca-d407-499b-aeb8-435f0fc10660
Host: 127.0.0.1:8000
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
```

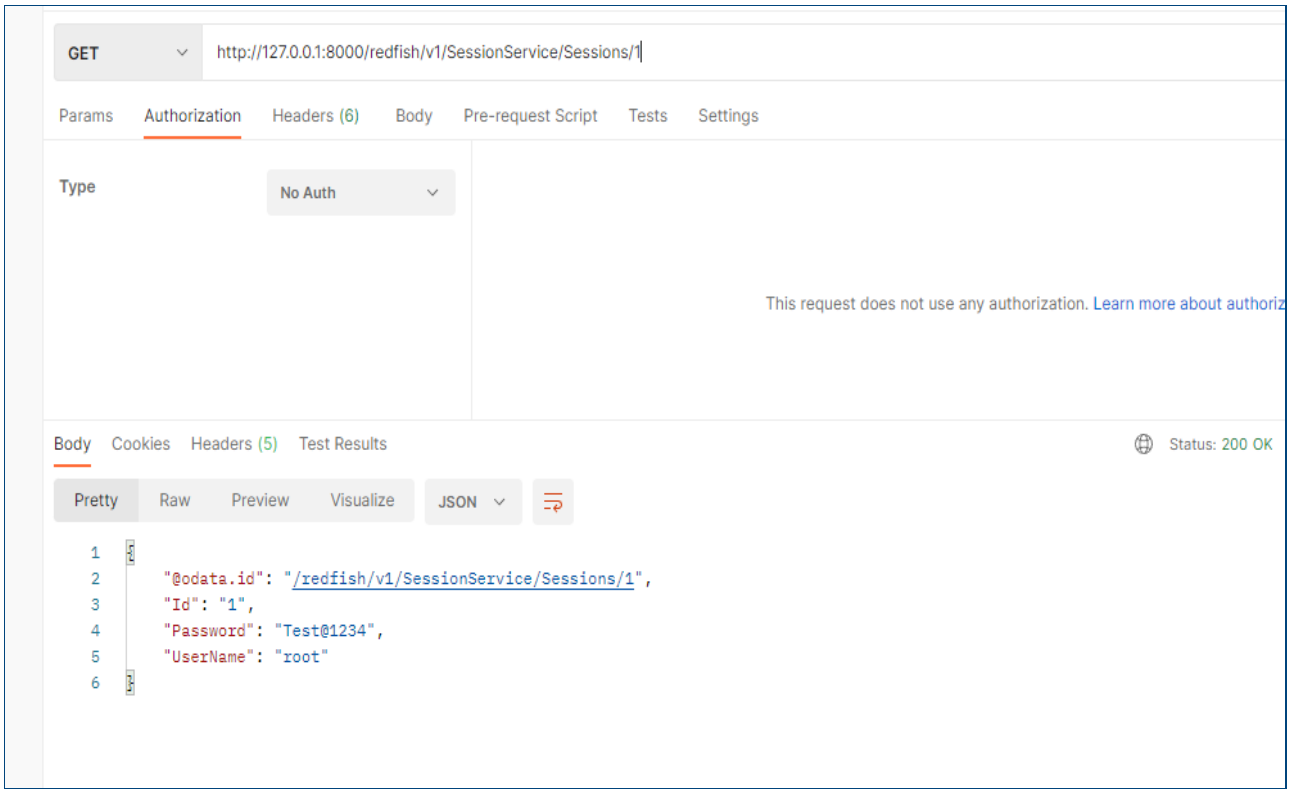

Figure 21 Example of a post request to create a session

