Redfish

Reference Guide

2024 May_v1.0

Preface

Copyright

Version 1.0

Disclaimer

Information contained in this document is furnished by MITAC COMPUTING TECHNOLOGY CORPORATION and has been reviewed for accuracy and reliability prior to printing. MITAC assumes no liability whatsoever, and disclaims any express or implied warranty, relating to sale and/or use of TYAN, loss of data or other malady resulting from errors or inaccuracies of information contained in this document.

Trademark Recognition

All registered and unregistered trademarks and company names contained in this manual are property of their respective owners including, but not limited to the following.

TYAN® is a trademark of MITAC COMPUTING TECHNOLOGY CORPORATION.
Intel® is a trademark of Intel® Corporation.
AMI®, AMIBIOS® and combinations thereof are trademarks of AMI Technologies.
Microsoft®, Windows® are trademarks of Microsoft Corporation.
IBM®, PC®, AT® and PS/2® are trademarks of IBM Corporation.
Winbond® is a trademark of Winbond Electronics Corporation.

Table of Contents

Document Revision

Date	Version	Document Update
May	1.0	Released to public.

Chapter 1. Introduction

Redfish is a software solution developed to be fully compliant with DMTF Redfish specification. It allows users to browse physical resources at the chassis and system level through an intuitive web-based user interface. Redfish is web based management protocol. It is built upon Representational State Transfer (REST) which is itself based on HTTP 1.1 protocol. Redfish improves the scalability and help customers to integrate with existing tools.

Redfish is a hypermedia API with a small set of defined URI's. This document provides the API list supported by the Redfish Server and the HTTP methods for each URL in addition to a detailed explanation of the request and JSON response properties. As Redfish is built on OData specification, it discusses the OData properties and the OData identifier for the resources.

Redfish provides information categorized under specific resource end point. The redfish clients allows to utilize the end points using following HTTP methods:

- GET
- POST
- PATCH
- PUT
- DELETE

Not all end-points support all these operations. When not supported it must send back 405 HTTP Status. Such details on the operations are provided by the Redfish JSON Schema.

Redfish Server follows DSP0266 1.11 Specification and Redfish Schema 2019.2.

Chapter 2. HTTP Request Methods

The following HTTP methods are used to implement different actions:
HTTP Request Methods

Method	Action	Description
GET	Read Requests	The method requests a representation of a specified resource. The representation can be either a single resource or a collection.
DELETE	Delete	The method removes a resource.
PATCH	Update	The method applies partial modifications to a resource.
POST	Create	The method creates a new resource. This request is submitted to the resource collection in which the new resource is meant to belong.
POST	Actions	The method initiates operations on the object (Actions). The POST operation may not be idempotent.
PUT	Replace	The method completely replaces a resource. Any properties omittedfrom the body of the request are reset to their default value.

2.1 Responses

There are four types of responses:
Responses

Response	Description
Metadata	Resources and types are exposed by the service to genericclients.
Resource Responses	An individual resource is displayed in JSON format.
Resource Collection	JSON representation of a collection of resources.
Error	Top-level JSON response providing additional information in thecase of an HTTP error.

2.2 HTTP Status Code

HTTP Status Code

Status Code	Description
200	OK
201	Created
202	Accepted
204	No Content
301	Moved Permanently
302	Found
304	Not Modified
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
405	Method Not Allowed
406	Not Acceptable
409	Conflict
410	Gone
411	Length Required
412	Precondition Failed
415	Unsupported Media Type
500	Internal Server Error
501	Not Implemented
503	Service Unavailable

Chapter 3. Using RESTful APIs

To receive API responses through programming, install Postman or any other Rest API client application(s).

3.1 Authentication

You are required to have authentication to access certain resources. Redfish offers two methods for usersto access Redfish URLs: “basic authentication” and “Redfish session login authentication.” The Service does not require you to create a session when Basic Authentication is used.

3.1.1 Basic Authentication

HTTP basic authentication uses compliant TLS connections to transport the data between any third-partyauthentication service and clients. Use local BMC credentials or remote protocols like LDAP, Active Directory, or RADIUS to log in with basic authentication.

3.1.2 Session Management

You can use session management to implement authentication. This includes orphaned session timeoutsand several simultaneous open sessions. You can create up to 16 sessions.
Step 1: You can post the following username/password information in the payload field, which will create anew session.

```
{
"UserName": "<username>",
"Password": "<password>"
}
```

The user will receive the “201” message code with the X-AUTH token created.

Session lifetime: For Redfish sessions, as long as you send requests for the session within the session timeout period, the session will remain open and the session authentication token will remain valid. If the session times out, the session will be automatically terminated.
According to Redfish specification, a user can define session time from 30 to 86400 seconds. If you are not active in the defined time frame, the token will be rendered invalid. You can always patch the “SessionTimeout” value if needed

Session termination or logout: A Redfish session is terminated when you log out. This is accomplishedby performing the DELETE method on the session resource identified by the link returned in the location header either when the session is created or if the Session ID is returned in the response data. Using theDELETE method on a session by specifying the session resource ID allows an administrator with sufficientprivilege to terminate other users’ sessions from a different session.

4.1 Firmware Inventory

This resource shall be used to represent a collection of firmware inventory.

URI: /redfish/v1/UpdateService/FirmwareInventory
Method: GET
Example:
After the request is sent to the target, such as S8056, the response status is 200 with response body in JSON format if the request is successful.

GET https://{{bmc}}/redfish/v1/UpdateService/FirmwareInventory Send

Params Auth Headers (7) Body Pre-req. Tests Settings Cookies

Body 200 OK 74 ms 1.1 KB Save as example

Pretty Raw Preview Visualize JSON


```
{
    "@odata.context": "/redfish/v1/$metadata#SoftwareInventoryCollection.Softwa
    "@odata.etag": "\"1686483272\"",
    "@odata.id": "/redfish/v1/UpdateService/FirmwareInventory",
    "@odata.type": "#SoftwareInventoryCollection.SoftwareInventoryCollection",
    "Description": "Collection of Firmware Inventory resources available to the
    "Members": [
        {
            "@odata.id": "/redfish/v1/UpdateService/FirmwareInventory/BIOS"
        },
        {
            "@odata.id": "/redfish/v1/UpdateService/FirmwareInventory/BMC"
        },
        {
            "@odata.id": "/redfish/v1/UpdateService/FirmwareInventory/CPLD"
        }
    ],
    "Members@odata.count": 3,
    "Name": "Firmware Inventory Collection"
}
```

4.2 Updating BIOS Firmware

Perform BIOS image update.

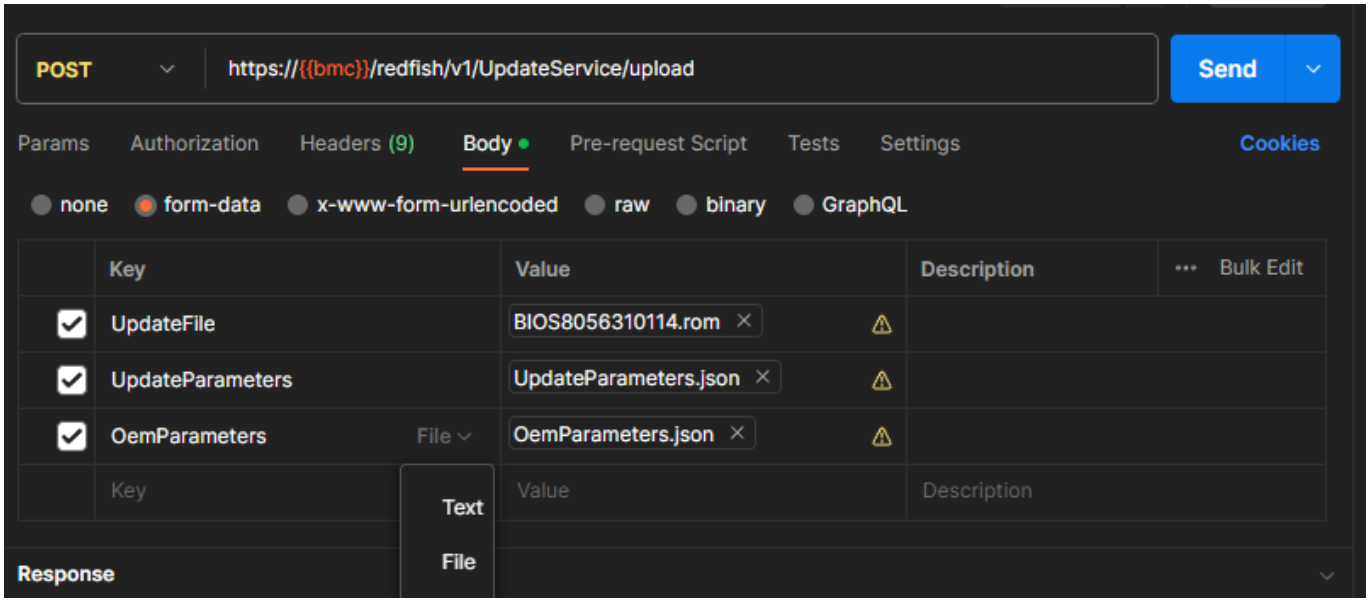
URI: /redfish/v1/UpdateService/upload
Method: POST
Example:
Prepare files
UpdateParameters.json
DMTF defined standard parameters in json format.

```
{
   "Targets" :[
      "/redfish/v1/UpdateService/FirmwareInventory/BIOS"
   ]
}
```

OemParameters.json
OEM parameters in json format.

```
{
   "ImageType" : "BIOS"
}
```

Update BIOS image.
The request body should be in the format of multipart/form-data and contain UpdateFile, UpdateParameters and OemParameters.
The value of UpdateFile is the path of the binary image for update.
The value of UpdateParameters is the path of UpdateParameters.json
The value of OemParameters is the path of OemParameters.json.



After the request is sent to the target, such as S8056, the response status is 202 with response body in JSON format if the request is successful. The POST creates TaskService with Task ID. You can check the progress of the update task.

POST https://{{bmc}}/redfish/v1/UpdateService/upload Send

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL

Key	Value	Description	Bulk Edit

Body Cookies Headers (14) Test Results 202 Accepted 2.62 s 1.12 KB Save as example

Pretty Raw Preview Visualize JSON


```
{
    "@odata.type": "#UpdateService.v1_6_0.UpdateService",
    "Messages": [
        {
            "@odata.type": "#Message.v1_0_8.Message",
            "Message": "A new task /redfish/v1/TaskService/Tasks/1 was created.",
            "MessageArgs": [
                "/redfish/v1/TaskService/Tasks/1"
            ],
            "MessageId": "Task.1.0.New",
            "Resolution": "None",
            "Severity": "OK"
        },
        {
            "@odata.type": "#Message.v1_0_8.Message",
            "Message": "Device is preparing flash firmware for action UpdateService.MultipartPush.",
            "MessageArgs": [
                "UpdateService.MultipartPush"
            ],
            "MessageId": "UpdateService.1.0.PrepareUpdate",
            "Resolution": "None",
            "Severity": "OK"
        }
    ]
}
```

Find the progress of BIOS image update.
After the BIOS request is sent, the progress of update can be checked by accessing /redfish/v1/TaskService/Tasks/1.

GET https://{{bmc}}/redfish/v1/TaskService/Tasks/1 Send

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL

Key	Value	Description	... Bulk Edit

Body Cookies Headers (14) Test Results 200 OK 22 ms 1.41 KB Save as example

Pretty Raw Preview Visualize JSON


```
2    "@odata.context": "/redfish/v1/$metadata#Task.Task",
3    "@odata.etag": "\"1686484420\"",
4    "@odata.id": "/redfish/v1/TaskService/Tasks/1",
5    "@odata.type": "#Task.v1_4_2.Task",
6    "Description": "Task for Update Service Task",
7    "EndTime": "2023-06-11T19:53:50+08:00",
8    "Id": "1",
9    "Messages": [
10       {
11           "@odata.type": "#Message.v1_0_8.Message",
12           "Message": "Task /redfish/v1/UpdateService/upload has completed.",
13           "MessageArgs": [
14               "/redfish/v1/UpdateService/upload"
15           ],
16           "MessageId": "Task.1.0.Completed",
17           "Resolution": "None",
18           "Severity": "OK"
19       },
20       {
21           "@odata.type": "#Message.v1_0_8.Message",
22           "Message": "Action /redfish/v1/UpdateService/upload firmware update is completed.",
23           "MessageArgs": [
24               "/redfish/v1/UpdateService/upload"
25           ],
26           "MessageId": "UpdateService.1.0.FirmwareUpdateCompleted",
27           "Resolution": "None",
28           "Severity": "OK"
29       }
30   ],
31   "Name": "Update Service Task",
32   "StartTime": "2023-06-11T19:53:40+08:00",
33   "TaskState": "Completed",
34   "TaskStatus": "OK"
35 }
```

4.3 Updating BMC Firmware

Perform BMC firmware update.

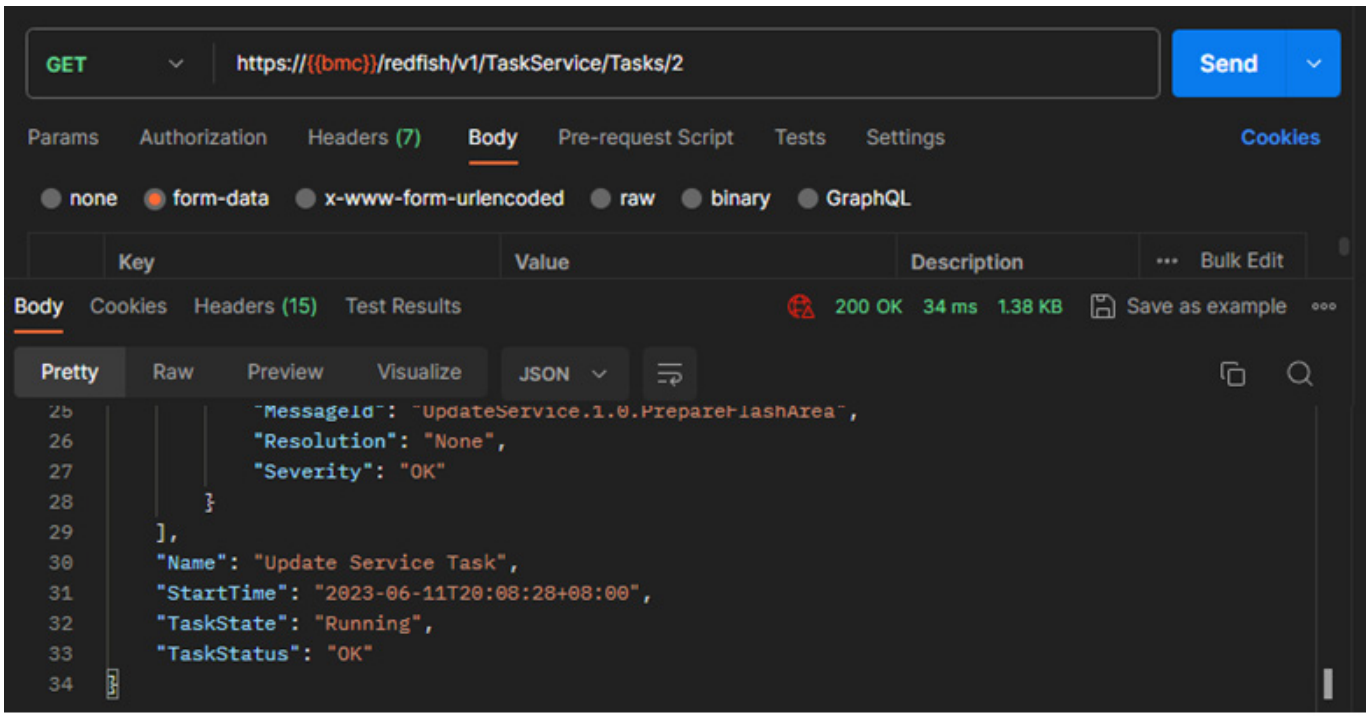
URI: /redfish/v1/UpdateService/upload
Method: POST
Example:
Prepare files
UpdateParameters.json
DMTF defined standard parameters in json format.

```
{
  "Targets": [
    "/redfish/v1/UpdateService/FirmwareInventory/BMC"
  ]
}
```

OemParameters.json
AMI OEM parameters in json format.

```
{
  "ImageType": "BMC"
}
```

Update BMC image. Please reference the second step in the example of "5.2 Updating BIOS". Find the progress of BMC firmware update. Please reference the third step in the example of "5.2 Updating BIOS". BMC will reboot after image update, so you can only query the update progress by accessing /redfish/v1/TaskService/Tasks/2 before BMC reboots.

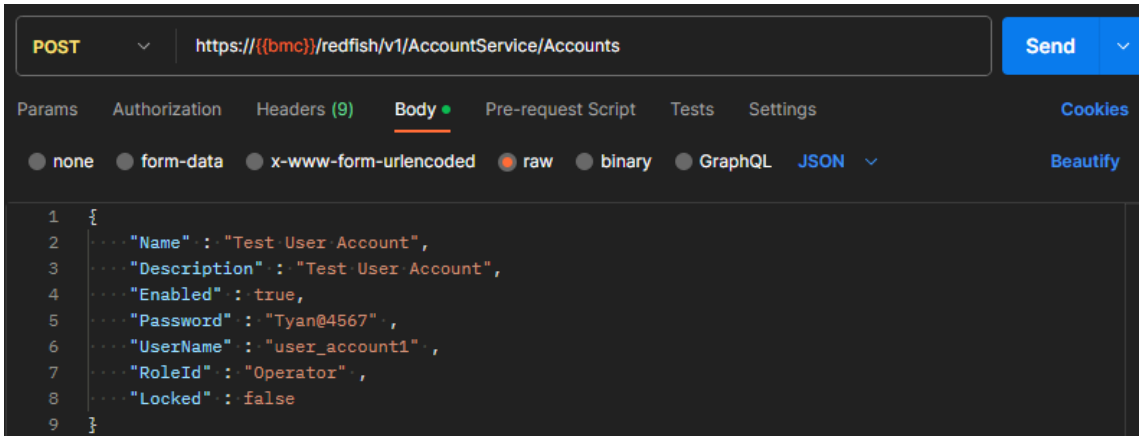


Chapter 5. Account Service

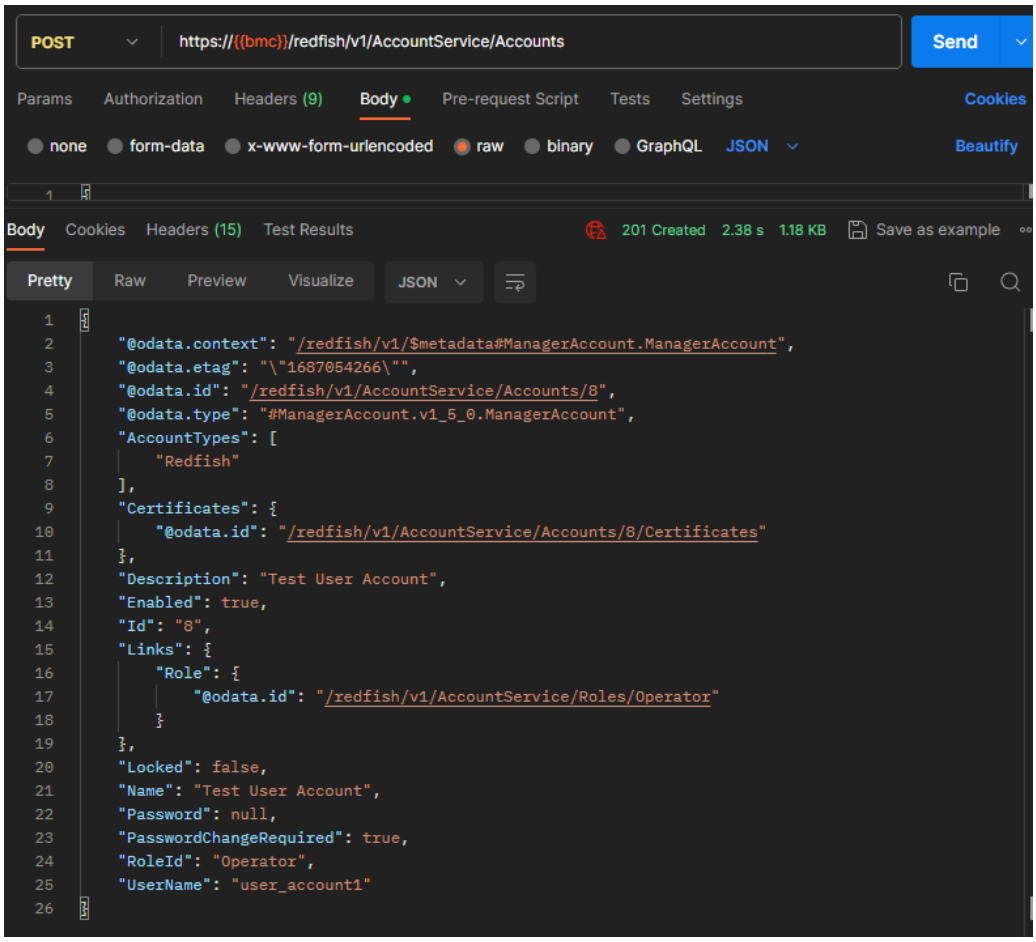
You can perform the following operations under /redfish/v1/AccountService.
Available Methods: GET, POST, PATCH, and DELETE.

