SUPERMICRO®

Redfish®
Reference Guide

Revision 2.0b

The information in this USER'S MANUAL has been carefully reviewed and is believed to be accurate. The vendor assumes no responsibility for any inaccuracies that may be contained in this document, makes no commitment to update or to keep current the information in this manual, or to notify any person organization of the updates. Please Note: For the most up-to-date version of this manual, please see our web site at www.supermicro.com.

Super Micro Computer, Inc. ("Supermicro") reserves the right to make changes to the product described in this manual at any time and without notice. This product, including software, if any, and documentation may not, in whole or in part, be copied, photocopied, reproduced, translated, or reduced to any medium or machine without prior written consent.

IN NO EVENT WILL SUPERMICRO BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, SPECULATIVE, OR CONSEQUENTIAL DAMAGES ARISING FROM THE USE OR INABILITY TO USE THIS PRODUCT OR DOCUMENTATION, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. IN PARTICULAR, SUPERMICRO SHALL NOT HAVE LIABILITY FOR ANY HARDWARE, SOFTWARE, OR DATA STORED OR USED WITH THE PRODUCT, INCLUDING THE COSTS OF REPAIRING, REPLACING, INTEGRATING, INSTALLING, OR RECOVERING SUCH HARDWARE, SOFTWARE, OR DATA.

Any disputes arising between manufacturer and customer shall be governed by the laws of Santa Clara County in the State of California, USA. The State of California, County of Santa Clara shall be the exclusive venue for the resolution of any such disputes. Super Micro's total liability for all claims will not exceed the price paid for the hardware product.

Information in this document is subject to change without notice. Other products and companies referred to herein are trademarks or registered trademarks of their respective companies or mark holders.

Printed in the United States of America

Manual Revision 2.0b

Release Date: November 19, 2019

Revision History

Date	Rev.	Description
August 10, 2015	1.0	Created document.
October 5, 2015	1.0a	Minor formatting modifications.
June 5, 2017	1.0b	Added more APIs to section 2.3. Added and modified list of OEM APIs (Section 3.6). Added content to Section 3.7. Modified screenshots in Chapter 4. Modified reference links in Chapter 5.
March 20, 2018	2.0	Added Section 4 (Update service). Modified Section 3.6 (OEM APIs). Added new APIs in Section 2.3. Added new examples/screenshots in Section 5.
February 7, 2019	2.0a	Formatting modifications. Updated content in Section 2.3. Updated content in Section 3. Moved content from Section 3 to Section 5. Added and deleted content in Section 4. Added content to Section 5.
November 19, 2019	2.0b	Formatting modifications. Added section 5 - Secureboot Added section 6.19 – KCS channel Control Updated section 2.3 – Added Following new APIs - /redfish/v1/Managers/1/KCSInterface - /redfish/v1/Chassis/MRVL.HA-RAID.[contoller_num].StorageModule - /redfish/v1/Chassis/MRVL.HA-RAID.[contoller_num].StorageModule/Drives/Disk.Bay.[disk_num] - /redfish/v1/Systems/1/Storage/MRVL.HA-RAID - /redfish/v1/Systems/1/Storage/MRVL.HA-RAID/Volumes - /redfish/v1/Systems/1/Storage/MRVL.HA-RAID/Volumes/Controller.[controller_num].Volume.[volume_num] - /redfish/v1/Systems/1/Storage/MRVL.HA-RAID/Volumes/Controller.[controller_num].Volume.[volume_num]/Actions/Oem/Volume.DeleteVD - /redfish/v1/Systems/1/Storage/MRVL.HA-RAID/Volumes/Controller.[controller_num].Volume.[volume_num]/Actions/Oem/Volume.RebuildVD - /redfish/v1/Systems/1/Storage/MRVL.HA-RAID/Volumes/Controller.[controller_num].Volume.[volume_num]/Actions/Oem/Volume.ImportVD - /redfish/v1/Systems/1/Storage/MRVL.HA-RAID/Actions/Oem/Storage.CreateVD Updated section 3.2 – User Lockout configuration Updated section 6.2 – Added Boot Order Configuration for System BIOS