```
127.0.0.1 - - [06/Feb/2024 14:16:44] "GET /redfish/v1/SessionService/Sessions
HTTP/1.1" 200 -
('GET', '/redfish/v1/SessionService/Sessions/1')
   GET: Headers: User-Agent: PostmanRuntime/7.29.2
Accept: */*
Postman-Token: b273a0ca-d407-499b-aeb8-435f0fc10660
Host: 127.0.0.1:8000
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
```

2.2.10 Example of a delete request to delete a virtual disk

The following is an example to get virtual disks for the storage controller, delete the virtual disk and confirm a virtual disk that no longer exists with mockup server logs.

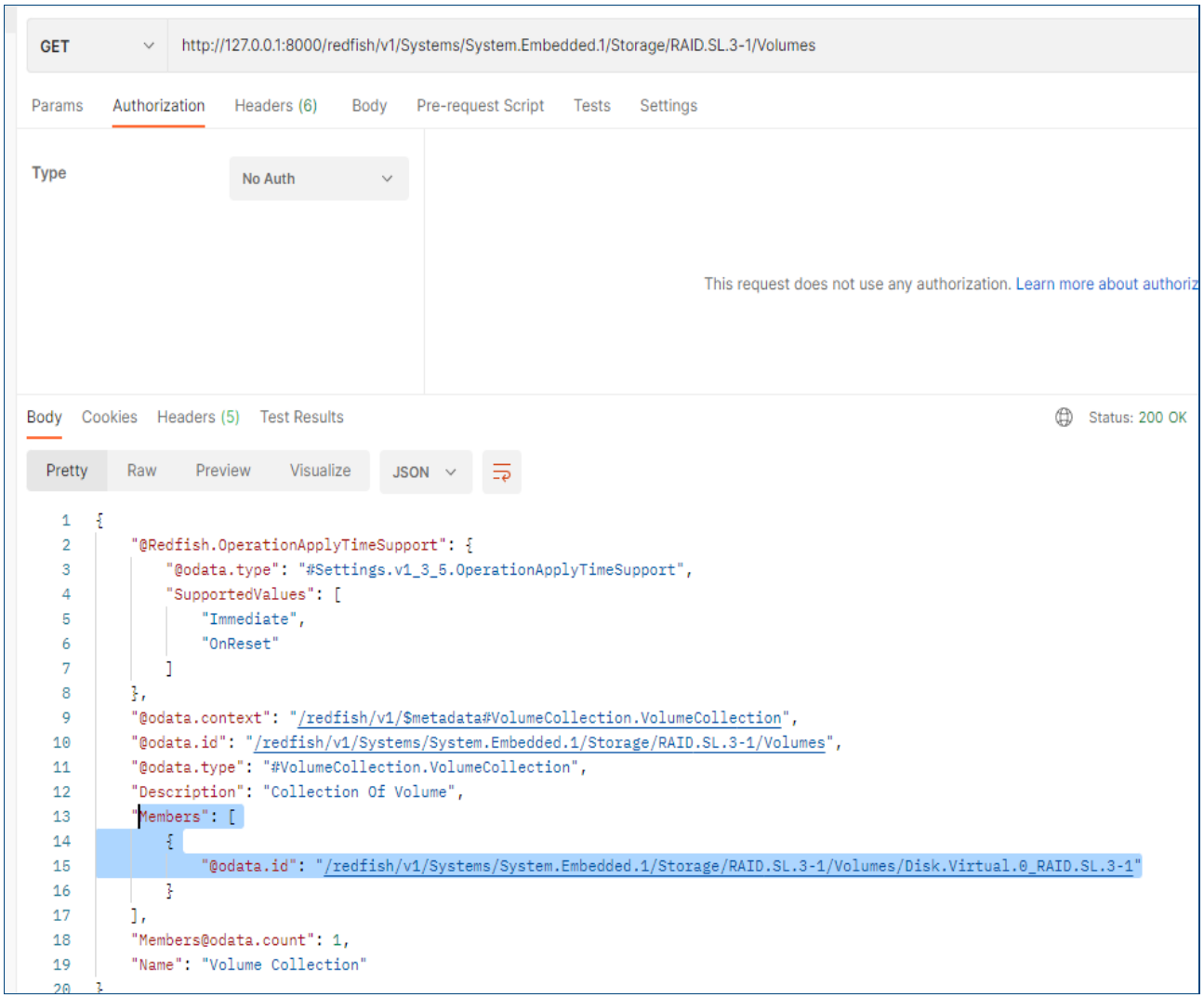

Figure 22 Example of a delete request to delete virtual disk