5.1 Creating a User

URI: /redfish/v1/AccountService/Accounts
Method: POST
Example:
Edit the request content in JSON format from Body.



After the request is sent to the target, such as S8056, the response status is 201 with body in JSON format if the request is successful.



5.2 Configuring User Lockout

URI: /redfish/v1/AccountService
Method: PATCH
Example:
Edit the request content in JSON format from Body.
Account Lockout Threshold:

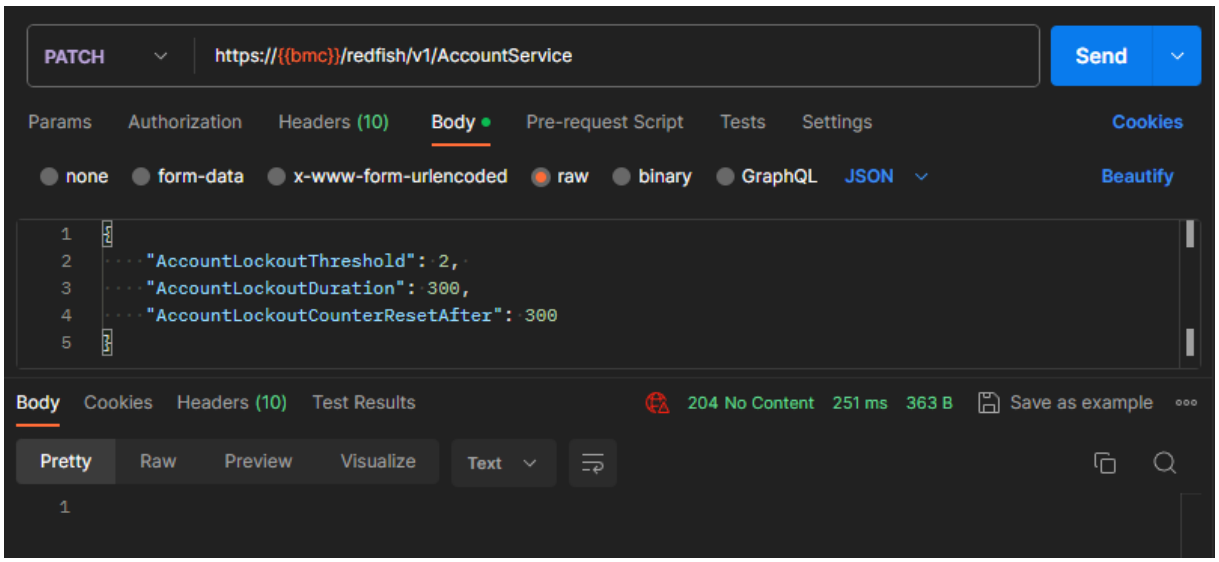
The number of failed login attempts before a user account is locked for a specified duration. (0=never locked).
Minimum Value: 0
Maximum Value: 100
Default Value: 5

Account Lockout Duration:
This property shall reference the period of time in seconds that an account is locked after the number of failed login attempts reaches the threshold referenced by Account Lockout Threshold, within the window of time referenced by Account Lockout Counter Reset After. The value shall be greater than or equal to the value of Account Lockout Reset After. If set to 0, no lockout shall occur.

Minimum Value: 0
Maximum Value: 10000
Default Value: 30

Account Lockout Counter Reset After:
This property shall reference the threshold of time in seconds from the last failed login attempt at which point the Account Lockout Threshold counter (that counts number of failed login attempts) is reset back to zero (at which point Account Lockout Threshold failures would be required before the account is locked). This value shall be less than or equal to Account Lockout Duration. The threshold counter also resets to zero after each successful login.

Minimum Value: 0
Maximum Value: 10000
Default Value: 30



After the request is sent to the target, such as S8056, the response status is 204 without response body if the request is successful.

5.3 Active Directory

URI: /redfish/v1/AccountService
METHOD: PATCH
Example:
Edit the request content in JSON format from Body.

PATCH https://{{bmc}}/redfish/v1/AccountService Send

Params Authorization Headers (10) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify

```
1  {
2      "ActiveDirectory" : {
3          "Authentication" : {
4              "Username" : "AD1",
5              "Password" : "AD@123",
6              "Oem" : {
7                  "Ami" : {
8                      "DomainName" : "abc123.com",
9                      "DomainControllerServerAddr1": "10.0.1.23",
10                     "DomainControllerServerAddr2": "",
11                     "DomainControllerServerAddr3": ""
12                 }
13             }
14         },
15         "ServiceEnabled" :true
16     }
17 }
```

After the request is sent to the target, such as S8056, the response status is 202 with response body in JSON format if the request is successful.

PATCH https://{{bmc}}/redfish/v1/AccountService

Params Authorization Headers (10) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

Body Cookies Headers (15) Test Results 202 Accepted 242 ms 833 B Sav

Pretty Raw Preview Visualize JSON

```
1  {
2      "@odata.context": "/redfish/v1/$metadata#Task.Task",
3      "@odata.id": "/redfish/v1/TaskService/Tasks/2",
4      "@odata.type": "#Task.v1_4_2.Task",
5      "Description": "Task for AccountService ActiveDirectory",
6      "Id": "2",
7      "Name": "AccountService ActiveDirectory",
```

5.4 LDAP

URI: /redfish/v1/AccountService
Method: PATCH
Example:
Edit the request content in JSON format from Body.

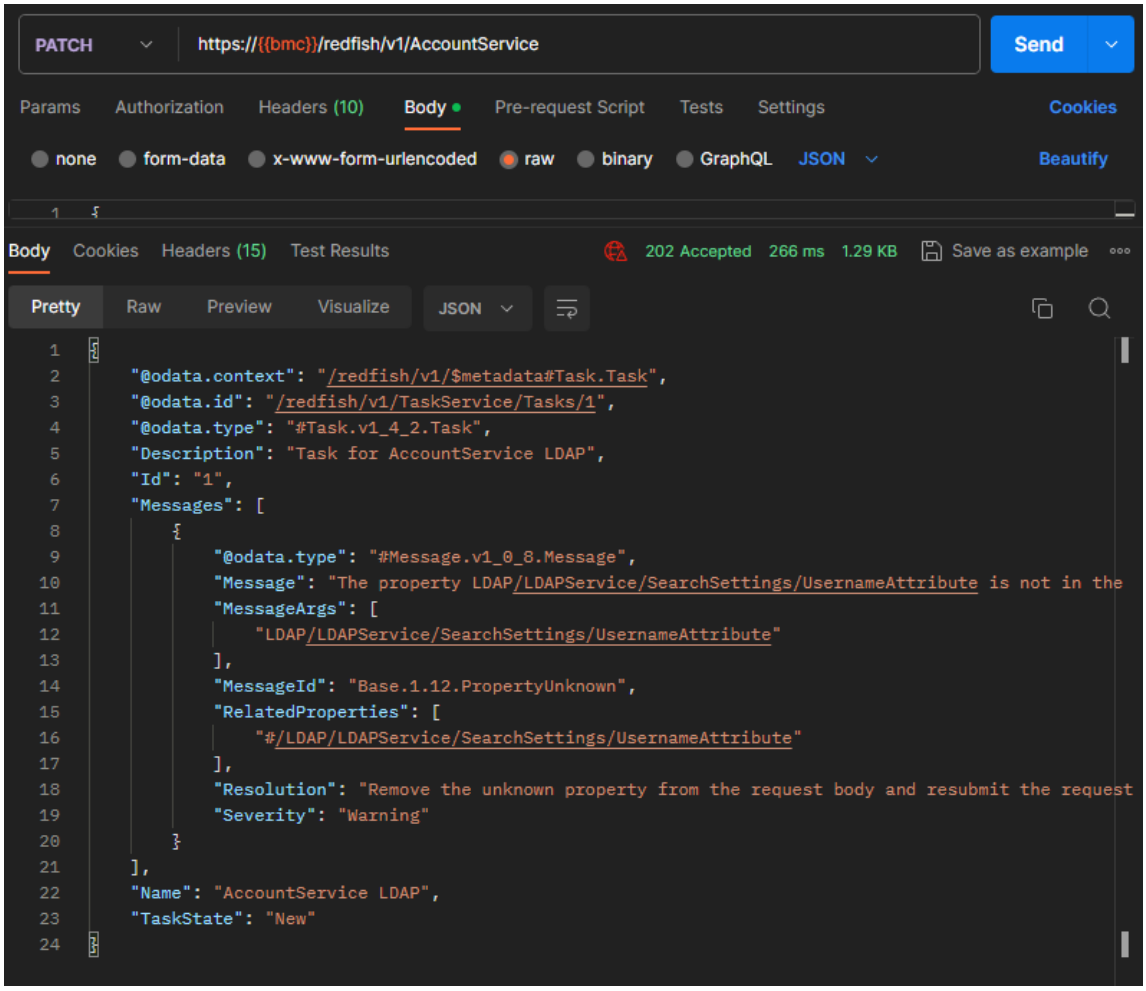
PATCH https://{{bmc}}/redfish/v1/AccountService Send

Params Authorization Headers (10) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify

```
{
    "LDAP" : {
        "Authentication" : {
            "Username" : "cn=admin,dc=coretesting,dc=com" ,
            "Password" : "ubuntu" ,
            "Oem" : {
                "Ami" : {
                    "EncryptionType" : "NoEncryption" ,
                    "CommonNameType" : "IPAddress"
                }
            }
        },
        "LDAPService" : {
            "SearchSettings" : {
                "BaseDistinguishedNames" : [
                    "dc=coretesting,dc=com"
                ],
                "UsernameAttribute" : "cn"
            }
        },
        "ServiceAddresses" : [
            "10.0.122.61:389"
        ],
        "ServiceEnabled" : true
    }
}
```

After the request is sent to the target, such as S8056, the response status is 202 with response body in JSON format if the request is successful.



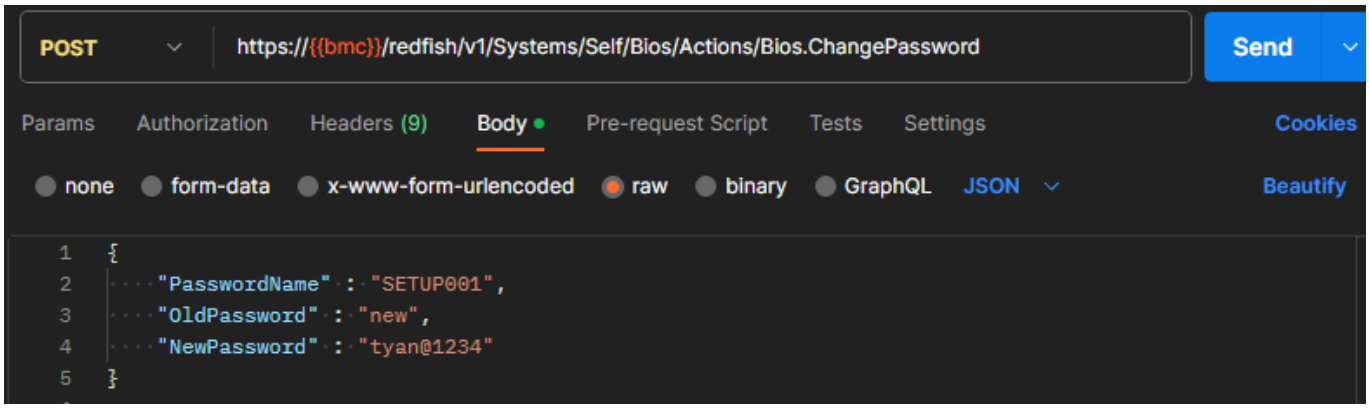
Chapter 6: BIOS Configuration

The properties of BIOS can be configured by using the BIOS redfish APIs.

Note: All the changes in BIOS attributes need a system reboot to take effect.

6.1 Changing a Password

URI: /redfish/v1/Systems/Self/Bios/Actions/Bios.ChangePassword
Method: POST
Example:
Edit the request content in JSON format from Body.

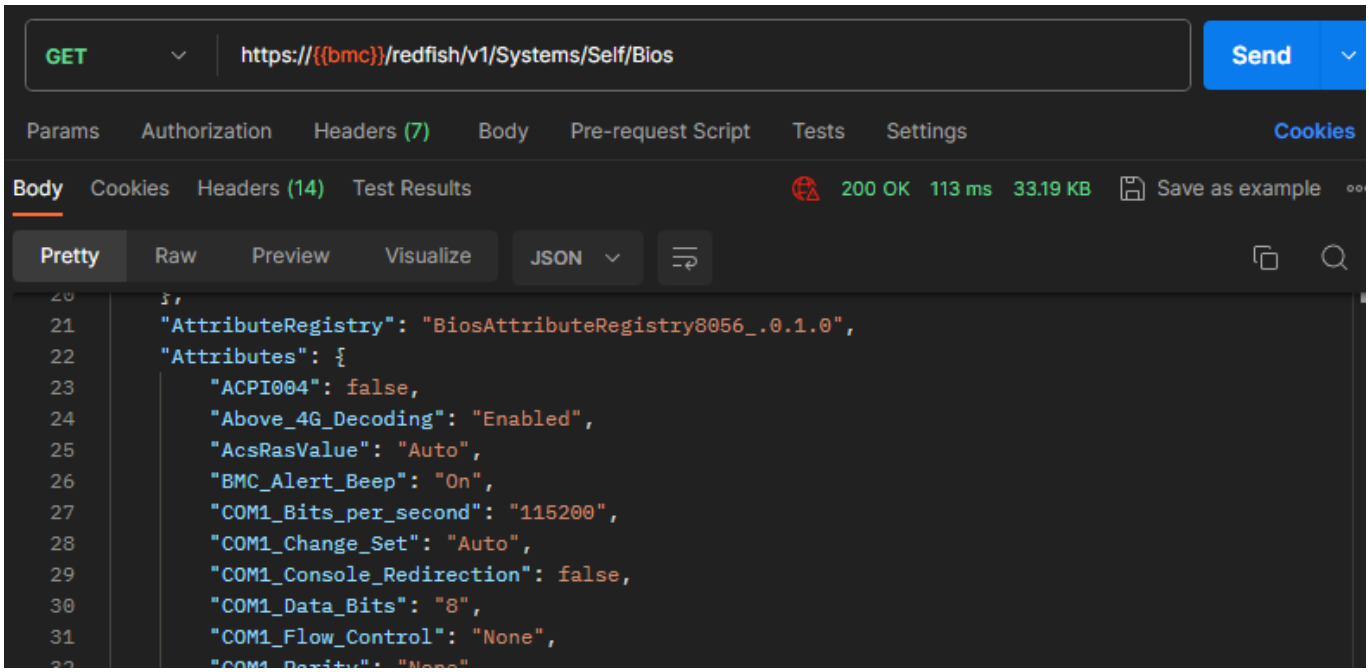


After the request is sent to the target, such as S8056, the response status is 204 with on body if the reques is successful.

After rebooting the machine, you need to give the new password to enter the BIOS setup menu.

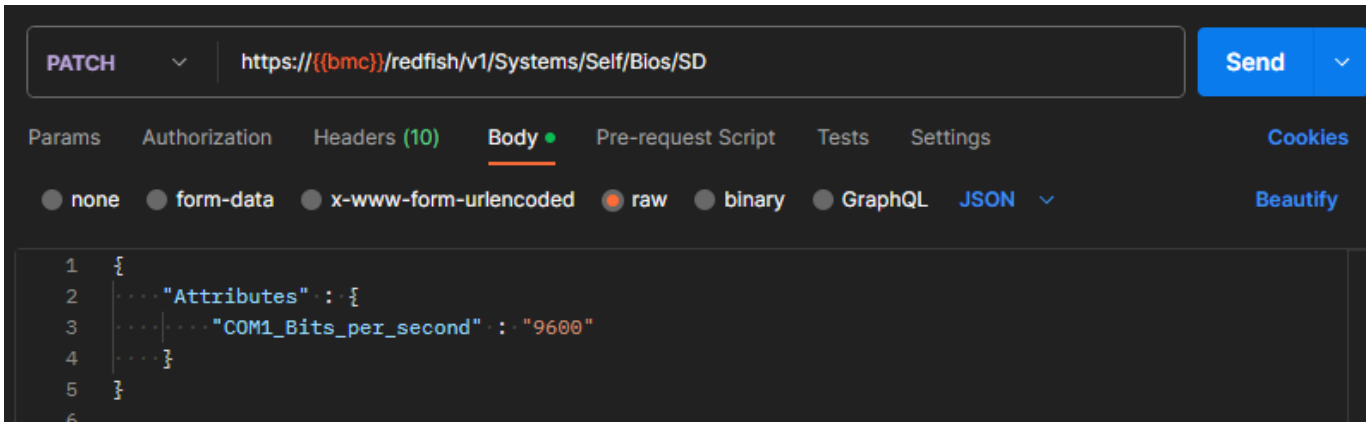
6.2 Configuring BIOS over Redfish

/redfish/v1/Registries/BiosAttributeRegistry8056_.en-US.0.1.0.json is the Bios Attribute Registry containing the list of supported attributes and its dependencies.
URI: /redfish/v1/Systems/Self/Bios
Method: GET
Example:
After the request is sent to the target, such as S8056, the response status is 200 with body in JSON format if the request is.



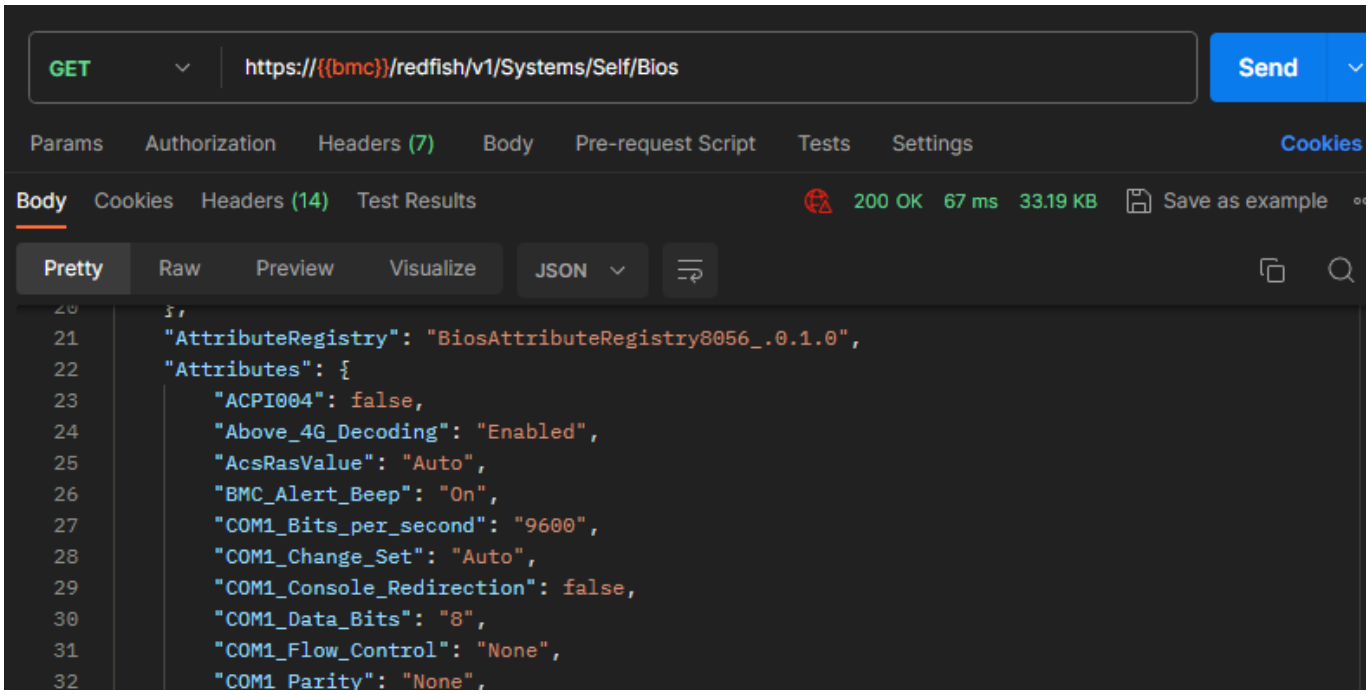
6.2.1 Modifying BIOS Attributes

URI: /redfish/v1/Systems/Self/Bios/SD
Method: PATCH
Example:
Edit the request content in JSON format from Body.