		Updated section 6.4 – Added example for Marvel SE9230 RAID configuration

Revision History 3
1 Introduction 7
2 HTTP Request Methods 7
2.1 Responses 8
2.2 HTTP Status Code Description 8
2.3 List of Available APIs 9
3 Using RESTful APIs 15
3.1 Authentication 15
3.1.1 Basic Authentication 15
3.1.2 Session Management 15
3.2 Account Service 17
3.2.1 Create User 17
3.2.2 User Lockout Configuration 18
3.3 Event Service 18
3.3.1 Add Subscription 18
3.3.2 Test Event Subscription 18
3.3.3 View All Subscriptions 19
3.3.4 Delete a Subscription 19
3.4 Registries 19
3.5 Jsonschema 19
4 UpdateService 20
4.1 Update SSL Certificate and Key 20
4.2 BIOS Update 20
4.3 BMC Firmware Update 22
4.4 Simple Update 23
5 Secure Boot 24
5.1 To Enable Redfish Secure Boot Refer to Below APIs 25
5.1.1 /redfish/v1/Systems/1/SecureBoot 25
5.1.2 /redfish/v1/Systems/1/Bios 25
6 Examples 26
6.1 System Reset 26
6.2 Configure the Boot Order in System BIOS 27
6.3 BIOS Configurations: Configure BIOS over Redfish 28

6.4 RAID Management Reference Examples 31

6.5 SMTP 32

A: SMTP SSL authentication Disabled 32

B: SMTP SSL authentication Enabled: 33

6.6 FanMode 33

6.7 Active Directory 33

6.8 Get/Set iKVM Mouse Mode 33

6.9 NTP 34

6.10 RADIUS 34

6.11 LDAP 34

6.12 Snooping 34

6.13 IP Access Control 34

6.14 SMCRAKP 35

6.15 SNMP 35

6.16 Syslog 35

6.17 Chassis Intrusion 35

6.18 IKVM 35

6.19 KCS Channel Control 36

6.20 Acknowledge Event 36

6.21 Getting MAC Address from System NICs 37

6.22 Python Code for Redfish API Response 38

7 Reference Links *38*

1 Introduction

The Redfish Scalable Platforms Management API ("Redfish") is a new interface that uses RESTful interface semantics to access data defined in a model format to perform out-of-band systems management. It is suitable for a wide range of servers, from stand-alone to rack mount and blade environments, but scales equally well for large scale cloud environments.

Redfish is a management standard which uses data model representation inside of a hypermedia RESTful interface. It is based on REST, which is why Redfish is easier to use and implement than many other solutions. Since it is model oriented, it is capable of expressing the relationships between components in modern systems as well as the semantics of the services and components within them. It is also easily extensible. By using a hypermedia approach to REST, Redfish can express a large variety of systems from multiple vendors. Utilizing JSON (JavaScript Object Notation) data format, which is in plain text, allows many types of parameters to be available such that it enables scalability, human readability, and flexibility for most programming environments by easily interpreting payload.

The model is displayed in terms of an interoperable OData Schema with the payload of the messages being expressed in JSON following OData JSON conventions. The schema (available in both XML and JSON formats) includes annotations to facilitate the automatic translation of the schema to JSON Schema. The ability to externally host the schema definition of the resources in a machine-readable format allows the metadata to be associated with the data without encumbering Redfish services with the metadata, thus enabling more advanced client scenarios as found in many data center and cloud environments.

Supermicro enables Redfish feature sets on their X10/X11 platforms with 3.xx and 1.xx BMC firmware respectively. These features are covered under SFT-OOB-LIC and SFT-DCMS-SINGLE license.
This document will provide you with an overview of Restful API services and describe how to receive Redfish API responses directly from a Supermicro BMC (Baseboard Management Controller).

2 HTTP Request Methods

The following HTTP methods are used to implement different actions, as described below.

- Read Requests (GET):

The GET method is used to request a representation of a specified resource. The representation can be either a single resource or a collection.

- Update (PATCH):

The PATCH method is used to apply partial modifications to a resource.

- Replace (PUT):

The PUT method is used to completely replace a resource. Any properties omitted from the body of the request are reset to their default value.

- Create (POST):

The POST method is used to create a new resource. This request is submitted to the resource collection in which the new resource is meant to belong.

- Actions (POST):

The POST method may also be used to initiate operations on the object (Actions). The POST operation may not be idempotent.

- Delete (DELETE):

The DELETE method is used to remove a resource.

2.1 Responses

Four types of responses are supported, as defined below.

- Metadata Responses:

These describe the resources and types exposed by the service to generic clients.

- Resource Responses:

JSON representation of an individual resource.

- Resource Collection Responses:

JSON representation of a collection of resources.