```
127.0.0.1 - - [06/Feb/2024 13:48:58] "GET
/redfish/v1/Systems/System.Embedded.1/Storage/RAID.SL.3-
1/Volumes/Disk.Virtual.0_RAID.SL.3-1 HTTP/1.1" 200 -
('GET', '/redfish/v1/Systems/System.Embedded.1/Storage/RAID.SL.3-1/Volumes')
   GET: Headers: Content-Type: text/plain
User-Agent: PostmanRuntime/7.29.2
Accept: */*
Postman-Token: a0eecfe4-c7ca-4b7a-a5fb-9dd9343815f1
Host: 127.0.0.1:8000
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Content-Length: 57
```

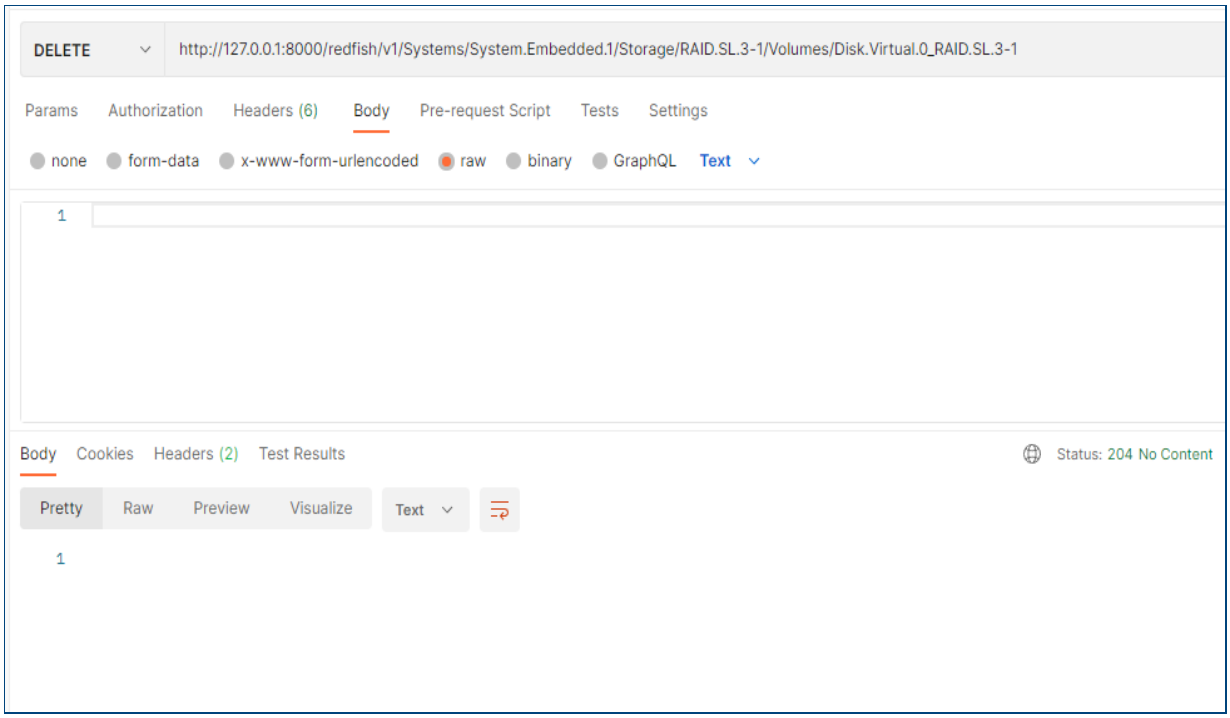

Figure 23 Example of a delete request to delete virtual disk

```
127.0.0.1 - - [06/Feb/2024 13:49:12] "GET
/redfish/v1/Systems/System.Embedded.1/Storage/RAID.SL.3-1/Volumes HTTP/1.1" 200
-
DELETE: Headers: User-Agent: PostmanRuntime/7.29.2
Accept: */*
Postman-Token: 155f8cd0-7f91-415d-b18a-db9a29cccc82
Host: 127.0.0.1:8000
Accept-Encoding: gzip, deflate, br
Connection: keep-alive