After the request is sent to the target, such as S8056, the response status is 204 without response body if the request is successful.

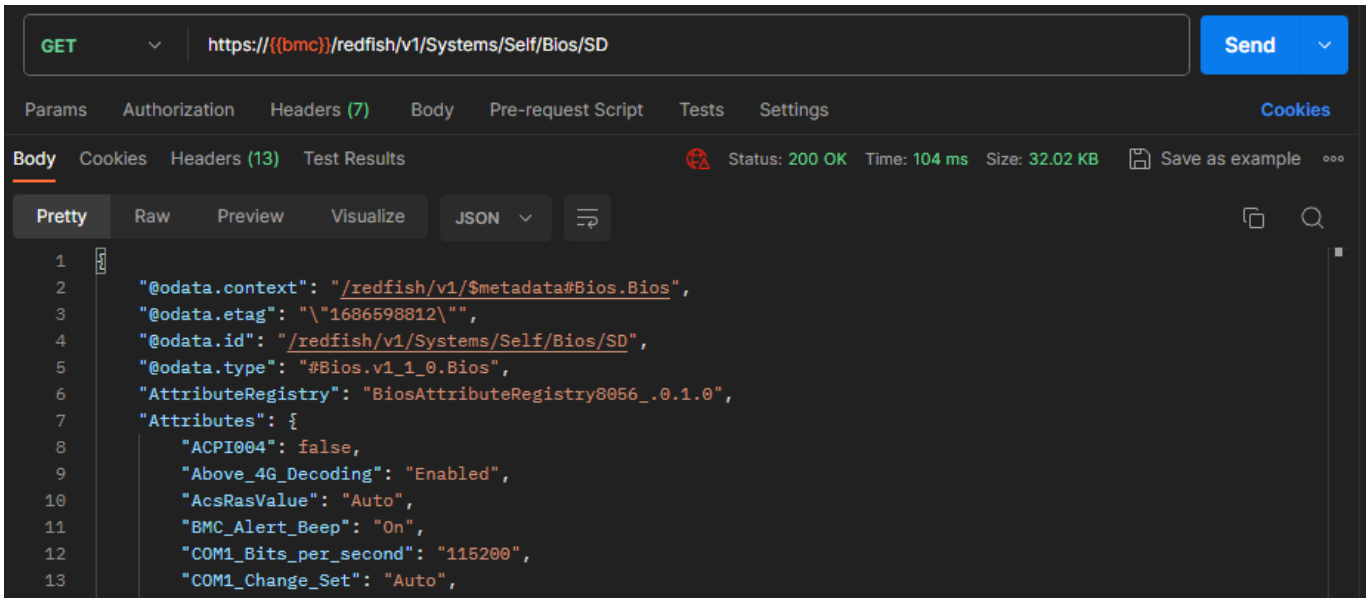
Then, you can check the result by querying the BIOS attributes.



6.2.2 Viewing Pending Settings

You can find all the pending settings after PATCH.

URI: /redfish/v1/Systems/Self/Bios/SD
Method: GET
Example:
After the request is sent to the target, such as S8056, the response status is 200 with body in JSON format if the request is successful.



6.3 Reset BIOS

POST a reset of the BIOS attributes to default values. After POST, you need to reset the system to apply values to BIOS.

URI: /redfish/v1/Systems/Self/Bios/Actions/Bios.ResetBios

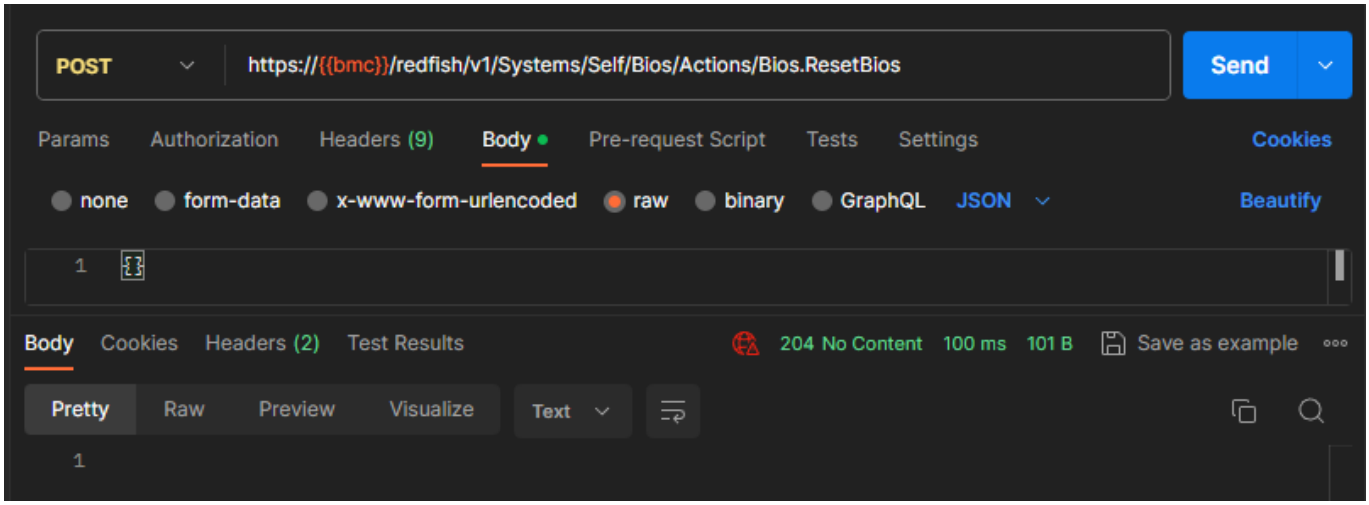
Method: POST

Example:

Example for POST request body (JSON format):

```
{
}
```

Edit the request content in JSON format from Body.



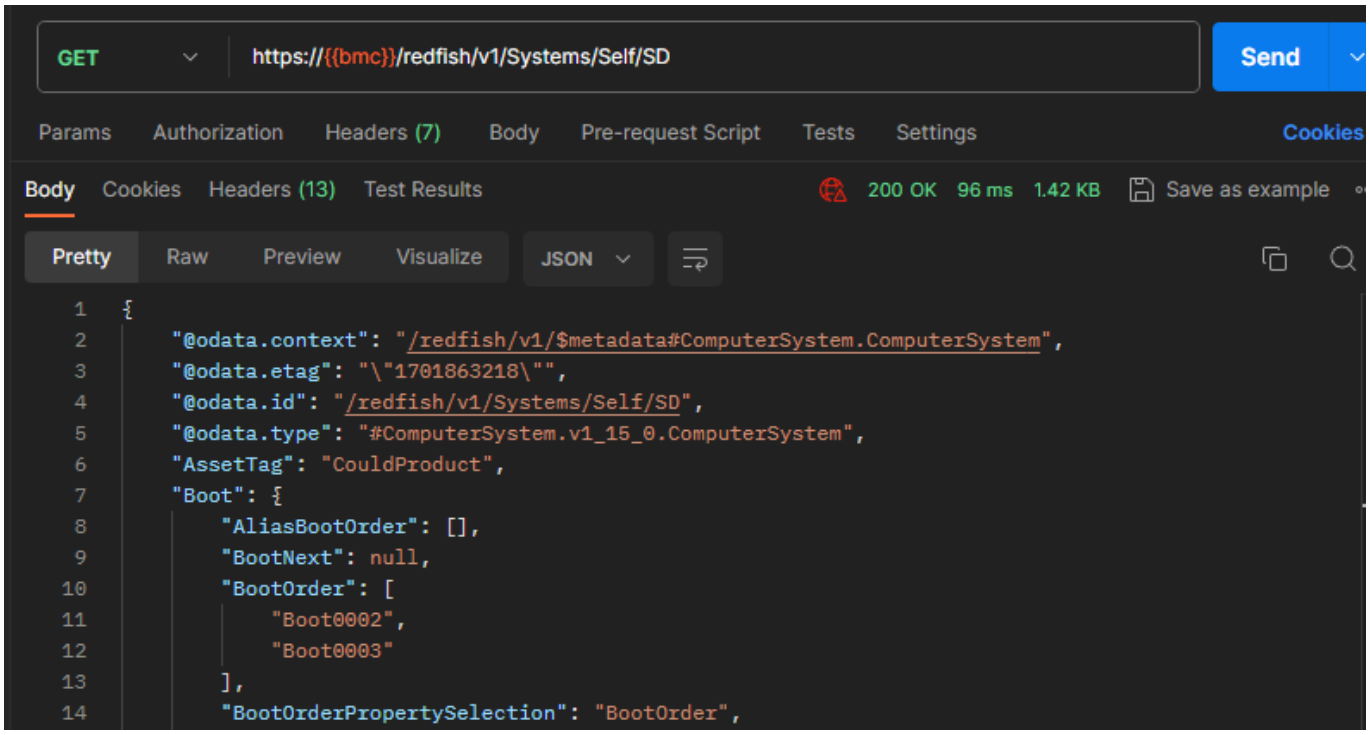
After the request is sent to the target, such as S8056, the response status is 204 without response body in JSON format if the request is successful.

6.4 Boot Options

6.4.1 Configuring the Boot Order in System BIOS

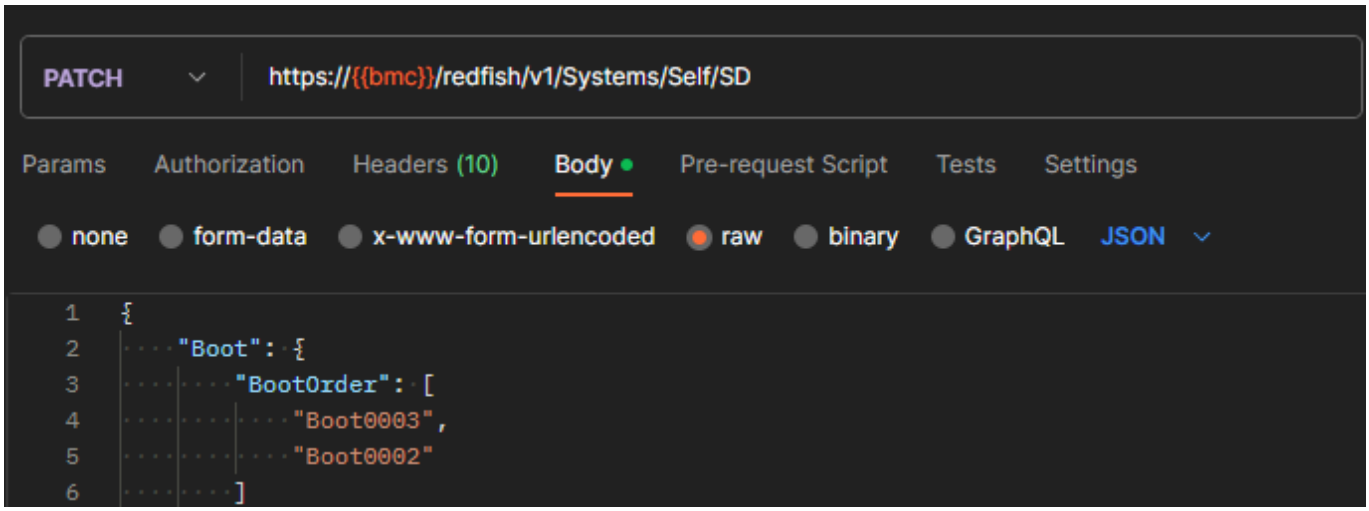
6.4.1.1 Changing the Boot Order Getting the Current Boot Order

URI: /redfish/v1/Systems/Self/SD
Method: GET
Example:
After the request is sent to the target, such as S8056, the response status is 200 without response body if the request is successful.



6.4.1.2 Changing the Current Boot Order

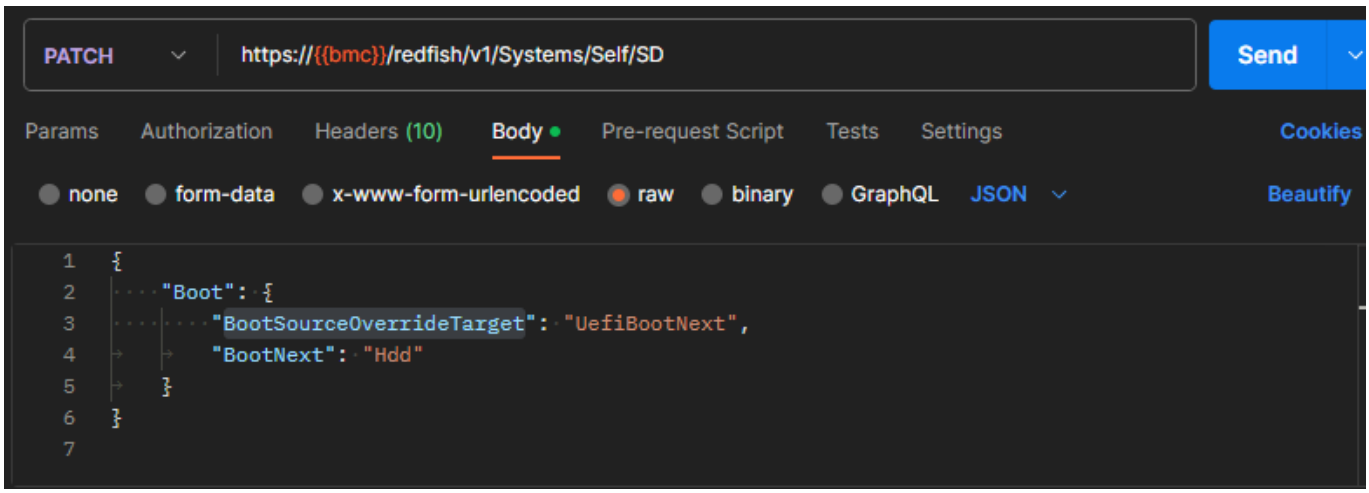
URI: /redfish/v1/Systems/Self/SD
Method: PATCH
Example:
Edit the request content in JSON format from Body.



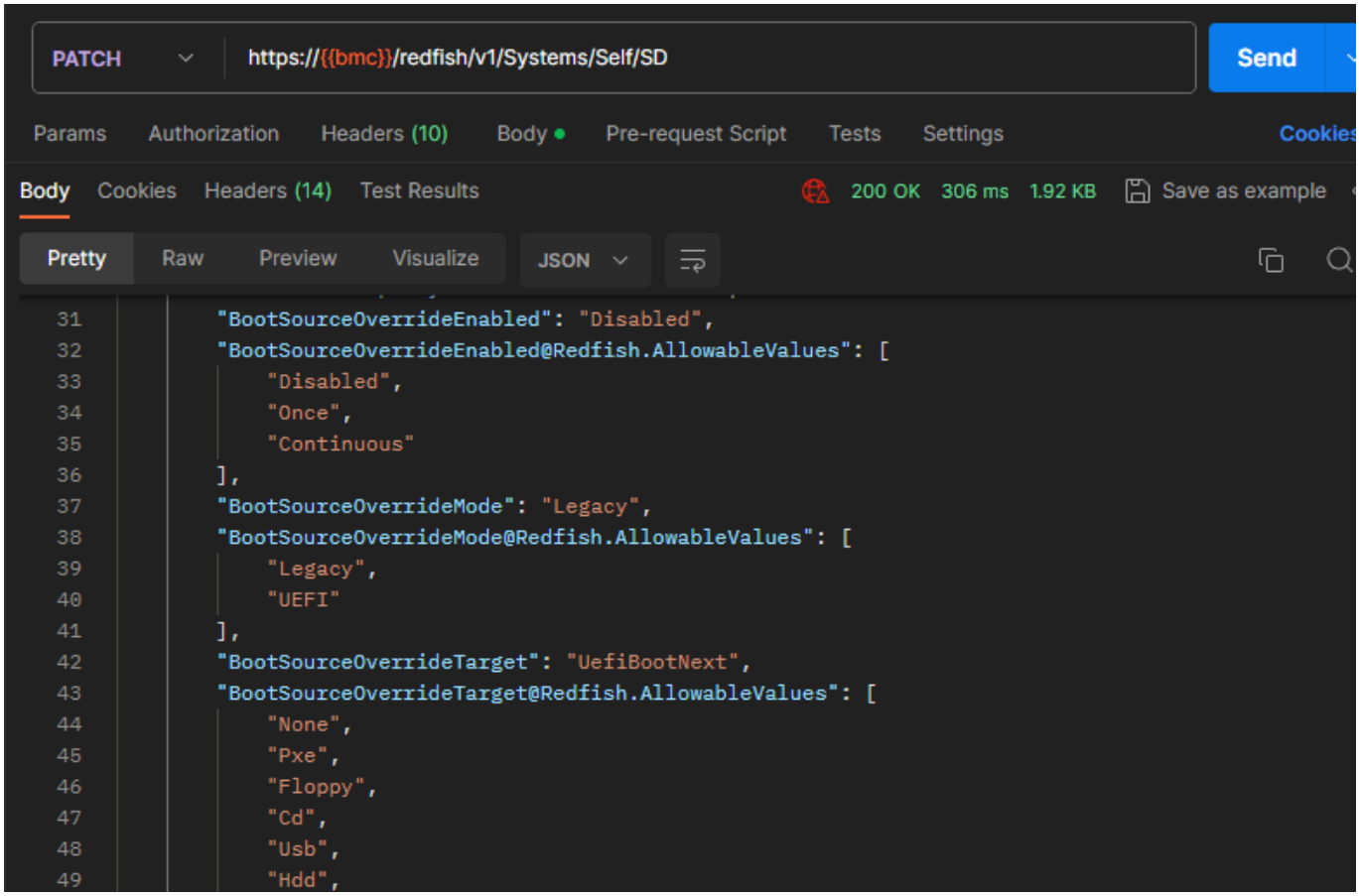
After the request is sent to the target, such as S8056, the response status is 204 without response body if the request is successful.

6.4.2 Configuring Uefi Boot Next

URI: /redfish/v1/Systems/Self/SD
Method: PATCH
Example:
Edit the request content in JSON format from Body.



After the request is sent to the target, such as S8056, the response status is 200 with response body in JSON format if the request is successful.



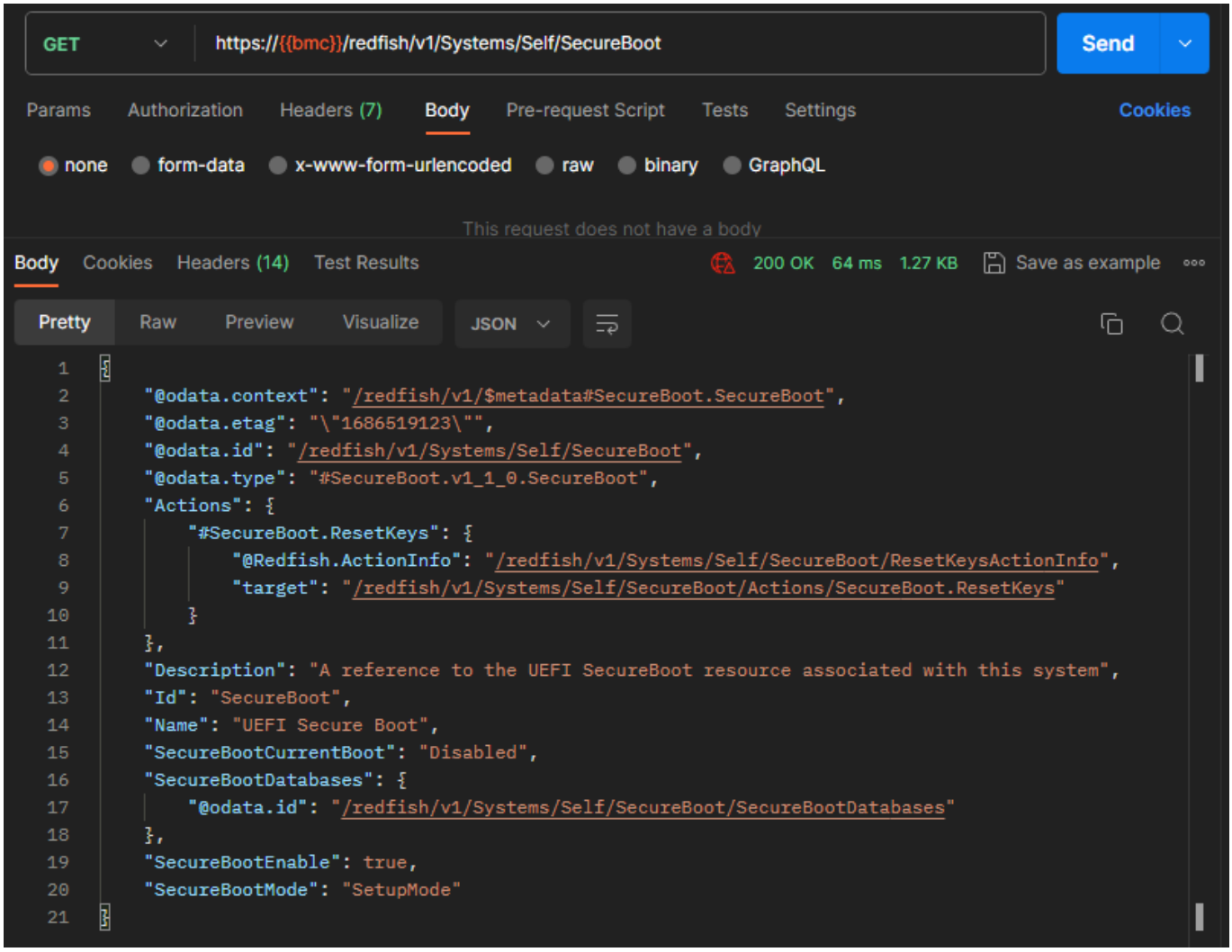
6.5 Secure Boot

UEFI Secure Boot was created to enhance security in the pre-boot environment. Secure Boot helps firmware, operating system and hardware providers cooperate to thwart the efforts of malware developers.