- Error Responses:

Top-level JSON response providing additional information in the case of an HTTP error.

2.2 HTTP Status Code Description

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

2.3 List of Available APIs

API List	**Notes:**
/redfish/v1	Service root
/redfish/v1/SessionService	
/redfish/v1/Chassis	
/redfish/v1/AccountService	
/redfish/v1/Managers	
/redfish/v1/Systems	
/redfish/v1/EventService	
/redfish/v1/UpdateService	
/redfish/v1/Registries	
/redfish/v1/JsonSchemas	
/redfish/v1/SessionService/Sessions	
/redfish/v1/SessionService/Sessions/[session_num]	
/redfish/v1/Chassis/1	
/redfish/v1/Chassis/1/Thermal	
/redfish/v1/Chassis/1/Power	
/redfish/v1/Chassis/HA-RAID.[contoller_num].StorageEnclosure.[enclosure_num]	For LSI 3108
/redfish/v1/Chassis/HA-RAID.[contoller_num].StorageEnclosure.[enclosure_num]/Drives/Disk.Bay.[disk_num]	For LSI 3108
/redfish/v1/Chassis/HA-RAID.[contoller_num].StorageEnclosure.[enclosure_num]/Drives/Disk.Bay.[disk_num]/Actions/Oem/Drive.Indicate	Light on physical drive indication LED: "Active":"true"
/redfish/v1/Chassis/HBA.[contoller_num].StorageEnclosure.[enclosure_num]	For LSI 3008
/redfish/v1/Chassis/HBA.[contoller_num].StorageEnclosure.[enclosure_num]/Drives/Disk.Bay.[disk_num]	For LSI 3008
/redfish/v1/Chassis/HBA.[contoller_num].StorageEnclosure.[enclosure_num]/Drives/Disk.Bay.[disk_num]/Actions/Oem/Drive.Indicate	Light on physical drive indication LED: "Active":"true"
/redfish/v1/Chassis/StorageBackplane	For PCH SATA or RSTe, TAS must be running
/redfish/v1/Chassis/StorageBackplane/Drives/Disk.Bay.[disk_num]	For PCH SATA or RSTe, TAS must be running
/redfish/v1/Chassis/NVMeSSD.[pcie_controller_num].Group.[group_num].StorageBackplane	For NVMe
/redfish/v1/Chassis/NVMeSSD.[pcie_controller_num].Group.[group_num].StorageBackplane/Drives/Disk.Bay.[disk_num]	For NVMe

/redfish/v1/Chassis/MRVL.HA-RAID.[contoller_num].StorageModule	For SE9230
/redfish/v1/Chassis/MRVL.HA-RAID.[contoller_num].StorageModule/Drives/Disk.Bay.[disk_num]	For SE9230
/redfish/v1/AccountService/Roles	
/redfish/v1/AccountService/Roles/Administrator	
/redfish/v1/AccountService/Roles/Operator	
/redfish/v1/AccountService/Roles/ReadOnly	
/redfish/v1/AccountService/Roles/Custom1	
/redfish/v1/AccountService/Accounts	
/redfish/v1/AccountService/Accounts/[account_num]	
/redfish/v1/Managers/1	
/redfish/v1/Managers/1/Actions/Manager.Reset	BMC cold reset
/redfish/v1/Managers/1/Actions/Oem/ManagerConfig.Reset	BMC factory default "Option": "PreserveUser" "Option": "ClearConfig" "Option": "ResetToADMIN"
/redfish/v1/Managers/1/SerialInterfaces	
/redfish/v1/Managers/1/NetworkProtocol	
/redfish/v1/Managers/1/LogServices	
/redfish/v1/Managers/1/LogServices/Log1	
/redfish/v1/Managers/1/LogServices/Log1/Actions/LogService.Reset	Clear event logs
/redfish/v1/Managers/1/LogServices/Log1/Entries	
/redfish/v1/Managers/1/LogServices/Log1/Entries/[log_num]	
/redfish/v1/Managers/1/VM1	
/redfish/v1/Managers/1/VM1/CfgCD	Configure ISO image settings: host, path, username/pass
/redfish/v1/Managers/1/VM1/CfgCD/Actions/IsoConfig.Mount	Mount ISO image
/redfish/v1/Managers/1/VM1/CfgCD/Actions/IsoConfig.UnMount	Unmount ISO image
/redfish/v1/Managers/1/VM1/CD[mounted_dev_num]	User must first mount image
/redfish/v1/Managers/1/VM1/Floppy[mounted_dev_num]	User must first mount image
/redfish/v1/Managers/1/VM1/USB[mounted_dev_num]	User must first mount image
/redfish/v1/Managers/1/EthernetInterfaces	
/redfish/v1/Managers/1/EthernetInterfaces/[eth_num]	
<Managers OEM APIs>	
/redfish/v1/Managers/1/SNMP	