127.0.0.1 - - [06/Feb/2024 13:51:13] "DELETE
/redfish/v1/Systems/System.Embedded.1/Storage/RAID.SL.3-
1/Volumes/Disk.Virtual.0_RAID.SL.3-1 HTTP/1.1" 204 -
```

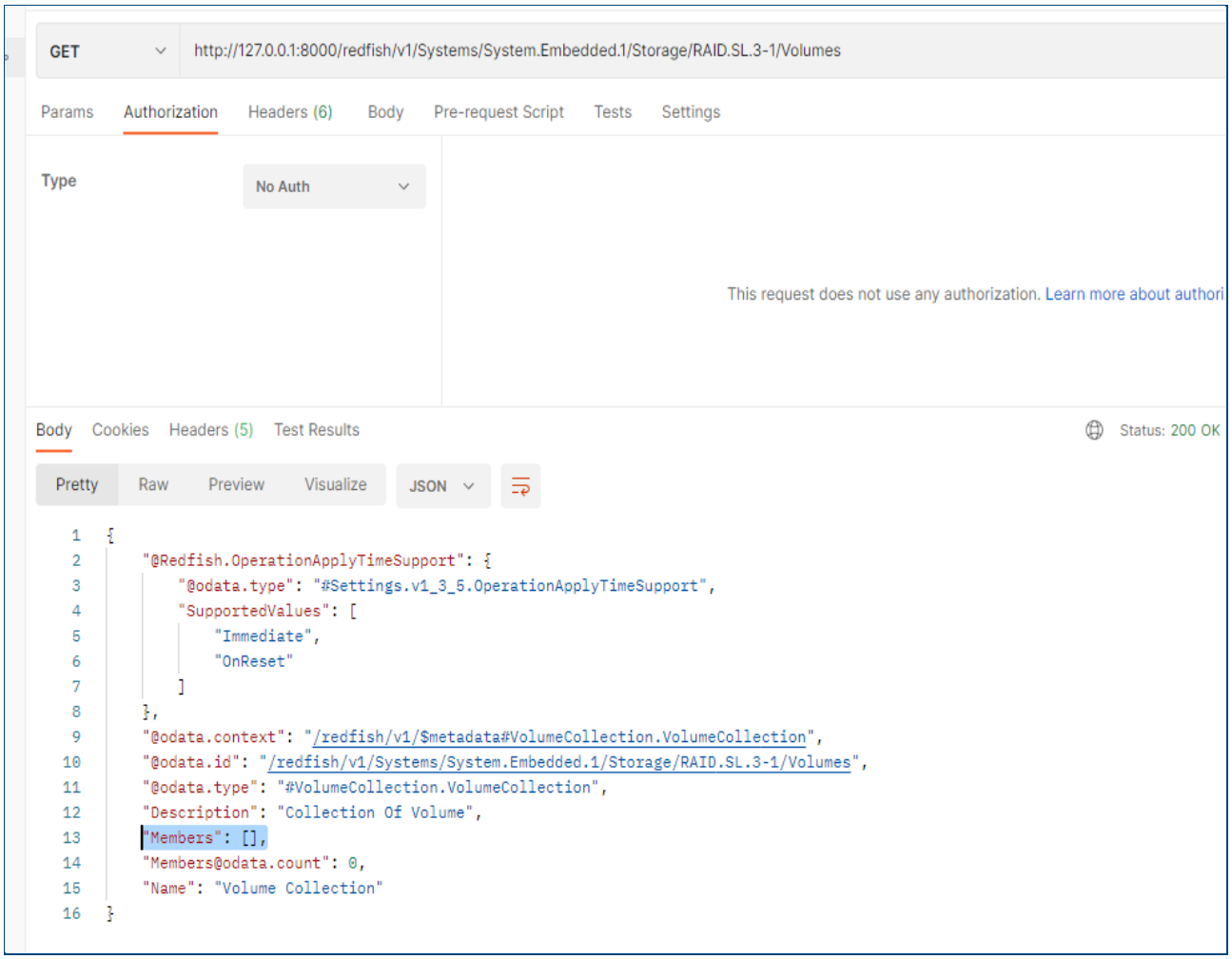

Figure 24 Example of a delete request to delete virtual disk

```
127.0.0.1 - - [06/Feb/2024 13:51:13] "DELETE
/redfish/v1/Systems/System.Embedded.1/Storage/RAID.SL.3-
1/Volumes/Disk.Virtual.0_RAID.SL.3-1 HTTP/1.1" 204 -
('GET', '/redfish/v1/Systems/System.Embedded.1/Storage/RAID.SL.3-1/Volumes')
   GET: Headers: User-Agent: PostmanRuntime/7.29.2
Accept: */*
Postman-Token: 8fc87501-0946-4279-aa53-874ebc638116
Host: 127.0.0.1:8000
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
```

3 Simulate streaming iDRAC Redfish events workflow

The Redfish mockup server and Redfish event listener enable you to simulate streaming iDRAC alert events. To simulate streaming iDRAC alert events, do the following:

- Configure Redfish event listener.
- Create a subscription.
- Verify the subscription exists.
- Start the Redfish event listener
- Submit the test event.
- Ensure that the Redfish event listener received the event.

To complete this simulation process, use the listeners available at the following location: Redfish-Event-Listener.

3.1 Configure Redfish event listener

Edit the Redfish event listener `config.ini` and ensure that the following properties are set:

- `ListenerPort = 80`
- `UseSSL = off`
- `Destination = http://{client IP that is running Redfish Event Listener}:80.`
- `ServerIPs = http://127.0.0.1:8000`
- `certcheck = off`

An example of an edited `config.ini`:

```
[Information]

Updated = April 24, 2017

Description = Redfish Event Listener Tool

[SystemInformation]

ListenerIP = 0.0.0.0

ListenerPort = 80

UseSSL = off

[CertificateDetails]

certfile = cert.pem

keyfile = server.key

[SubsciptionDetails]

Destination = http://192.168.0.130:80

Context = Public
```

```
Protocol = Redfish

SubscriptionURI = /redfish/v1/EventService/Subscriptions

[ServerInformation]

ServerIPs = http://127.0.0.1:8000

UserNames = root

Passwords = calvin

certcheck = off
```

3.2 Create a subscription

To create a subscription, use the same destination URI string that you passed in the `config.ini` file.

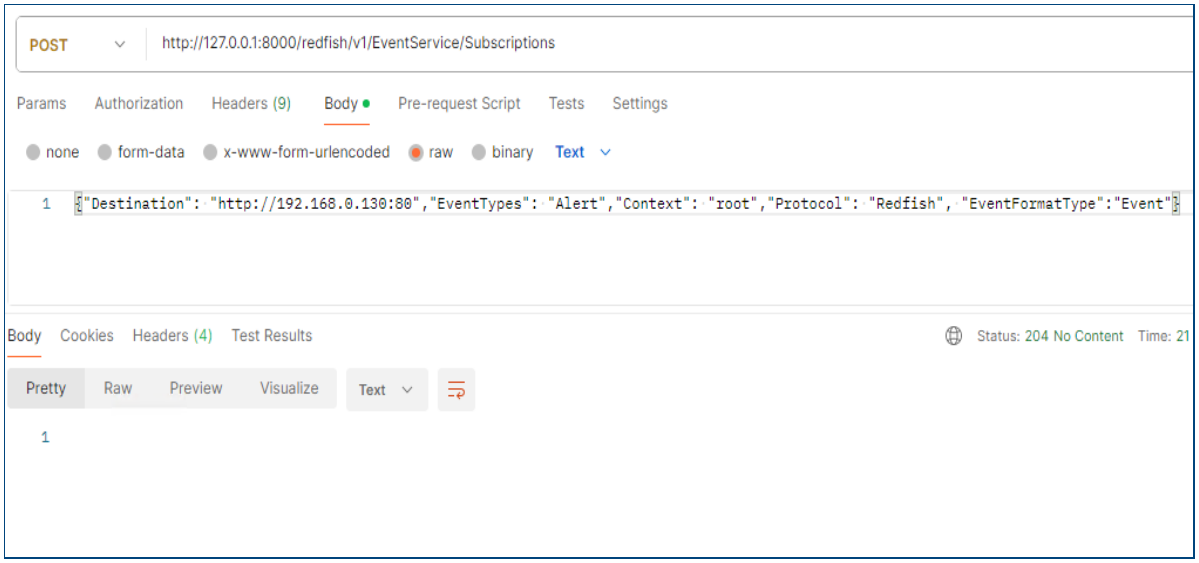

Figure 25 Create a subscription

3.3 Get subscription information

Get subscription information that is created to validate a new subscription.

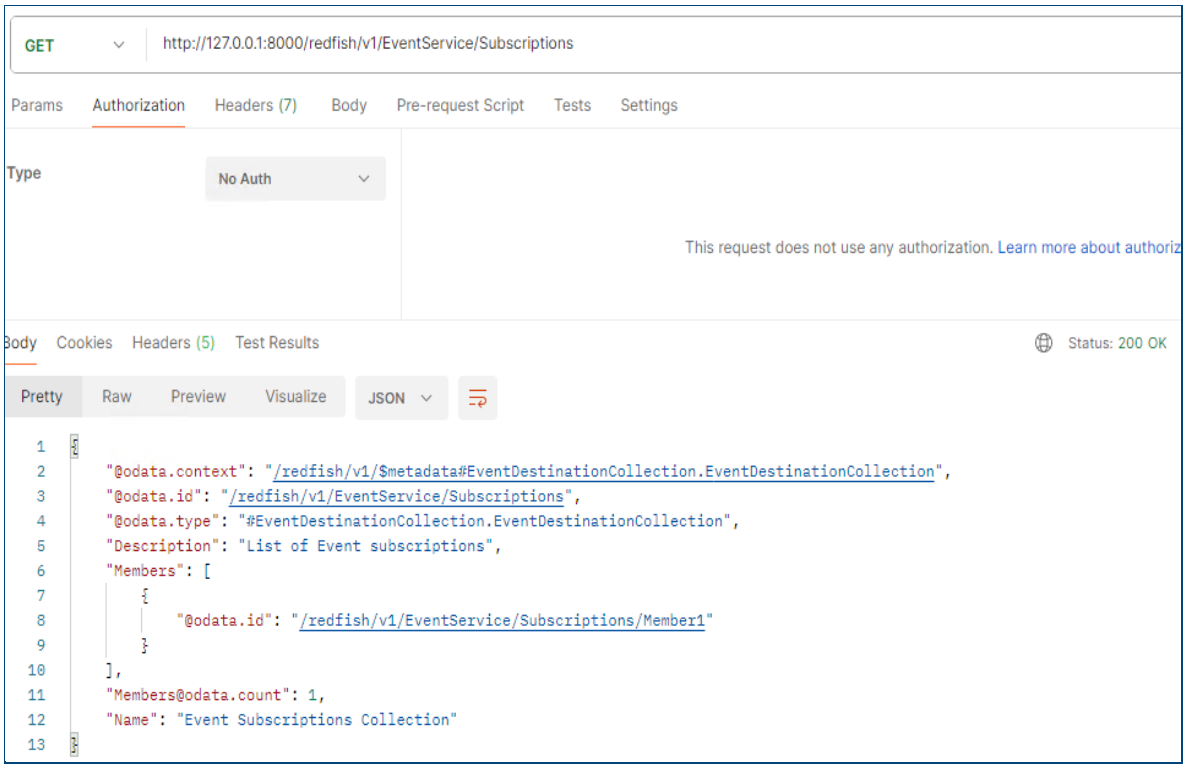

Figure 26 Get subscription information

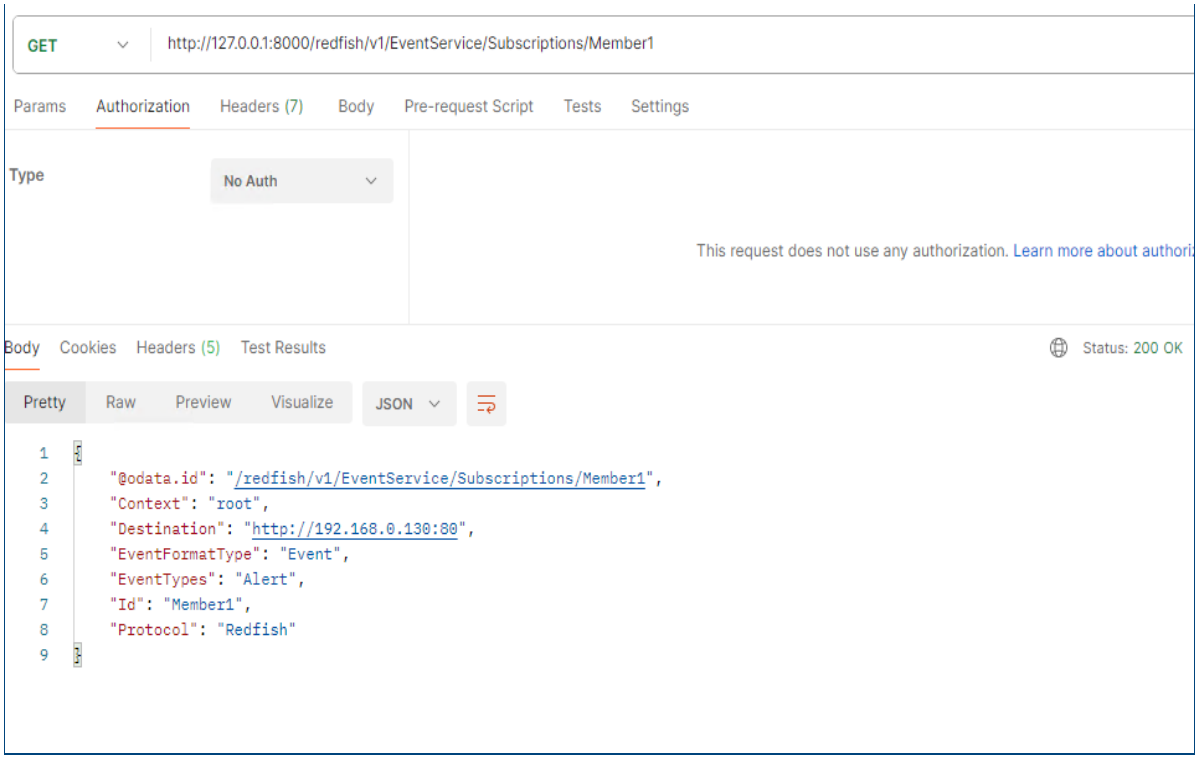

Figure 27 Get subscription information

3.4 Start Redfish event listener

You can start the Redfish event listener. If you observe any issues, see the relevant Release Notes on the GitHub page.