Note: Please use the supported BIOS to use this function.

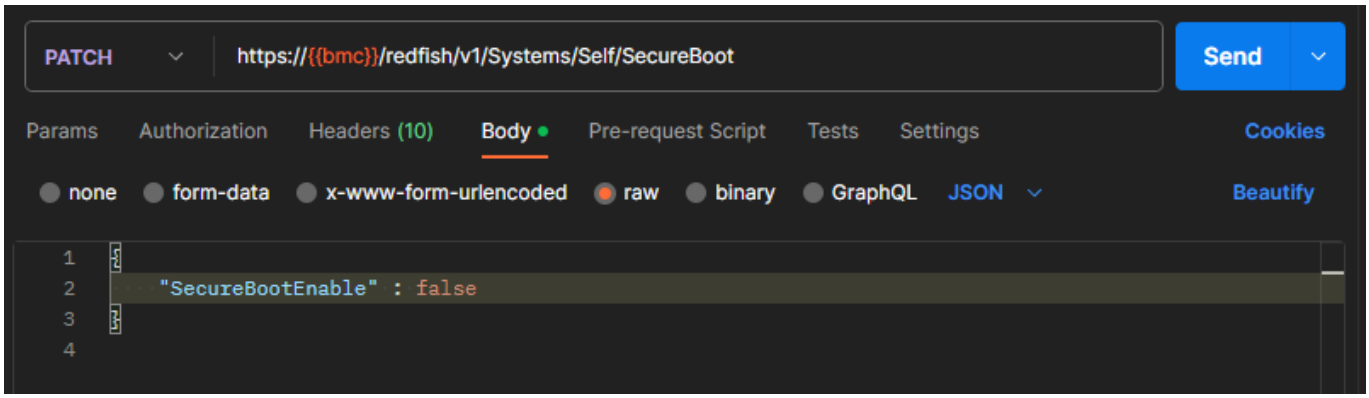
6.5.1 Enabling Redfish Secure Boot by GET

URI: /redfish/v1/Systems/Self/SecureBoot
Method: GET
Example:
After the request is sent to the target, such as S8056, the response status is 200 with response body in JSON format if the request is successful.



6.5.2 Enabling Redfish Secure Boot by PATCH

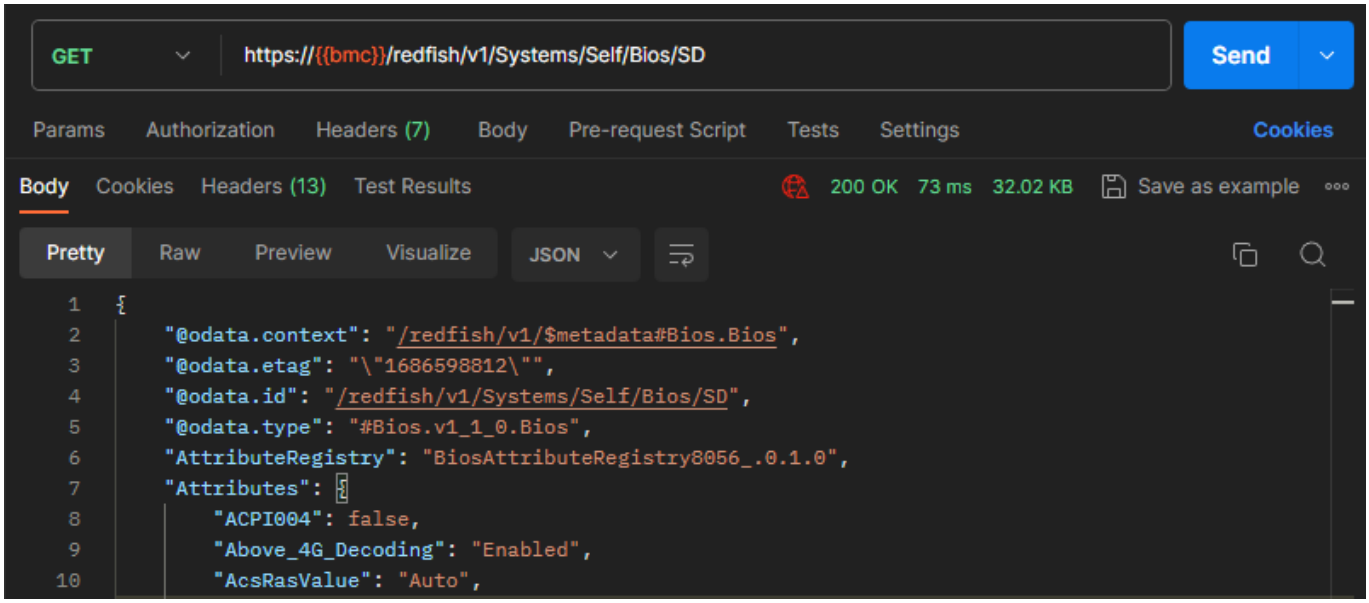
URI: /redfish/v2/Systems/Self/SecureBoot
Method: PATCH
Example:
Edit the request content in JSON format from Body.



After the request is sent to the target, such as S8056, the response status is 204 without response body if the request is successful.

6.5.3 Confirming in Pending Settings

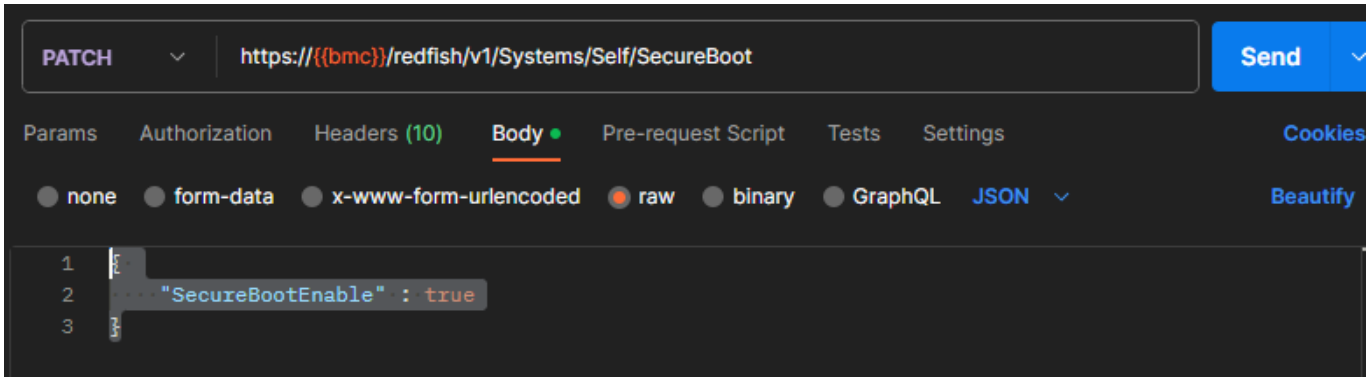
URI: /redfish/v1/Systems/Self/Bios/SD
Method: GET
Example:
After the request is sent to the target, such as S8056, the response status is 200 with response body in JSON format if the request is successful.



6.5.4 Enabling Secure Boot in BIOS

URI: /redfish/v1/Systems/Self/SecureBoot
Method: PATCH

Example:
Edit the request content in JSON format from Body.



After the request is sent to the target, such as S8056, the response status is 204 without response body if the request is successful.

Chapter 7. Certificate Service

The CertificateService describes a Certificate Service that represents the actions available to managecertificates and links to the certificates.

URI: /redfish/v1/CertificateService
Method: GET
Example:
After the request is sent to the target, such as S8056, the response status is 200 with response body in JSON format if the request is successful.

GET https://{{bmc}}/redfish/v1/CertificateService Send

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Body Cookies Headers (14) Test Results Status: 200 OK Time: 118 ms Size: 1.47 KB Save as example

Pretty Raw Preview Visualize JSON


```
{
    "@odata.context": "/redfish/v1/$metadata#CertificateService.CertificateService",
    "@odata.etag": "\"1686680208\"",
    "@odata.id": "/redfish/v1/CertificateService",
    "@odata.type": "#CertificateService.v1_0_2.CertificateService",
    "Actions": {
        "#CertificateService.GenerateCSR": {
            "@Redfish.ActionInfo": "/redfish/v1/CertificateService/CertificateService.GenerateCSRActionInfo",
            "target": "/redfish/v1/CertificateService/Actions/CertificateService.GenerateCSR"
        },
        "#CertificateService.ReplaceCertificate": {
            "@Redfish.ActionInfo": "/redfish/v1/CertificateService/CertificateService.ReplaceCertificateActionInfo",
            "target": "/redfish/v1/CertificateService/Actions/CertificateService.ReplaceCertificate"
        }
    },
    "CertificateLocations": {
        "@odata.id": "/redfish/v1/CertificateService/CertificateLocations"
    },
    "Description": "Get the available to manage certificates and link to certificates",
    "Id": "CertificateService",
    "Name": "Certificate Service"
}
```

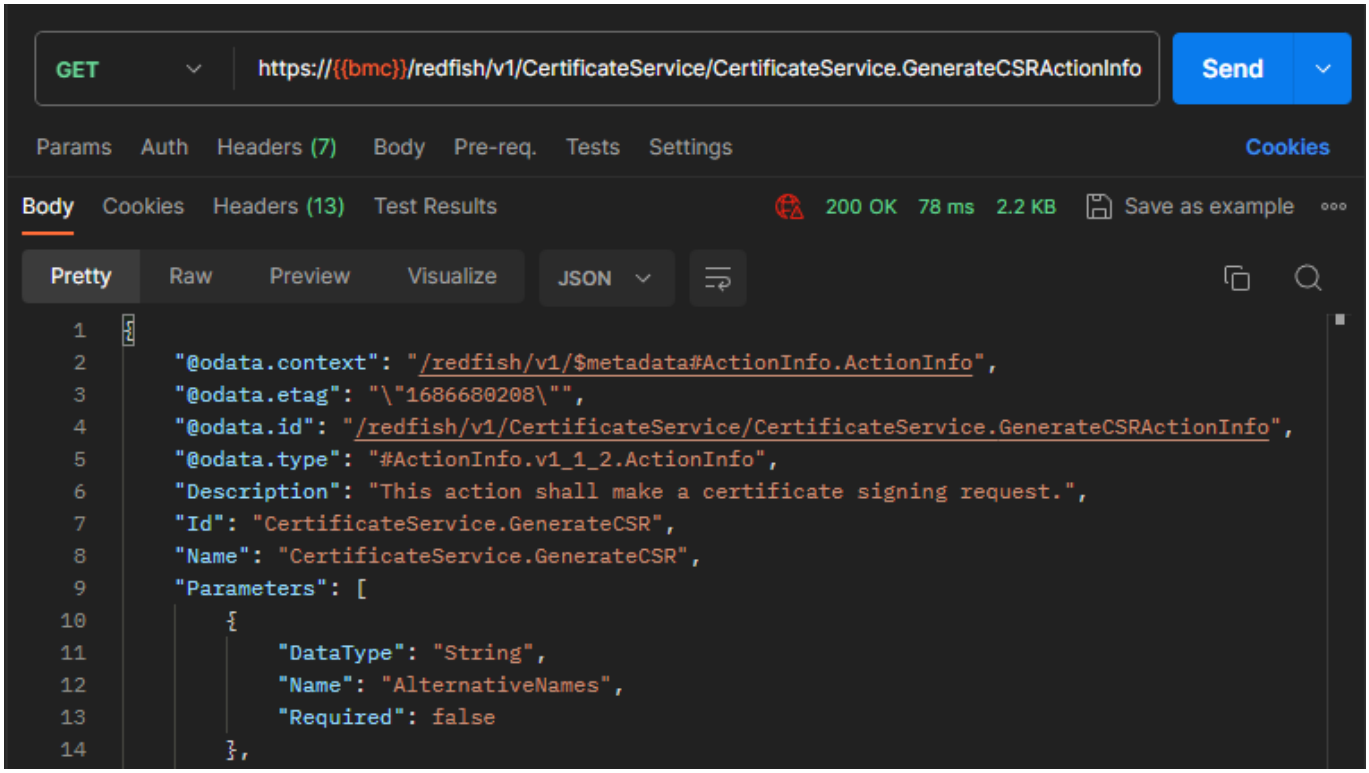
7.1 Generating CSR

Generate a certificate signing request (CSR) for the SSL certificate.

7.1.1 Generating CSR Action Info

List the supported and required parameters to generate CSR.

URI: /redfish/v1/CertificateService/CertificateService.GenerateCSRActionInfo
Method: GET
Example:
After the request is sent to the target, such as S8056, the response status is 200 with response body in JSON format if the request is successful.

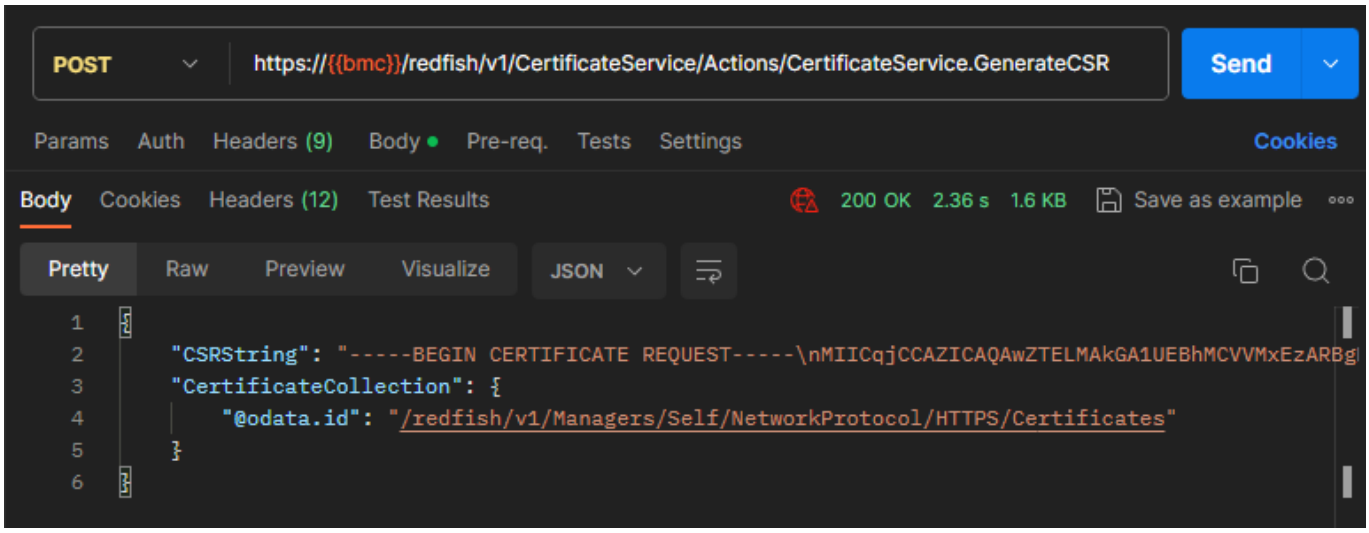


7.1.2 Generating a CSR Request

This action is used to perform a certificate signing request.

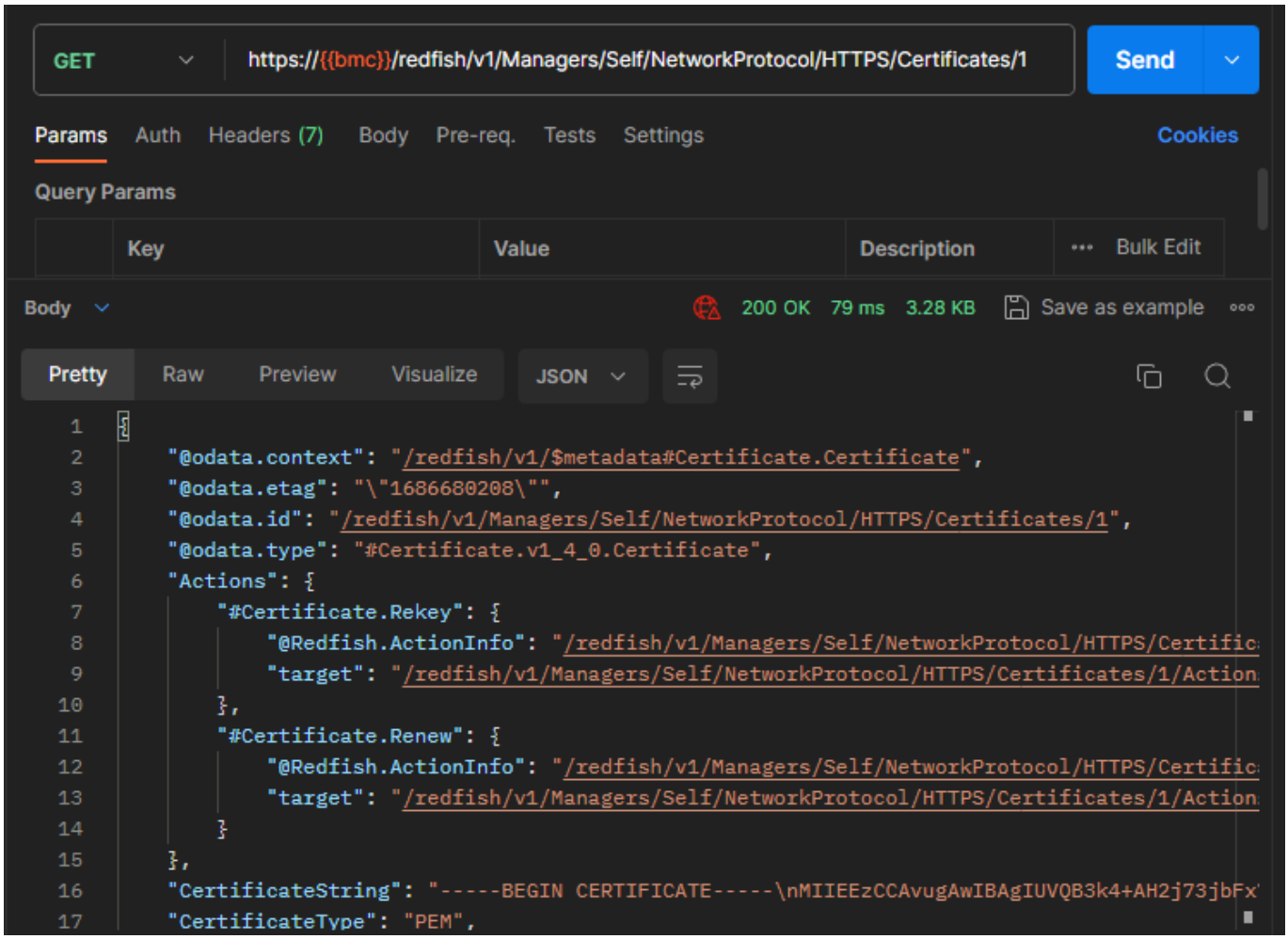
URI: /redfish/v1/CertificateService/Actions/CertificateService.GenerateCSR
Method: POST
Example:
Edit the request content in JSON format from Body.

After the request is sent to the target, such as S8056, the response status is 200 with response body in JSON format if the request is successful.



7.1.3 Viewing Certificate Details

URI: /redfish/v1/Managers/Self/NetworkProtocol/HTTPS/Certificates/1
Method: GET
Example:
After the request is sent to the target, such as S8056, the response status is 200 with response body in JSON format if the request is successful.



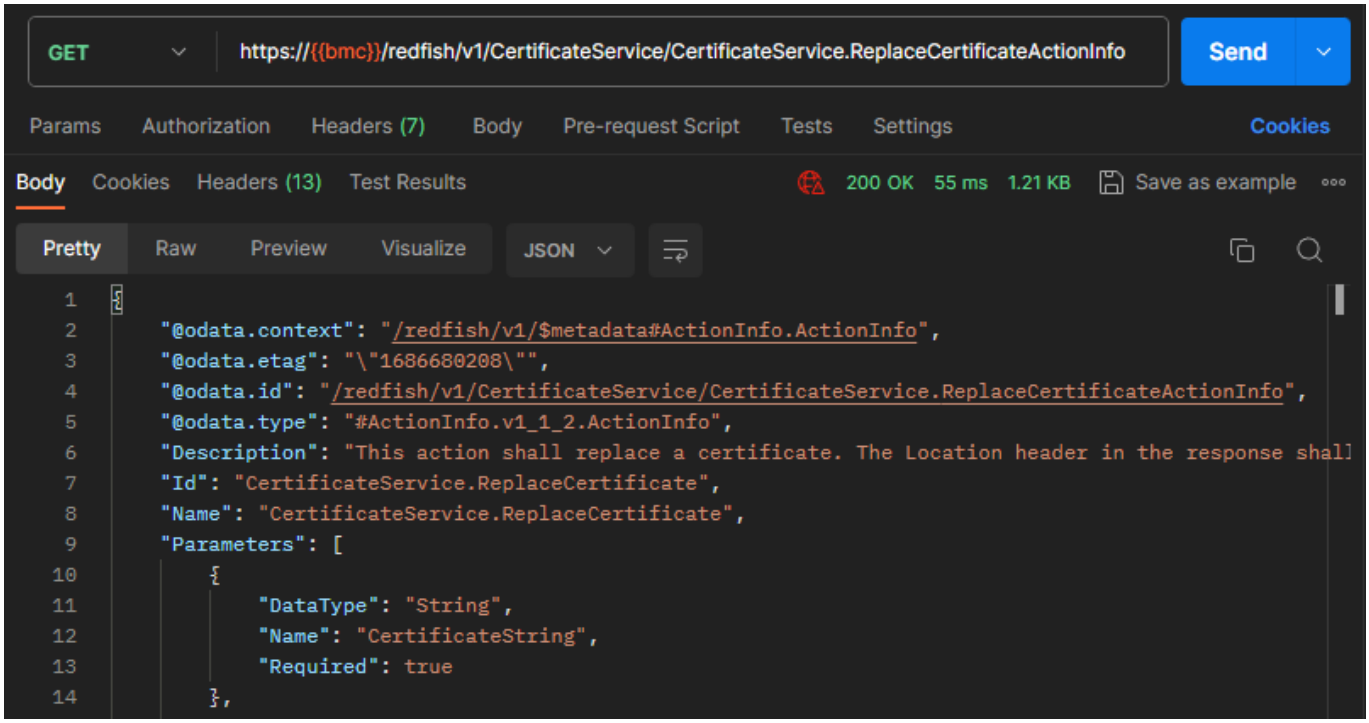
7.2 Replacing a Certificate

You can replace an existing certificate. Note that the new file must be a signed certificate.

7.2.1 Replacing Certificate Action Info

View the list of supported and required parameters to generate CSR.

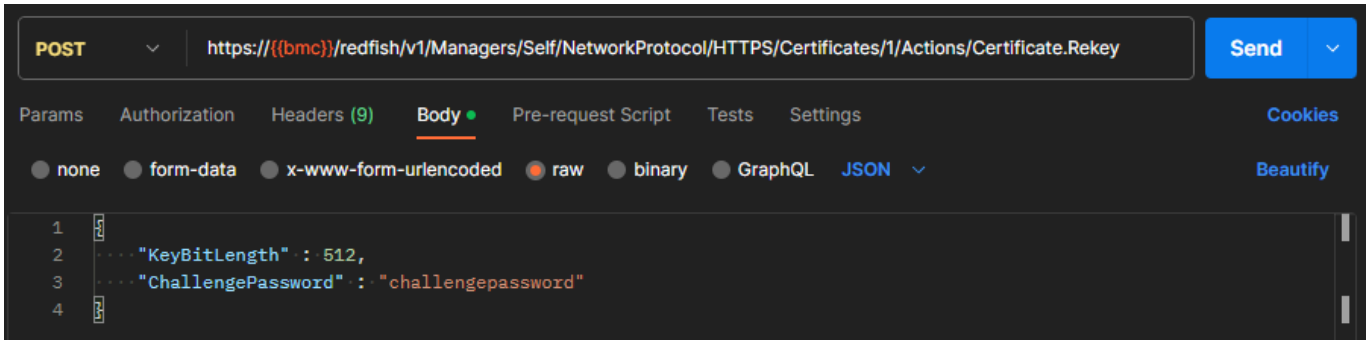
URI: /redfish/v1/CertificateService/CertificateService.ReplaceCertificateActionInfo
Method: GET
Example:
After the request is sent to the target, such as S8056, the response status is 200 with response body in JSON format if the request is successful.



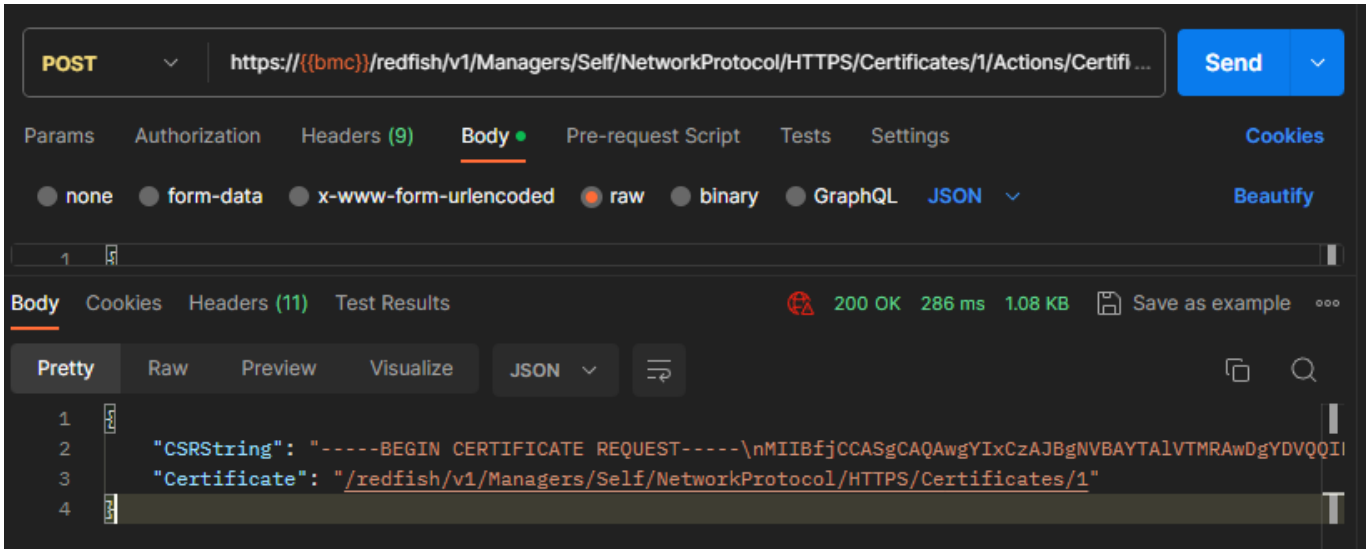
7.3 Replacing the Key Certificate

This action shall generate a new key pair for an existing certificate using the existingcertificate data. The response shall contain a signing request that is to be signed by a certificate authority (CA). The service should retain the private key used for the generation of this request when the certificate is installed.

The private key should not be part of the response.
URI: /redfish/v1/Managers/Self/NetworkProtocol/HTTPS/Certificates/1/Actions/Certificate.Rekey
Method: POST
Example:
Edit the request content in JSON format from Body.



After the request is sent to the target, such as S8056, the response status is 200 with response body in JSON format if the request is successful.

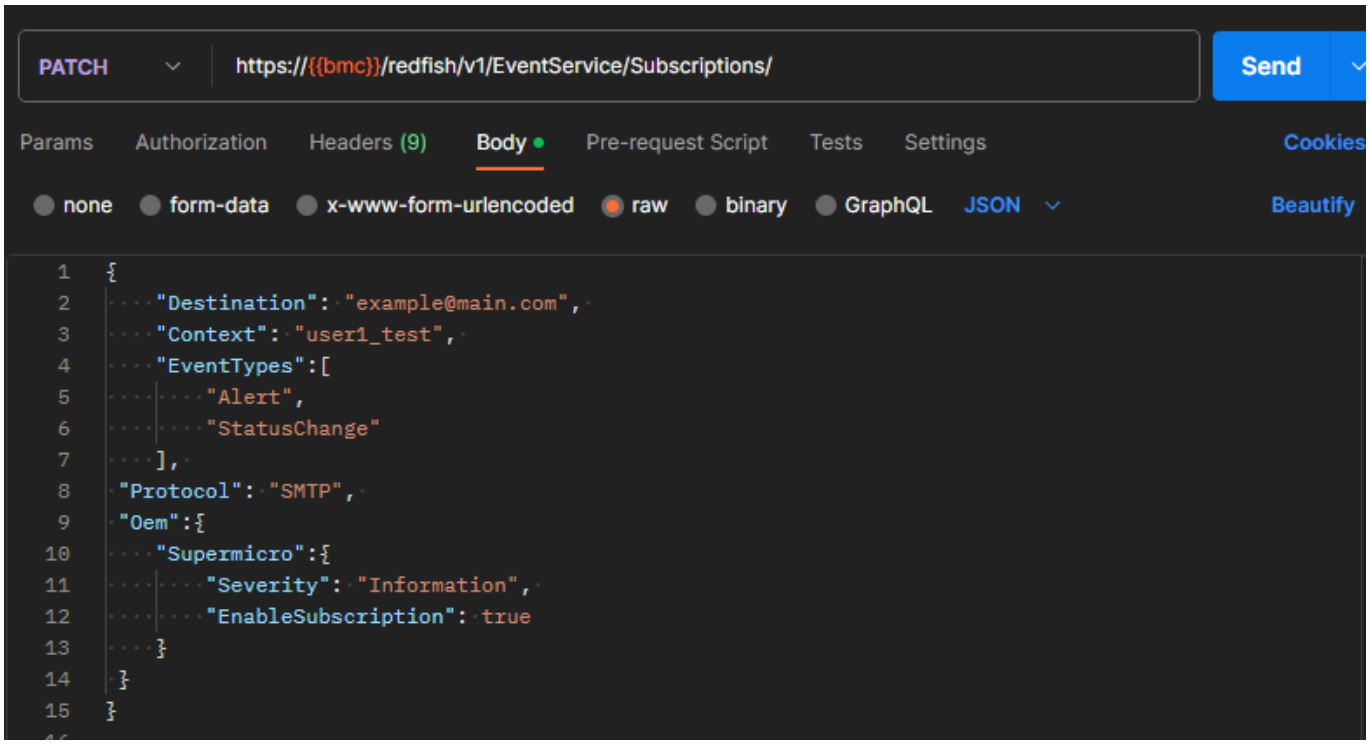


Chapter 8. Event Service

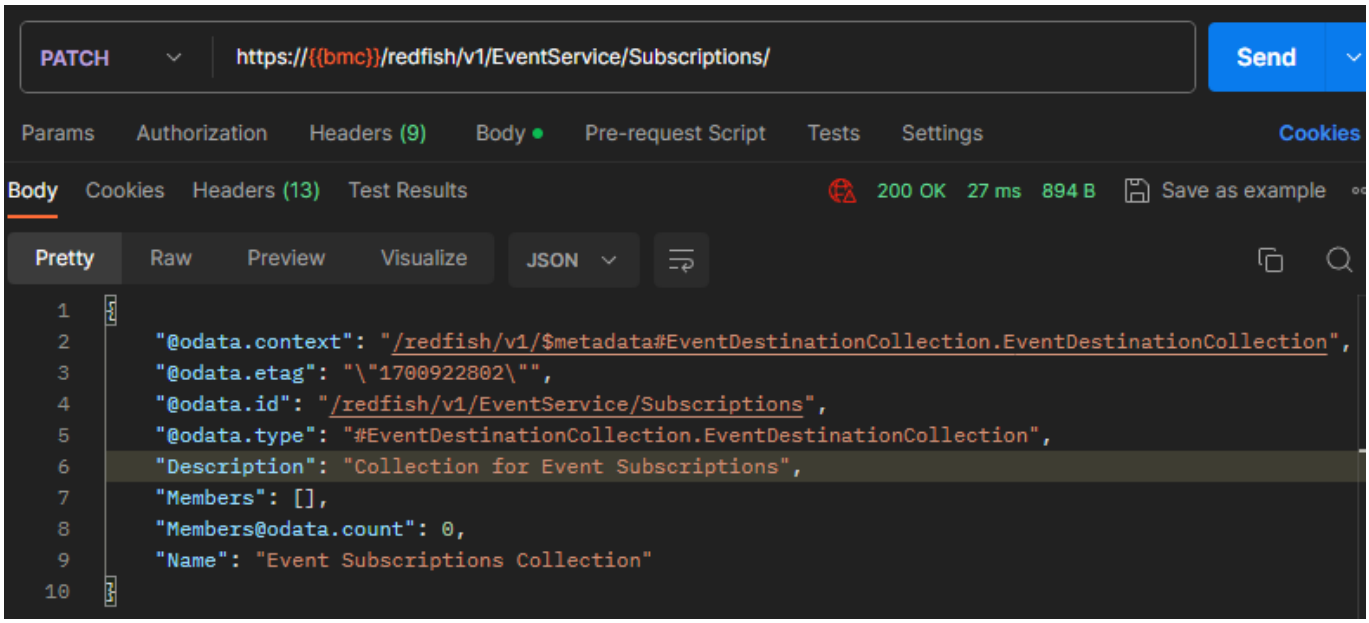
The event service is an alert mechanism for Redfish. This alert will be sent out through HTTP or HTTPS toa web service that is subscribed to the service.

8.1 Adding a Subscription

URI: /redfish/v1/EventService/Subscriptions/
Method: PATCH
Example:
Edit the request content in JSON format from Body.

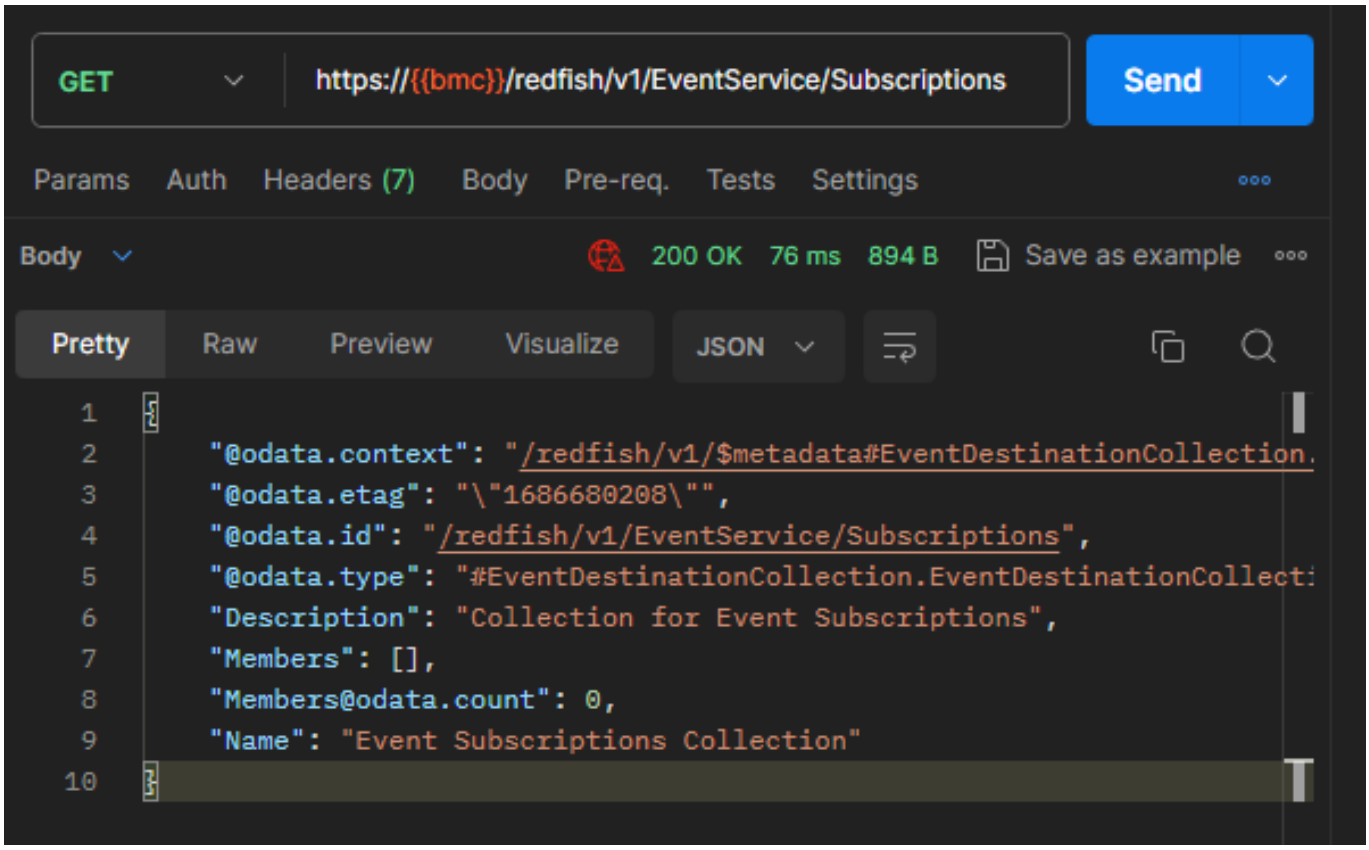


After the request is sent to the target, such as S8056, the response status is 200 with response body in JSON format if the request is successful.



8.2 Viewing All Subscriptions

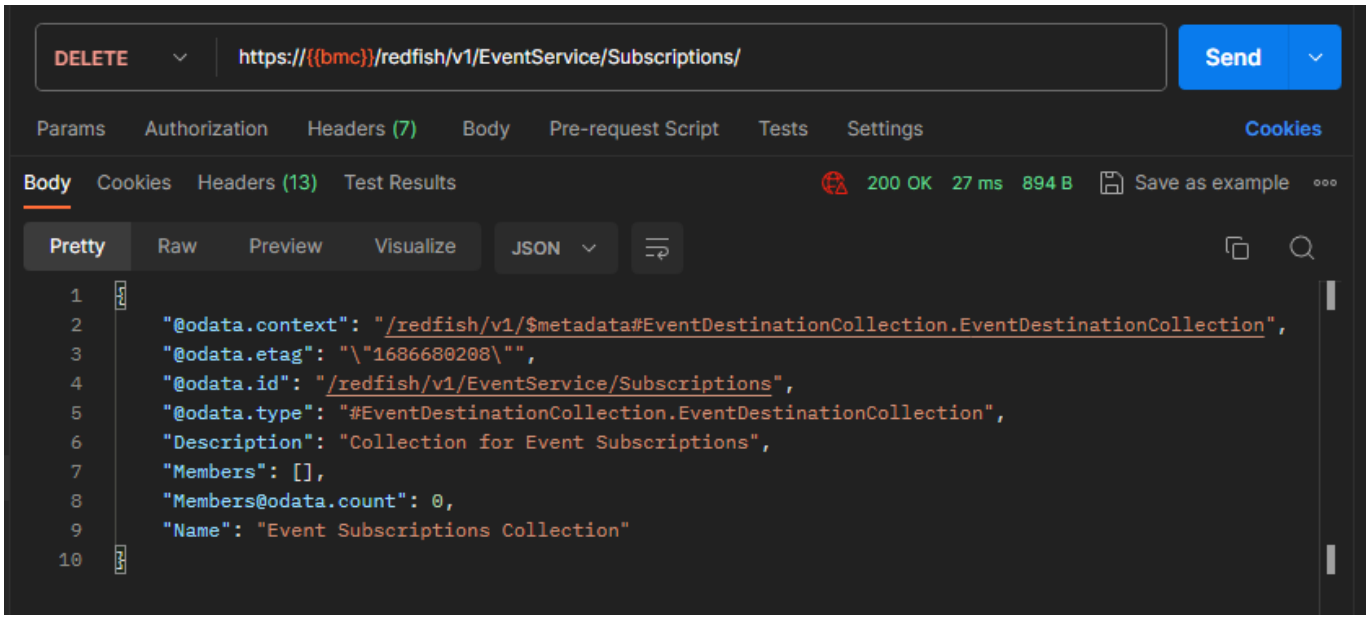
URI: /redfish/v1/EventService/Subscriptions
Method: GET
Example:
After the request is sent to the target, such as S8056, the response status is 200 with response body in JSON format if the request is successful.