```
C:\Python310>python RedfishEventListener_v1.py
Redfish Event Listener v1.1.3
ServerIP:: http://127.0.0.1:8000
UserName:: root
Attempt 1 of /redfish/v1/
Response Time for GET to /redfish/v1/: 0.021674100076779723 seconds.
Attempt 1 of /redfish/v1/SessionService/Sessions
Response Time for POST to /redfish/v1/SessionService/Sessions:
0.009719900088384748 seconds.
Login returned code 204:
Attempt 1 of /redfish/v1/
Response Time for GET to /redfish/v1/: 0.010551500134170055 seconds.
Attempt 1 of /redfish/v1/EventService
Response Time for GET to /redfish/v1/EventService: 0.011112900217995048 seconds.
Attempt 1 of /redfish/v1/EventService/Subscriptions
Response Time for POST to /redfish/v1/EventService/Subscriptions:
0.009794200072064996 seconds.
Subscription is successful for http://127.0.0.1:8000,
/redfish/v1/EventService/Subscriptions/Member2
The service responded with invalid JSON at URI
/redfish/v1/EventService/Subscriptions

Continuing with Listener.
Listening on 0.0.0.0:80 via HTTP
Press Ctrl-C to close program
..
```

3.5 Submit a test event

Submit a test event using valid iDRAC message ID from the message registry. To get supported message IDs, run GET on URI `redfish/v1/Registries/Messages/EEMIRegistry`.

Note—In the following example, the body properties that are listed must submit a test event. Else the Error 400 message is displayed. To get the property values, enter random strings for testing.

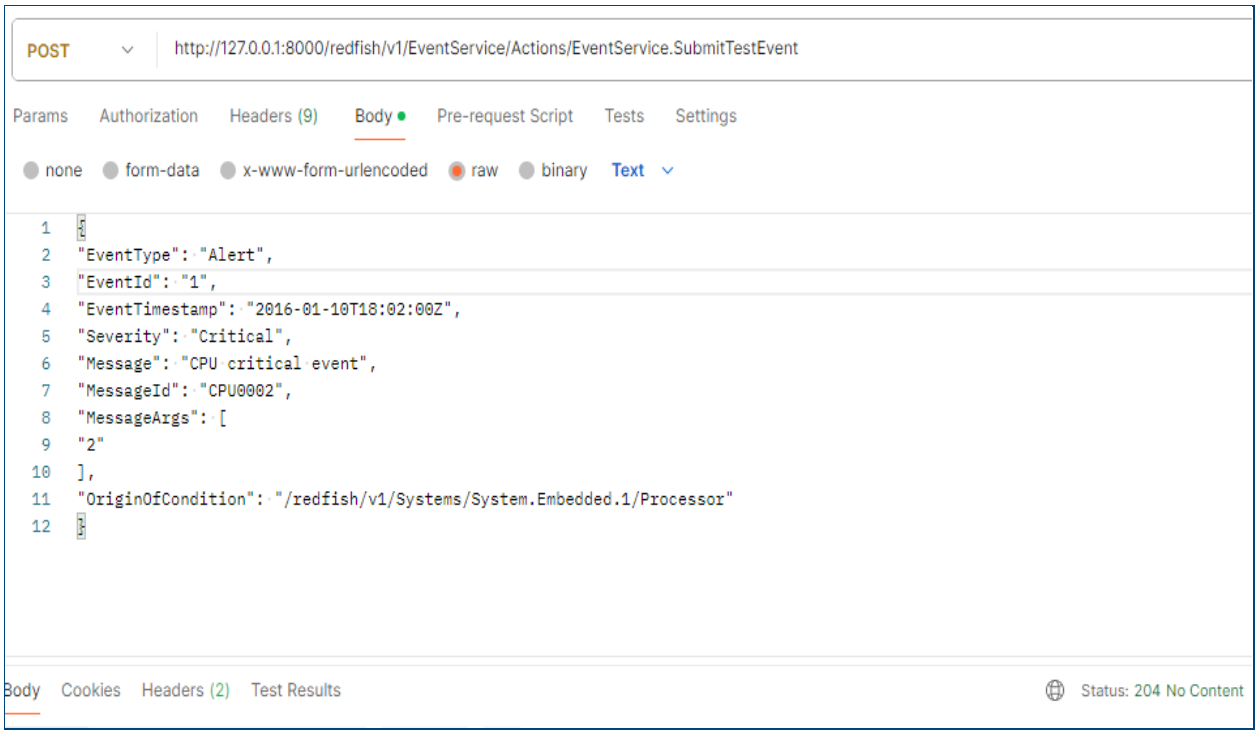

Figure 28 Submit a test event

3.6 Confirm that Redfish event listener receives event

Check the Redfish event listener and confirm that the test event that you submitted is reported.