8.3 Deleting a Subscription

You can delete a subscription.

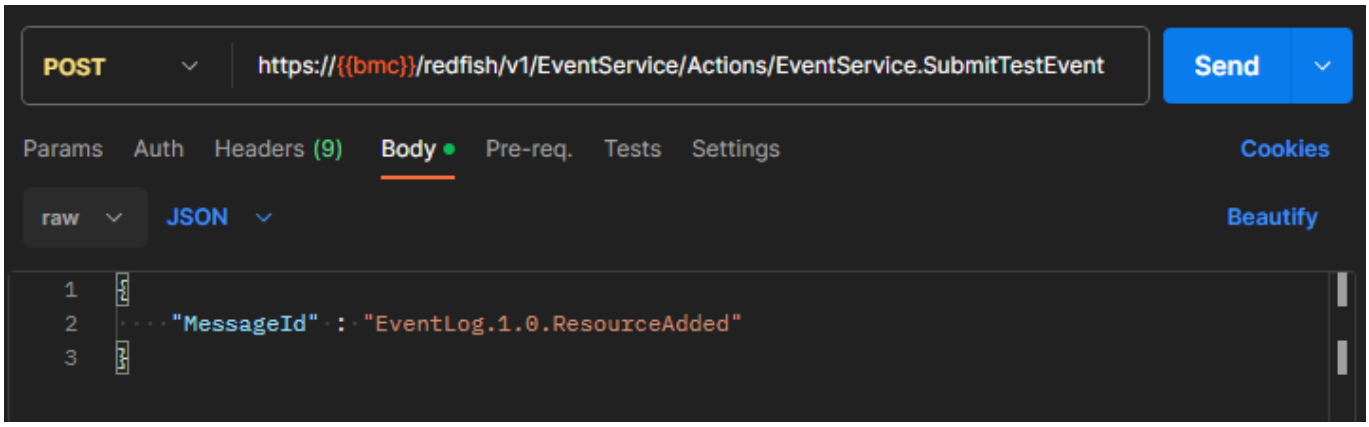
URI: /redfish/v1/EventService/Subscriptions/
Method: DELETE
Example:
After the request will be sent to the target, such as S8056, the response status is 200 with response body in JSON format if the request is successful.



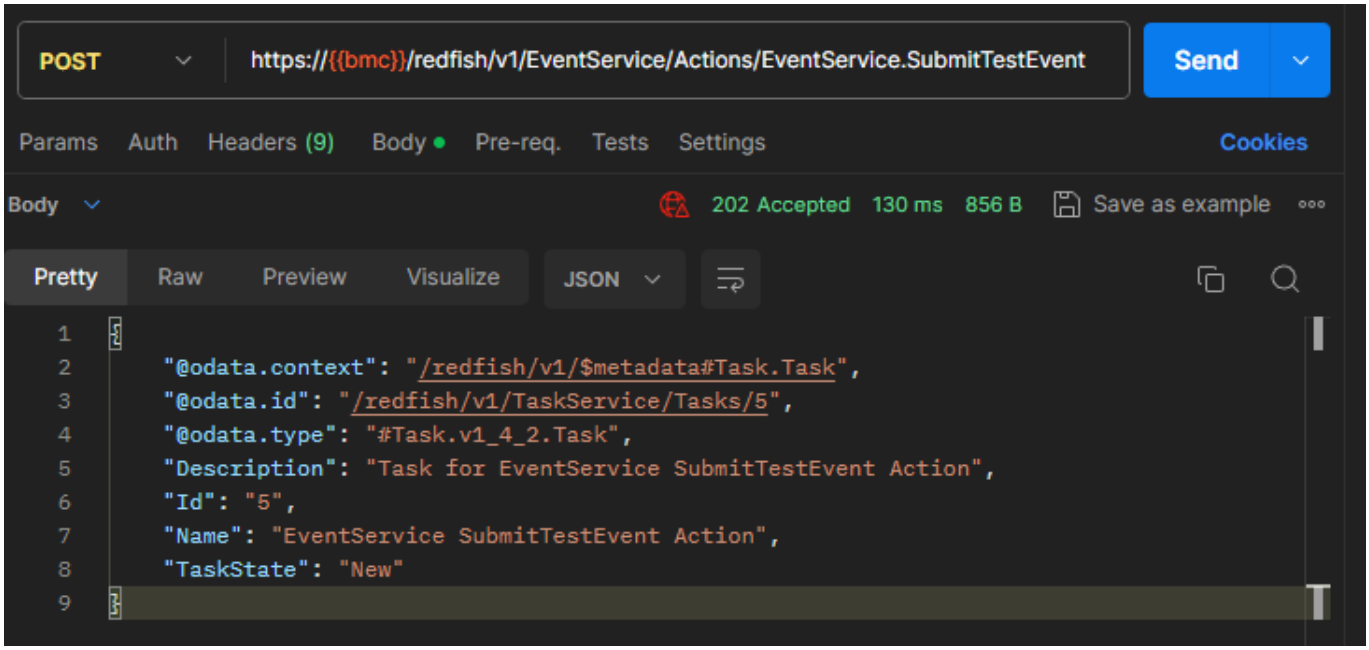
8.4 Testing an Event Subscription

You can send a test event with “SendTestEvent” or generate an event in the BMC, Redfish will thenautomatically send event alerts to the subscriber(s).

URI: /redfish/v1/EventService/Actions/EventService.SubmitTestEvent
Method: POST
Example:
Edit the request content in JSON format from Body.



After the request is sent to the target, such as S8056, the response status is 200 with response body in JSON format if the request is successful.

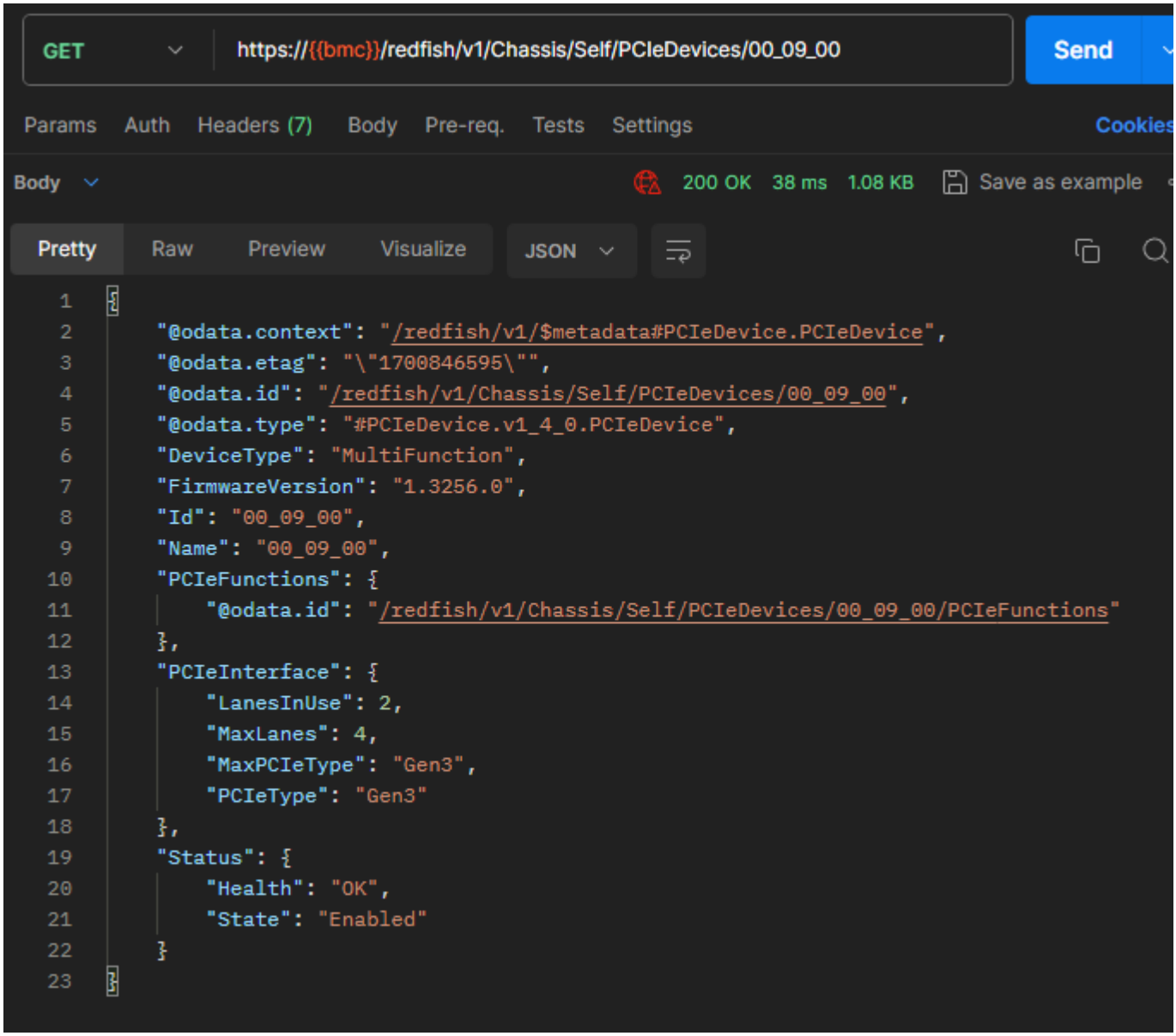


You can find details about all available network devices under /redfish/v1/Chassis/1/PCIeDevices

9.1 NIC Device

Method: GET
Example:
Assume the bus, device and function for NIC are 00, 09 and 00, respectively. Then, the redfish API for NIC is /redfish/v1/Chassis/Self/PCIeDevices/00_09_00.

After the request is sent to the target, such as S8056, the response status is 200 with response body in JSON format if the request is successful.



9.2 GPU

Method: GET
Example:

Assume the bus, device and function for NIC are 00, C1 and 00, respectively. Then, the redfish API for NIC is /redfish/v1/Chassis/Self/PCIeDevices/00_C1_00.

After the request is sent to the target, such as S8056, the response status is 200 with response body in JSON format if the request is successful.

GET https://{{bmc}}/redfish/v1/Chassis/Self/PCIeDevices/00_C1_00 Send

Params Auth Headers (7) Body Pre-req. Tests Settings Cookies

Body 200 OK 147 ms 1.05 KB Save as example

Pretty Raw Preview Visualize JSON


```
{
    "@odata.context": "/redfish/v1/$metadata#PCIeDevice.PCIeDevice",
    "@odata.etag": "\"1701281350\"",
    "@odata.id": "/redfish/v1/Chassis/Self/PCIeDevices/00_C1_00",
    "@odata.type": "#PCIeDevice.v1_4_0.PCIeDevice",
    "DeviceType": "SingleFunction",
    "Id": "00_C1_00",
    "Name": "00_C1_00",
    "PCIeFunctions": {
        "@odata.id": "/redfish/v1/Chassis/Self/PCIeDevices/00_C1_00/PCIeFunctions"
    },
    "PCIeInterface": {
        "LanesInUse": 16,
        "MaxLanes": 16,
        "MaxPCIeType": "Gen3",
        "PCIeType": "Gen3"
    },
    "Status": {
        "Health": "OK",
        "State": "Enabled"
    }
}
```

9.3 NVMe SSD

Method: GET
Example:
Assume the bus, device and function for NIC are 00, 83 and 00, respectively. Then, the redfish API for NIC is /redfish/v1/Chassis/Self/PCIeDevices/00_83_00.

After the request is sent to the target, such as S8056, the response status is 200 with response body in JSON format if the request is successful.

GET https://{{bmc}}/redfish/v1/Chassis/Self/PCIeDevices/00_83_00 Send

Params Auth Headers (7) Body Pre-req. Tests Settings Cookies

Body 200 OK 24 ms 1.05 KB Save as example

Pretty Raw Preview Visualize JSON


```
{
    "@odata.context": "/redfish/v1/$metadata#PCIeDevice.PCIeDevice",
    "@odata.etag": "\"1700846595\"",
    "@odata.id": "/redfish/v1/Chassis/Self/PCIeDevices/00_83_00",
    "@odata.type": "#PCIeDevice.v1_4_0.PCIeDevice",
    "DeviceType": "SingleFunction",
    "Id": "00_83_00",
    "Name": "00_83_00",
    "PCIeFunctions": {
        "@odata.id": "/redfish/v1/Chassis/Self/PCIeDevices/00_83_00/PCIeFunctions"
    },
    "PCIeInterface": {
        "LanesInUse": 4,
        "MaxLanes": 4,
        "MaxPCIeType": "Gen3",
        "PCIeType": "Gen3"
    },
    "Status": {
        "Health": "OK",
        "State": "Enabled"
    }
}
```

9.4 PCIe Functions

Method: GET
Example:
Assume the bus, device and function for this PCIe deviced are 00, 05 and 00, respectively. Then, the redfish API for NIC is /redfish/v1/Chassis/Self/PCIeDevices/00_05_00/PCIeFunctions/DevType3_LAN1_DevIndexF.

After the request is sent to the target, such as S8056, the response status is 200 with response body in JSON format if the request is successful.

GET https://{{bmc}}/redfish/v1/Chassis/Self/PCIeDevices/00_05_00/PCIeFunctions/DevType3_LAN1_DevIndexF Send

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Body Cookies Headers (14) Test Results Status: 200 OK Time: 48 ms Size: 1.38 KB Save as example

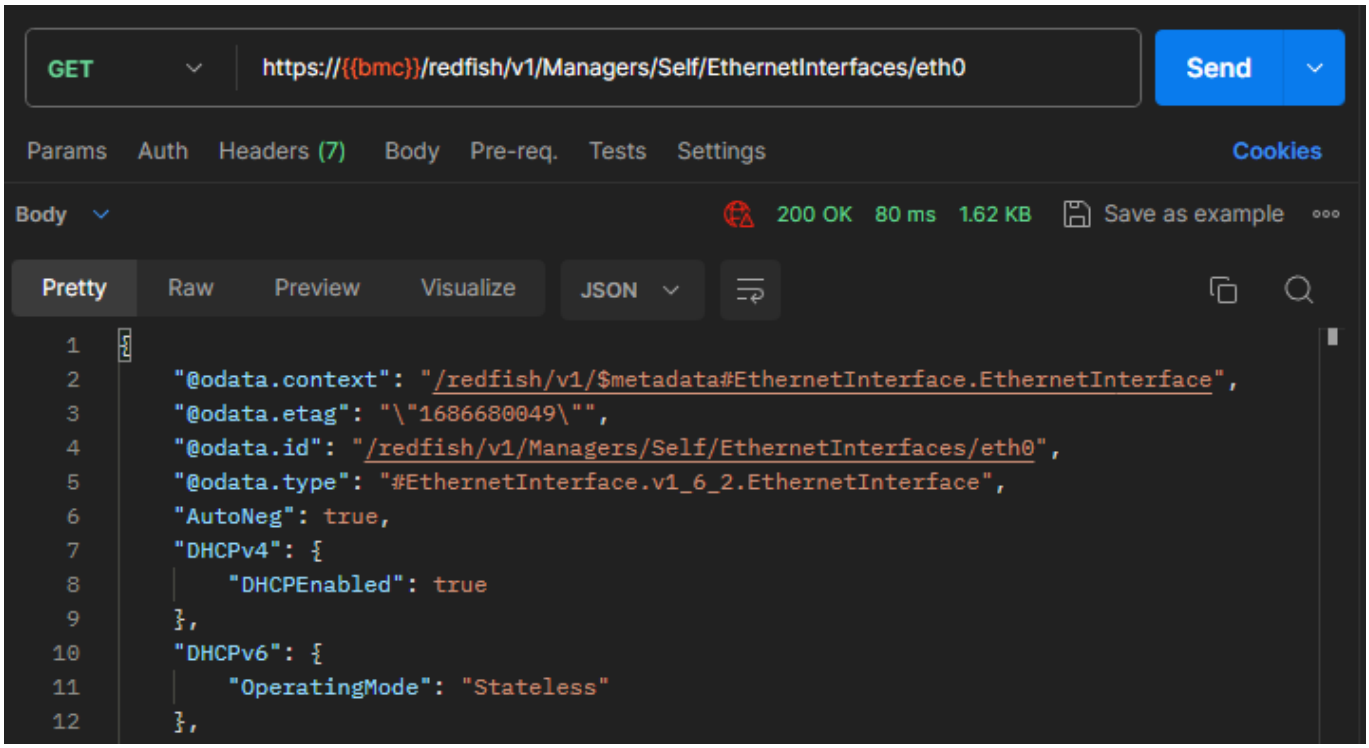
Pretty Raw Preview Visualize JSON


```
{
    "@odata.context": "/redfish/v1/$metadata#PCIeFunction.PCIeFunction",
    "@odata.etag": "\"1701428697\"",
    "@odata.id": "/redfish/v1/Chassis/Self/PCIeDevices/00_05_00/PCIeFunctions/DevType3_LAN1_DevIndexF",
    "@odata.type": "#PCIeFunction.v1_2_3.PCIeFunction",
    "ClassCode": "0x020000",
    "DeviceClass": "NetworkController",
    "DeviceId": "0x15FF",
    "FunctionId": 0,
    "FunctionType": "Physical",
    "Id": "DevType3_LAN1_DevIndexF",
    "Links": {
        "NetworkDeviceFunctions": [
            {
                "@odata.id": "/redfish/v1/Chassis/Self/NetworkAdapters/DevType7_NIC0/NetworkDeviceFunctions/NetworkDeviceFunction0"
            }
        ],
        "NetworkDeviceFunctions@odata.count": 1,
        "PCIeDevice": {
            "@odata.id": "/redfish/v1/Chassis/Self/PCIeDevices/00_05_00"
        }
    },
    "Name": "DevType3_LAN1_DevIndexF",
    "RevisionId": "0x02",
    "Status": {
        "Health": "OK",
        "State": "Enabled"
    },
    "SubsystemId": "0x0000",
    "SubsystemVendorId": "0x8086",
    "VendorId": "0x8086"
}
```

Ethernet Interfaces resources are used to manage BMC network configuration.

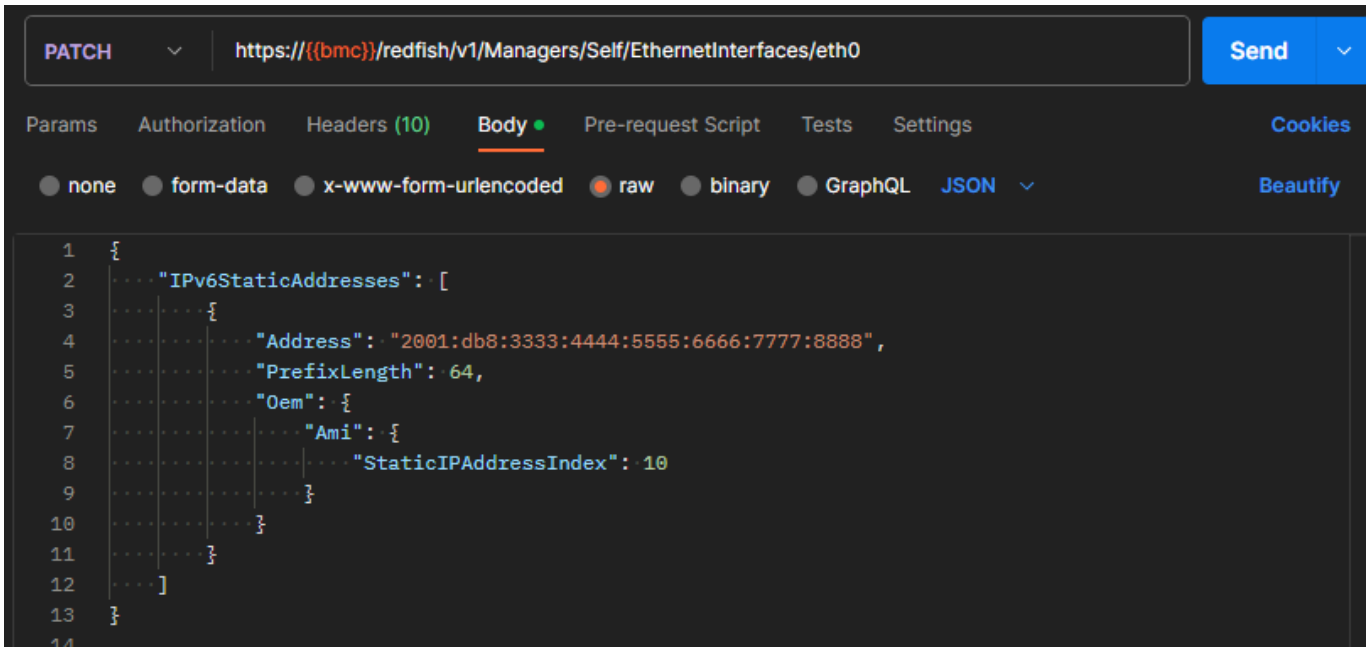
10.1 Viewing Network Settings

URI: /redfish/v1/Managers/Self/EthernetInterfaces/eth0
Method: GET
Example:
After the request is sent to the target, such as S8056, the response status is 200 with response body in JSON format if the request is successful.

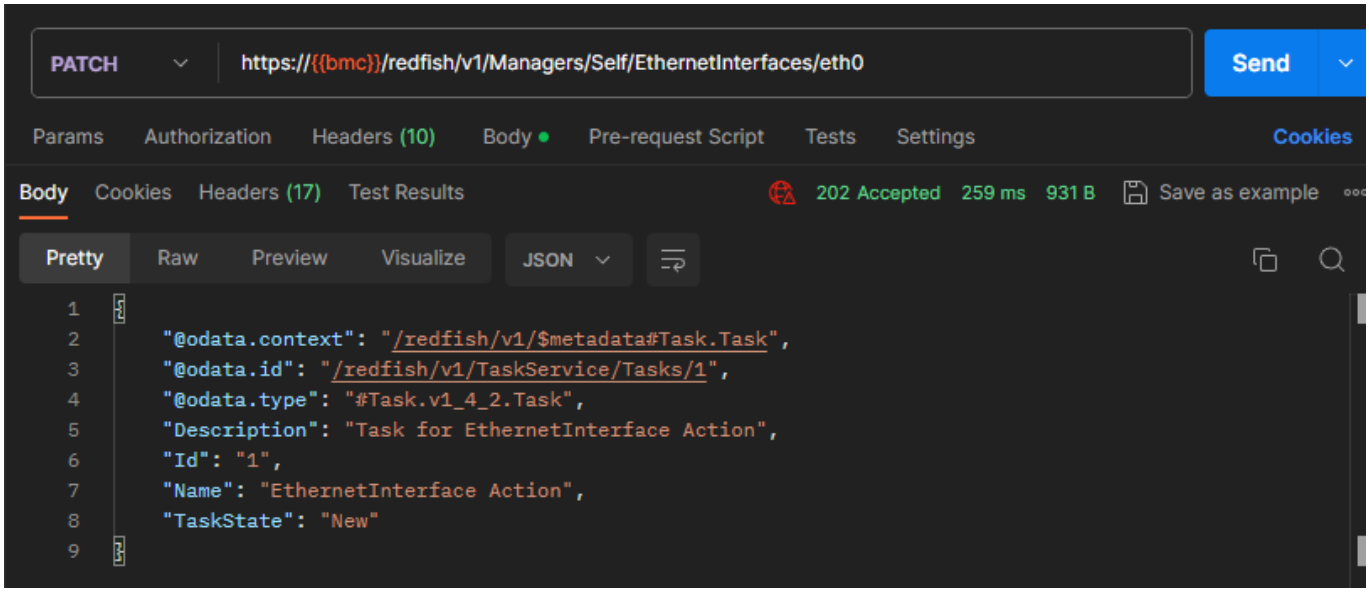


10.2 IPv6 Configuration

URI: /redfish/v1/Managers/Self/HostInterfaces/Self
Method: PATCH
Example:
Edit the request content in JSON format from Body.



After the request will be sent to the target, such as S8056, the response status is 202 with response body in JSON format if the request is successful.



Then, you can access /redfish/v1/TaskService/Tasks/1 to check the progress of setting the IPv6 address.

GET https://{{bmc}}/redfish/v1/TaskService/Tasks/1 Send

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Body Cookies Headers (14) Test Results 200 OK 59 ms 1.12 KB Save as example

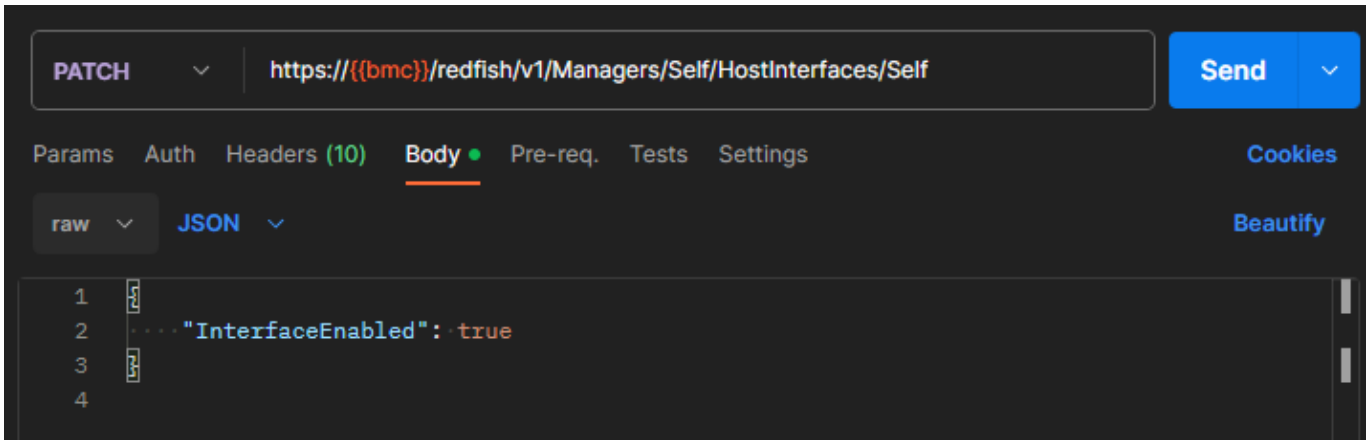
Pretty Raw Preview Visualize JSON


```
{
    "@odata.context": "/redfish/v1/$metadata#Task.Task",
    "@odata.etag": "\"1701261066\"",
    "@odata.id": "/redfish/v1/TaskService/Tasks/1",
    "@odata.type": "#Task.v1_4_2.Task",
    "Description": "Task for EthernetInterface Action",
    "EndTime": "2023-11-29T12:31:32+00:00",
    "Id": "1",
    "Messages": [
        {
            "@odata.type": "#Message.v1_0_8.Message",
            "Message": "The task with id 1 has completed.",
            "MessageArgs": [
                "1"
            ],
            "MessageId": "TaskEvent.1.0.TaskCompletedOK",
            "Resolution": "None.",
            "Severity": "OK"
        }
    ],
    "Name": "EthernetInterface Action",
    "StartTime": "2023-11-29T12:31:06+00:00",
    "TaskState": "Completed",
    "TaskStatus": "OK"
}
```

10.3 Host Interface

10.3.1 Enabling Host Interface

URI: /redfish/v1/Managers/Self/HostInterfaces/Self
Method: PATCH
Example:
Edit the request content in JSON format from Body.



After the request will be sent to the target, such as S8056, the response status is 200 without response body if the request is successful.

Chapter 11. Log Service

This resource represents system health event logs and maintenance event logs.

11.1 System Health Event Log

URI: /redfish/v1/Systems/Self/LogServices/BIOS
Method: GET
Example:
After the request is sent to the target, such as S8056, the response status is 200 with response body in JSON format if the request is successful.

GET https://{{bmc}}/redfish/v1/Systems/Self/LogServices/BIOS Send

Params Auth Headers (7) Body Pre-req. Tests Settings Cookies

Body Cookies Headers (14) Test Results 200 OK 66 ms 1.25 KB Save as example

Pretty Raw Preview Visualize JSON

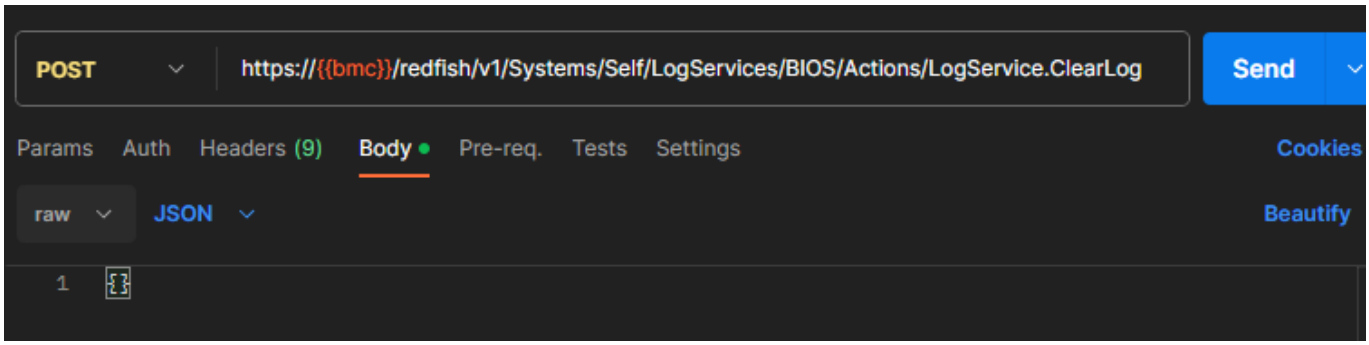

```
{
    "@odata.context": "/redfish/v1/$metadata#LogService.LogService",
    "@odata.etag": "\"1686679950\"",
    "@odata.id": "/redfish/v1/Systems/Self/LogServices/BIOS",
    "@odata.type": "#LogService.v1_1_3.LogService",
    "Actions": {
        "#LogService.ClearLog": {
            "target": "/redfish/v1/Systems/Self/LogServices/BIOS/Actions/LogService.ClearLog"
        }
    },
    "DateTime": "2023-06-13T20:56:57-00:00",
    "DateTimeLocalOffset": "-00:00",
    "Description": "IPMI SEL events for this manager",
    "Entries": {
        "@odata.id": "/redfish/v1/Systems/Self/LogServices/BIOS/Entries"
    },
    "Id": "BIOS",
    "LogEntryType": "SEL",
    "MaxNumberOfRecords": 150,
    "Name": "System BIOS Log Service",
    "OverWritePolicy": "WrapsWhenFull",
    "ServiceEnabled": true,
    "Status": {
        "Health": "OK",
        "State": "Enabled"
    }
}
```

11.1.1 Supported Actions

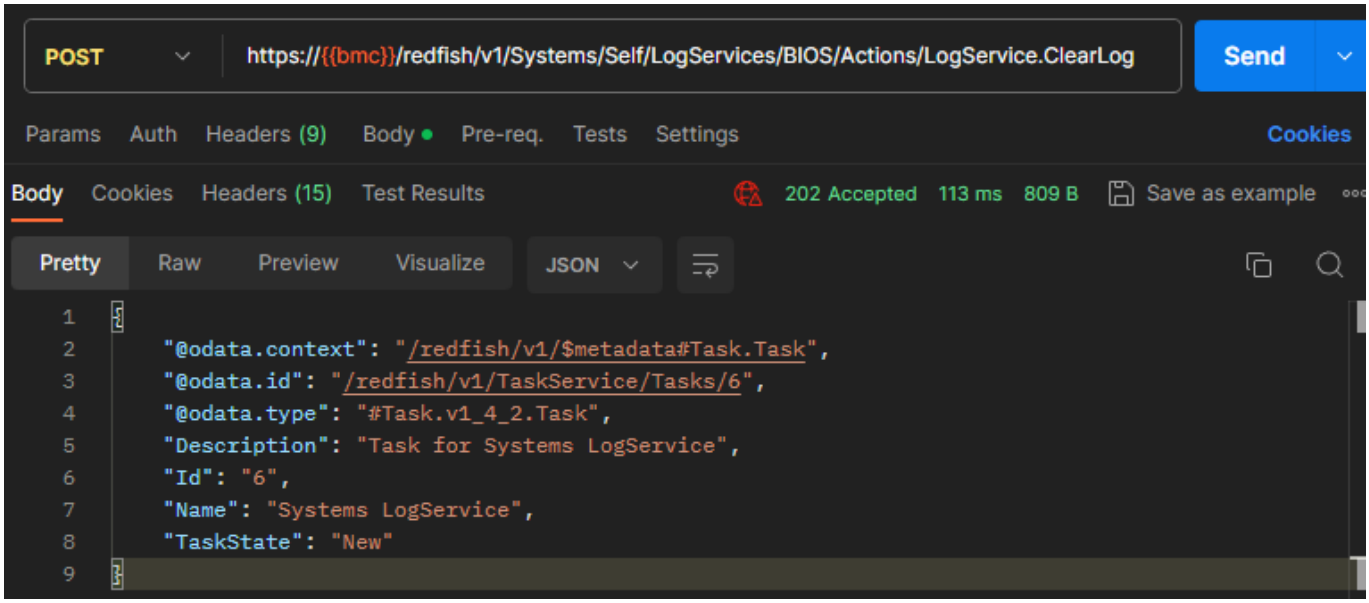
11.1.1.1 Clearing Logs

Use this API to delete all system health event log entries.

URI: /redfish/v1/Systems/Self/LogServices/BIOS/Actions/LogService.ClearLog
Method: POST
Example:
Edit the request content in JSON format from Body.



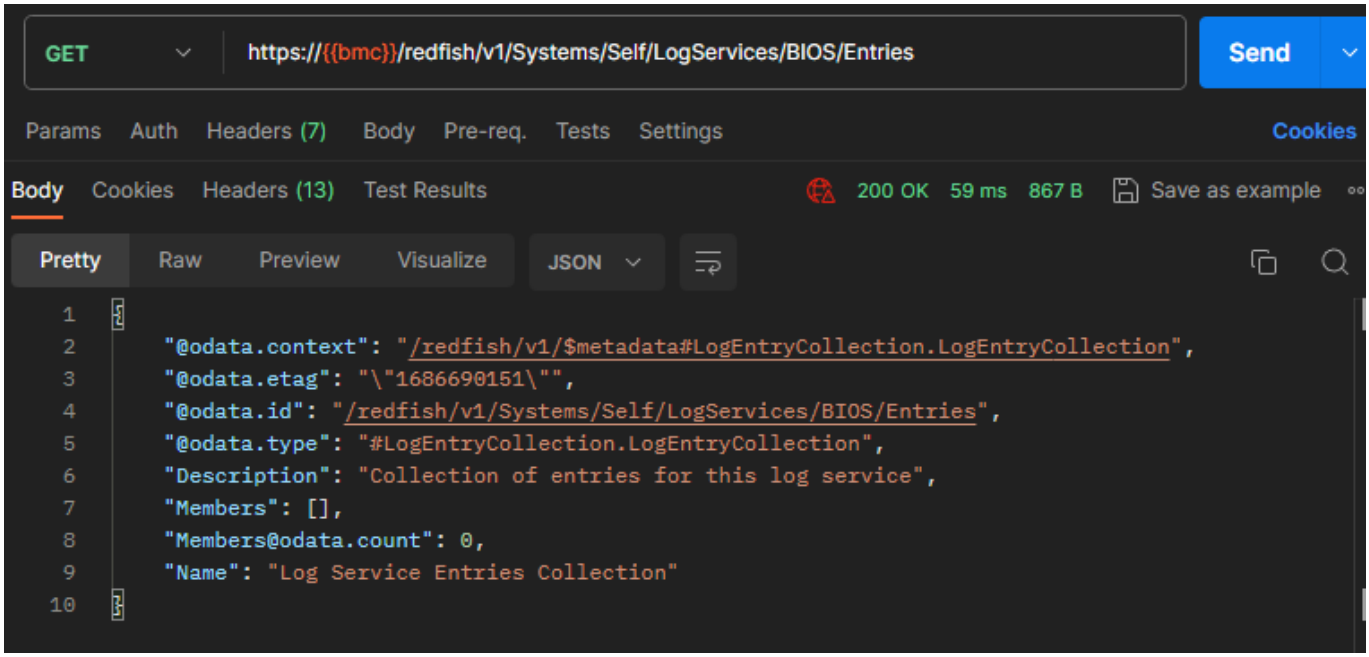
After the request is sent to the target, such as S8056, the response status is 202 with response body in JSON format if the request is successful.



11.1.1.2 Log Entry Collection

Navigate to view a collection of Log Entry resource instances.
/redfish/v1/Systems/1/LogServices

URI: /redfish/v1/Systems/Self/LogServices/BIOS/Entries
Method: GET
Example:
After the request will be sent to the target, such as S8056, the response status is 200 with response body in JSON format if the request is successful.



11.2 Maintenance Event Log

URI: /redfish/v1/Managers/1/LogServices/[logservice id]
Method: GET
Example:
After the request is sent to the target, such as S8056, the response status is 200 with response body in JSON format if the request is successful.

GET https://{{bmc}}/redfish/v1/Managers/Self/LogServices/SEL/Entries/1 Send

Params Auth Headers (7) Body Pre-req. Tests Settings Cookies

Body 200 OK 48 ms 1.4 KB Save as example

Pretty Raw Preview Visualize JSON

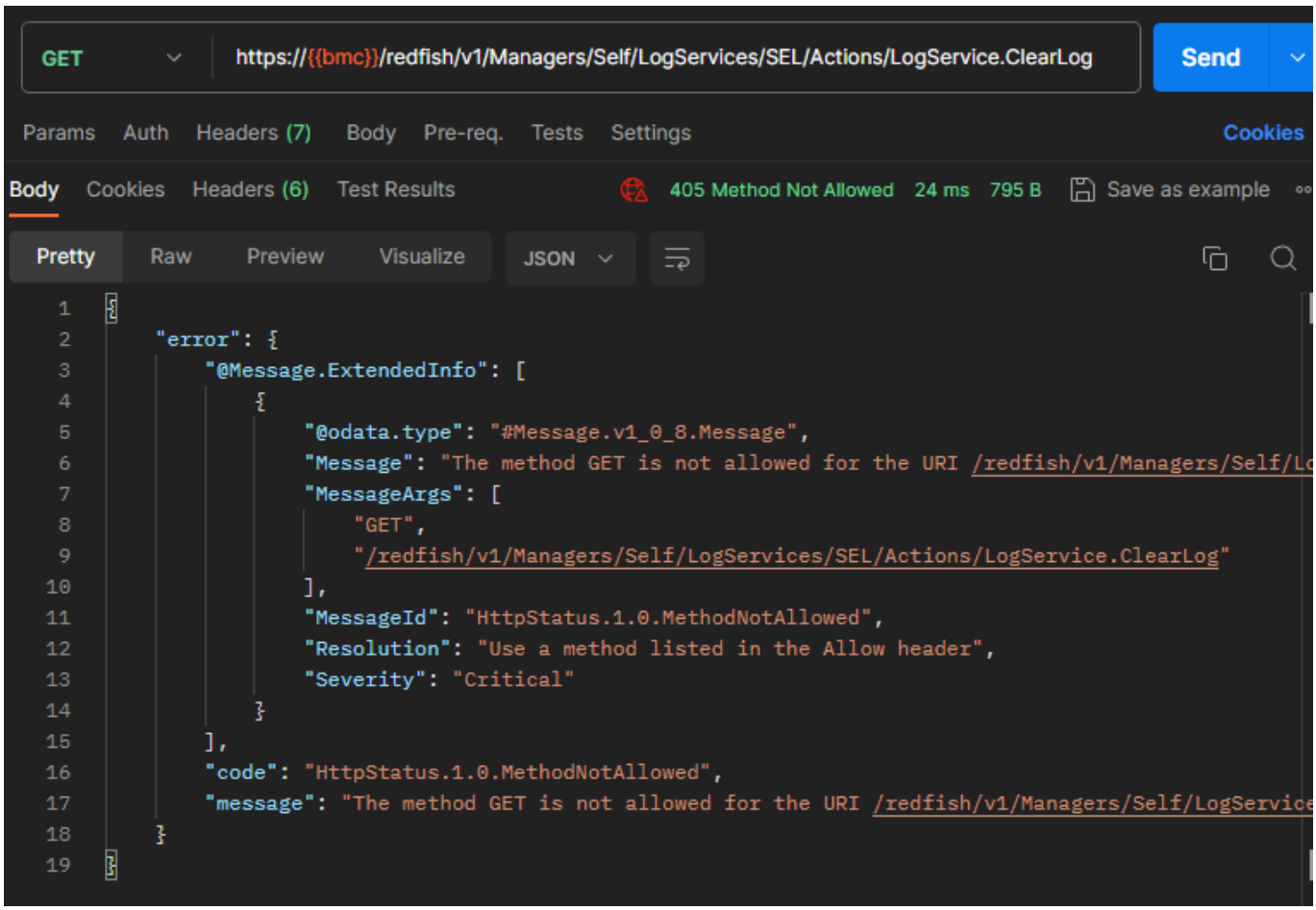

```
{
    "@odata.context": "/redfish/v1/$metadata#LogEntry.LogEntry",
    "@odata.etag": "\"1686690151\"",
    "@odata.id": "/redfish/v1/Managers/Self/LogServices/SEL/Entries/1",
    "@odata.type": "#LogEntry.v1_4_3.LogEntry",
    "Created": "2000-01-01T00:00:17+00:00",
    "Description": "SEL 3",
    "EntryCode": "Assert",
    "EntryType": "SEL",
    "EventTimestamp": "2023-08-17T00:18:13+00:00",
    "Id": "1",
    "Links": {
        "OriginOfCondition": {
            "@odata.id": "/redfish/v1/Systems/Self"
        }
    },
    "Message": "Event_Data_1 : 0, Record_Type : system event record, Sensor_Number : 13
    "MessageId": "0x00FFFF",
    "Name": "SEL 3",
    "SensorNumber": 131,
    "SensorType": "System ACPI PowerState",
    "Severity": "Ok"
}
```

11.2.1 Supported Actions

11.2.1.1 Clearing Logs

Use this API to delete all maintenance event log entries.