```
C:\Python310>python RedfishEventListener_v1.py
Redfish Event Listener v1.1.3
ServerIP:: http://127.0.0.1:8000
UserName:: root
Attempt 1 of /redfish/v1/
Response Time for GET to /redfish/v1/: 0.021674100076779723 seconds.
Attempt 1 of /redfish/v1/SessionService/Sessions
Response Time for POST to /redfish/v1/SessionService/Sessions:
0.009719900088384748 seconds.
Login returned code 204:
Attempt 1 of /redfish/v1/
Response Time for GET to /redfish/v1/: 0.010551500134170055 seconds.
Attempt 1 of /redfish/v1/EventService
Response Time for GET to /redfish/v1/EventService: 0.011112900217995048 seconds.
Attempt 1 of /redfish/v1/EventService/Subscriptions
Response Time for POST to /redfish/v1/EventService/Subscriptions:
0.009794200072064996 seconds.
Subscription is successful for http://127.0.0.1:8000,
/redfish/v1/EventService/Subscriptions/Member2
The service responded with invalid JSON at URI
/redfish/v1/EventService/Subscriptions
```

```
Continuing with Listener.
Listening on 0.0.0.0:80 via HTTP
Press Ctrl-C to close program
.......................
Socket connected::
headers: IOrderedDict([('Host', '192.168.0.130'), ('User-Agent', 'python-
requests/2.27.1'), ('Accept-Encoding', 'gzip, deflate'), ('Accept', '*/*'),
('Connection', 'keep-alive'), ('Content-Type', 'application/json'), ('Content-
Length', '374')])


bodydata: {"@odata.type": "#Event.v1_2_1.Event", "Name": "Test Event", "Id":
"1", "Events": [{"EventType": "Alert", "EventId": "1", "EventTimestamp": "2016-
01-10T18:02:00Z", "Severity": "Critical", "Message": "CPU critical event",
"MessageId": "CPU0002", "MessageArgs": ["2"], "OriginOfCondition": {"@odata.id":
"/redfish/v1/Systems/System.Embedded.1/Processor"}}], "Context": "root"}


Server IP Address is 192.168.0.130
Server PORT number is 54633
Listener IP is 192.168.0.130


Context (root) does not match with the server (Public).
Event Counter for Host 192.168.0.130 = 1


.....................
```

4 Tips or limitations

- There is no authentication required to run the simulated Redfish calls.
- When a Redfish call is run, the Redfish mockup server session logs request information.
- The User-Agent property logs data about the interface that invoked the Redfish request.
- Redfish mockup server logs for PATCH operations also return the updated resource information that displays the updated property value.
- Restart Redfish mockup server to clear the settings applied. For example, run delete VD operation which deletes the VD from storage inventory, restart Redfish mockup server for the VD to show up again in storage inventory).
- For any issues, concerns, or new iDRAC Redfish mockup client suggestions, you can submit at: iDRAC-Redfish-Scripting issues.
- A Grequests module is required to start Redfish mockup server (This module is not native to Python)
- After you create an iDRAC user session, get the user or session data that reports the password in clear text. This operation is not performed on iDRAC but on the mockup server.
- Server-Sent Events (SSE) streaming does not support Redfish events. Use Redfish Event Listener.
- The Redfish mockup server does not support the firmware update operation (`SimpleUpdate`, `MultipartUpload`).

5 Resource links

- Dell Redfish API Guide
- Dell iDRAC Redfish FAQ Whitepaper
- Dell iDRAC User Interface to Redfish Mapping Whitepaper
- Dell iDRAC RACADM to Redfish Mapping Whitepaper
- DMTF Redfish