URI: /redfish/v1/Managers/Self/LogServices/SEL/Actions/LogService.ClearLog
Method: GET
Example:
After the request is sent to the target, such as S8056, the response status is 200 with response body in JSON format if the request is successful.



11.2.2 Log Entry Collection

Navigate to view the collection of Log Entry resource instances.

URI: /redfish/v1/Managers/Self/LogServices/SEL/Entries
Method: GET
Example:
After the request is sent to the target, such as S8056, the response status is 200 with response body in JSON format if the request is successful.

GET https://{{bmc}}/redfish/v1/Managers/Self/LogServices/SEL/Entries Send

Params Auth Headers (7) Body Pre-req. Tests Settings Cookies

Body Cookies Headers (14) Test Results 200 OK 104 ms 12.29 KB Save as example

Pretty Raw Preview Visualize JSON

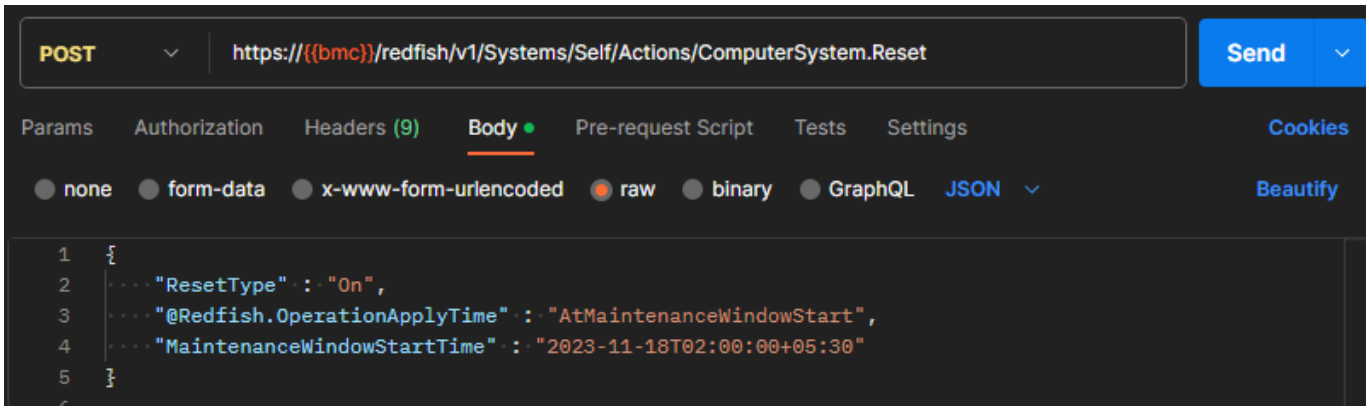

```
{
    "@odata.context": "/redfish/v1/$metadata#LogEntryCollection.LogEntryCollection",
    "@odata.etag": "\"1686679950\"",
    "@odata.id": "/redfish/v1/Managers/Self/LogServices/SEL/Entries",
    "@odata.type": "#LogEntryCollection.LogEntryCollection",
    "Description": "Collection of entries for this log service",
    "Members": [
        {
            "@odata.id": "/redfish/v1/Managers/Self/LogServices/SEL/Entries/1",
            "@odata.type": "#LogEntry.v1_4_3.LogEntry",
            "Created": "2000-01-01T00:00:17+00:00",
            "Description": "SEL 3",
            "EntryCode": "Assert",
            "EntryType": "SEL",
            "EventTimestamp": "2023-08-17T00:18:13+00:00",
            "Id": "1",
            "Links": {
                "OriginOfCondition": {
                    "@odata.id": "/redfish/v1/Systems/Self"
                }
            },
            "Message": "Event_Data_1 : 0, Record_Type : system event record, Sensor_Number : 131,
            "MessageId": "0x00FFFF",
            "Name": "SEL 3",
            "SensorNumber": 131,
            "SensorType": "System ACPI PowerState",
            "Severity": "Ok"
        },
```

Chapter 12. BMC Configuration Examples

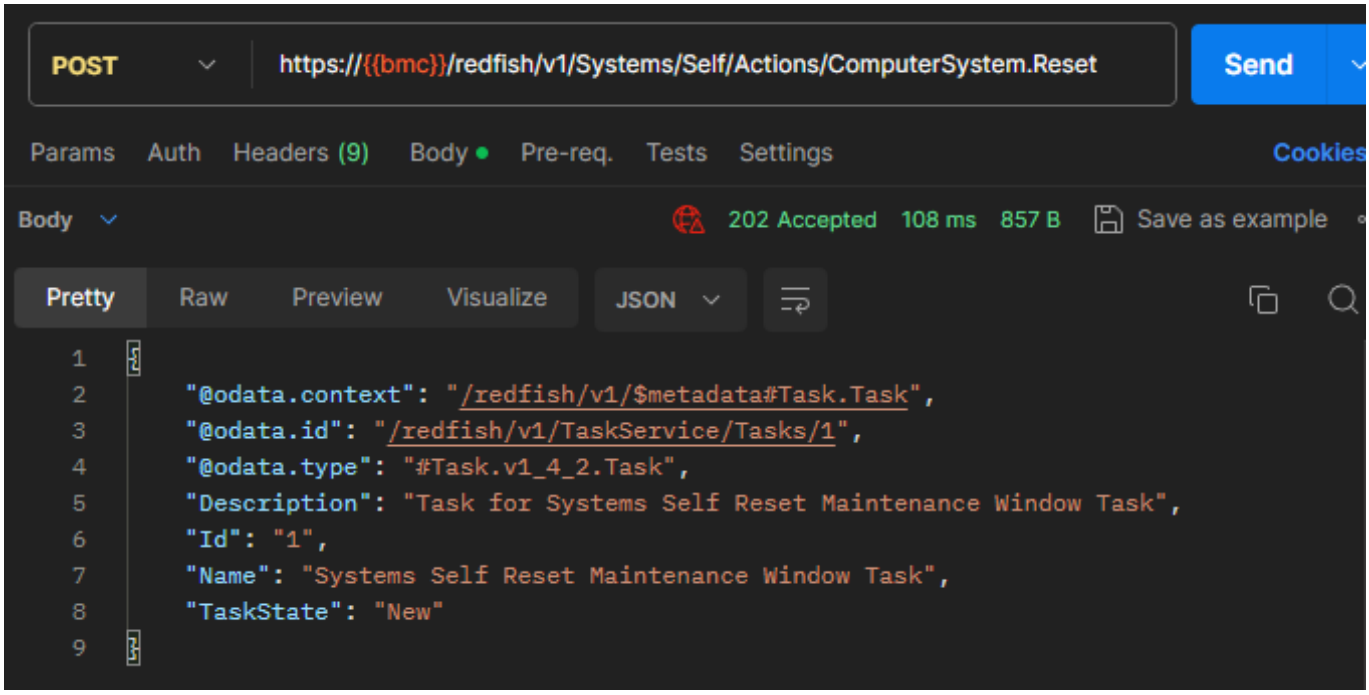
You can integrate current APIs into their software and applications in order to receive all services providedby Redfish APIs.

12.1 System Reset

URI: /redfish/v1/Systems/Self/Actions/ComputerSystem.Reset
Method: POST
Example:
Edit the request content in JSON format from Body.



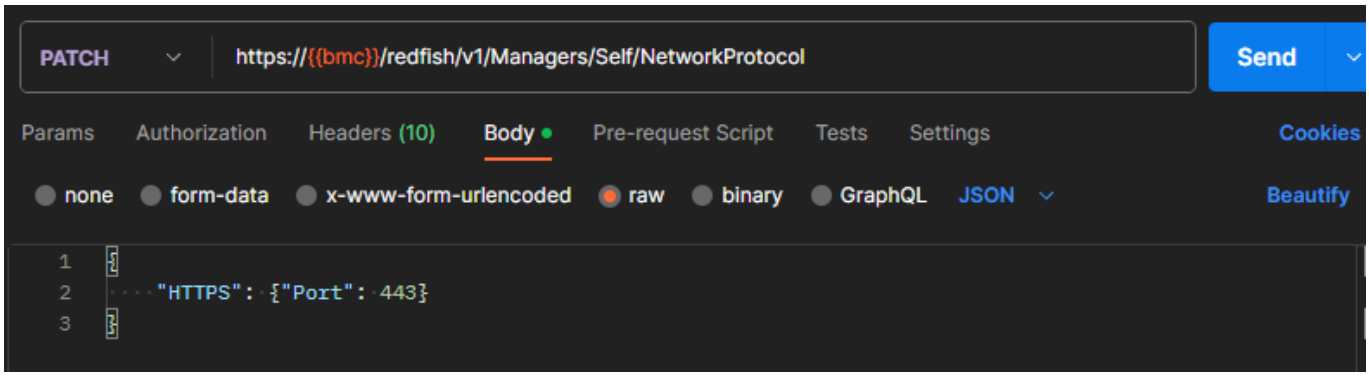
After the request is sent to the target, such as S8056, the response status is 202 with response body in JSON format if the request is successful.



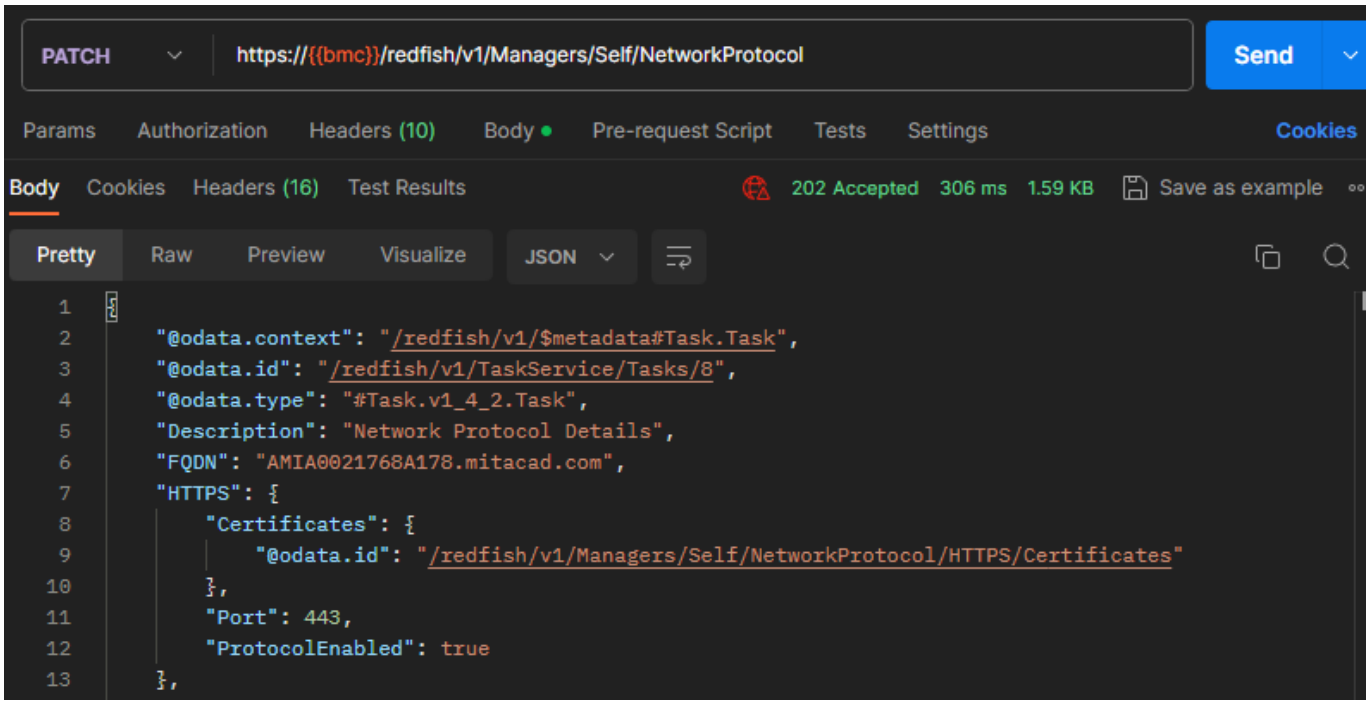
12.2 Notifications

12.2.1 SNMP

URI: /redfish/v1/Managers/Self/NetworkProtocol
Method: PATCH
Example:
Edit the request content in JSON format from Body.



After the request is sent to the target, such as S8056, the response status is 200 with response body in JSON format if the request is successful.



12.3 Getting MAC Addresses from System NICs

URI: /redfish/v1/Systems/Self/EthernetInterfaces/EthernetInterface0
Method: GET
Example:
After the request is sent to the target, such as S8056, the response status is 200 with response body in JSON format if the request is successful.

GET https://{{bmc}}/redfish/v1/Systems/Self/EthernetInterfaces/EthernetInterface0 Send

Params Auth Headers (7) Body Pre-req. Tests Settings Cookies

Body Cookies Headers (16) Test Results 200 OK 76 ms 1.87 KB Save as example

Pretty Raw Preview Visualize JSON


```
            "AddressOrigin": "Static",
            "SubnetMask": "0.0.0.0"
        }
    ],
    "IPv6Addresses": [
        {
            "Address": "FE80::A242:3FFF:FE50:830",
            "AddressOrigin": "Static",
            "PrefixLength": 64
        }
    ],
    "IPv6DefaultGateway": "::",
    "Id": "EthernetInterface0",
    "InterfaceEnabled": false,
    "LinkStatus": "LinkDown",
    "Links": {
        "Chassis": {
            "@odata.id": "/redfish/v1/Chassis/Self"
        },
        "NetworkDeviceFunction": {
            "@odata.id": "/redfish/v1/Chassis/Self/NetworkAdapters/DevType7_NIC0/NetworkDevi
        }
    },
    "MACAddress": "A0:42:3F:50:08:30",
    "MTUSize": 1500,
    "Name": "EthernetInterface0",
    "PermanentMACAddress": "A0:42:3F:50:08:30",
    "Status": {
        "Health": "OK",
        "State": "Disabled"
    },
```

12.4 Chassis Intrusion

URI: /redfish/v1/Chassis/Self
Method: GET/PATCH
Response: 200
Example:
After the request is sent to the target, such as S8056, the response status is 200 with response body in JSON format if the request is successful.

GET https://{{bmc}}/redfish/v1/Chassis/Self Send

Params Auth Headers (7) Body Pre-req. Tests Settings Cookies

Body Cookies Headers (22) Test Results 200 OK 88 ms 3.42 KB Save as example

Pretty Raw Preview Visualize JSON


```
{
    "@odata.context": "/redfish/v1/$metadata#Chassis.Chassis",
    "@odata.etag": "\"1686701329\"",
    "@odata.id": "/redfish/v1/Chassis/Self",
    "@odata.type": "#Chassis.v1_14_0.Chassis",
    "Actions": {
        "#Chassis.Reset": {
            "@Redfish.ActionInfo": "/redfish/v1/Chassis/Self/ResetActionInfo",
            "@Redfish.OperationApplyTimeSupport": {
                "@odata.type": "#Settings.v1_2_2.OperationApplyTimeSupport",
                "MaintenanceWindowDurationInSeconds": 600,
                "MaintenanceWindowResource": {
                    "@odata.id": "/redfish/v1/Chassis/Self"
                },
                "SupportedValues": [
                    "Immediate",
                    "AtMaintenanceWindowStart"
                ]
            },
            "target": "/redfish/v1/Chassis/Self/Actions/Chassis.Reset"
        }
    },
```

12.5 Network DNS

URI: /redfish/v1/Managers/Self/EthernetInterfaces/eth0
Method: GET/PATCH
Example:
After the request is sent to the target, such as S8056, the response status is 200 with response body in JSON format if the request is successful.

GET https://{{bmc}}/redfish/v1/Managers/Self/EthernetInterfaces/eth0 Send

Params Auth Headers (7) Body Pre-req. Tests Settings Cookies

Body Cookies Headers (13) Test Results 200 OK 101 ms 1.62 KB Save as example

Pretty Raw Preview Visualize JSON


```
{
    "@odata.context": "/redfish/v1/$metadata#EthernetInterface.EthernetInterface",
    "@odata.etag": "\"1686701336\"",
    "@odata.id": "/redfish/v1/Managers/Self/EthernetInterfaces/eth0",
    "@odata.type": "#EthernetInterface.v1_6_2.EthernetInterface",
    "AutoNeg": true,
    "DHCPv4": {
        "DHCPEnabled": true
    },
    "DHCPv6": {
        "OperatingMode": "Stateless"
    },
    "Description": "Ethernet Interface eth0",
    "FQDN": "AMIA0021768A178.mitacad.com",
    "FullDuplex": true,
    "HostName": "AMIA0021768A178",
    "IPv4Addresses": [
        {
            "Address": "10.99.6.2",
            "AddressOrigin": "DHCP",
            "Gateway": "10.99.7.254",
            "SubnetMask": "255.255.240.0"
        }
    ],
```

Chapter 13. Reference Links

DMTF Redfish: http://www.dmtf.org/standards/redfishhttp://redfish.dmtf.org/
Mockups: http://redfish.dmtf.org/redfish/

Technical Support

If a problem arises with your system, you should first turn to your dealer for direct support. Your system has most likely been configured or designed by them and they should have the best idea of what hardware and software your system contains. Hence, they should be of the most assistance for you. Furthermore, if you purchased your system from a dealer near you, take the system to them directly to have it serviced instead of attempting to do so yourself (which can have expensive consequence).

If these options are not available for you then MITAC COMPUTING TECHNOLOGY CORPORATION can help. Besides designing innovative and quality products for over a decade, MITAC has continuously offered customers service beyond their expectations. TYAN's website (http://www.tyan.com) provides easy-to-access resources such as in-depth Linux Online Support sections with downloadable Linux drivers and comprehensive compatibility reports for chassis, memory and much more. With all these convenient resources just a few keystrokes away, users can easily find their latest software and operating system components to keep their systems running as powerful and productive as possible. MITAC also ranks high for its commitment to fast and friendly customer support through email. By offering plenty of options for users, MITAC serves multiple market segments with the industry's most competitive services to support them.

Please feel free to contact us directly for this service at tech-support@tyan.com

Help Resources:

1. See the TYAN's website for FAQ's, bulletins, driver updates, and other information: http://www.tyan.com
2. Contact your dealer for help before calling TYAN.

Returning Merchandise for Service

During the warranty period, contact your distributor or system vendor FIRST for any product problems. This warranty only covers normal customer use and does not cover damages incurred during shipping or failure due to the alteration, misuse, abuse, or improper maintenance of products.

TYAN® Redfish Service Engineer's Manual V1.0

Document No.: D2619 - 100