/redfish/v1/Managers/1/SNMP/SNMPv2	
/redfish/v1/Managers/1/SNMP/SNMPv3	
/redfish/v1/Managers/1/FanMode	
/redfish/v1/Managers/1/MouseMode	
/redfish/v1/Managers/1/Snooping	
/redfish/v1/Managers/1/ActiveDirectory	
/redfish/v1/Managers/1/ActiveDirectory/RoleGroups	
/redfish/v1/Managers/1/ActiveDirectory/RoleGroups/[role_group]	
/redfish/v1/Managers/1/SMTP	
/redfish/v1/Managers/1/Syslog	
/redfish/v1/Managers/1/RADIUS	
/redfish/v1/Managers/1/LDAP	
/redfish/v1/Managers/1/SMCRAKP	
/redfish/v1/Managers/1/IPAccessControl	
/redfish/v1/Managers/1/IPAccessControl/FilterRule	
/redfish/v1/Managers/1/IPAccessControl/FilterRule/[rule_num]	
/redfish/v1/Managers/1/NTP	
/redfish/v1/Managers/1/IKVM	Get a URL link to launch iKVM/HTML5
/redfish/v1/Managers/1/KCSInterface	
/redfish/v1/Systems/1	
/redfish/v1/Systems/1/Actions/ComputerSystem.Reset	System reset
/redfish/v1/Systems/1/Processors	
/redfish/v1/Systems/1/Processors/[processor_num]	
/redfish/v1/Systems/1/Memory	
/redfish/v1/Systems/1/Memory/[memory_num]	
/redfish/v1/Systems/1/EthernetInterfaces	
/redfish/v1/Systems/1/EthernetInterfaces/[eth_num]	Data from BIOS and TAS
/redfish/v1/Systems/1/SimpleStorage	
/redfish/v1/Systems/1/SimpleStorage/[controller_num]	
/redfish/v1/Systems/1/Storage	
/redfish/v1/Systems/1/Storage/HA-RAID	For LSI 3108
/redfish/v1/Systems/1/Storage/HA-RAID/Volumes	For LSI 3108
/redfish/v1/Systems/1/Storage/HA-RAID/Volumes/Controller.[controller_num].Volume.[volume_num]	For LSI 3108

/redfish/v1/Systems/1/Storage/HA-RAID/Volumes/Controller.[controller_num].Volume.[volume_num]/Actions/Oem/Volume.Indicate	For LSI 3108; light on virtual drive indication LED: "Active":"true"
/redfish/v1/Systems/1/Storage/HA-RAID/Volumes/Controller.[controller_num].Volume.[volume_num]/Actions/Oem/Volume.Delete	For LSI 3108; in logical view to delete specific virtual drive
/redfish/v1/Systems/1/Storage/HA-RAID/Actions/Oem/Storage.CreateVolume	For LSI 3108; create virtual drives
/redfish/v1/Systems/1/Storage/HA-RAID/Actions/Oem/Storage.ClearVolumes	For LSI 3108; in logical view to clear all configurations
/redfish/v1/Systems/1/Storage/HA-RAID/Actions/Oem/HARAIDController.Save	For LSI 3108; save controller's "BIOS Boot Mode"
/redfish/v1/Systems/1/Storage/HBA	For LSI 3008
/redfish/v1/Systems/1/Storage/RAIDIntegrated	For RSTe, TAS must be running
/redfish/v1/Systems/1/Storage/RAIDIntegrated/Volumes	For RSTe, TAS must be running
/redfish/v1/Systems/1/Storage/RAIDIntegrated/Volumes/[volume_num]	For RSTe, TAS must be running
/redfish/v1/Systems/1/Storage/SATAEmbedded	For PCH SATA, TAS must be running
/redfish/v1/Systems/1/Storage/SATAEmbedded/Volumes	For PCH SATA, TAS must be running
/redfish/v1/Systems/1/Storage/SATAEmbedded/Volumes/[volume_num]	For PCH SATA, TAS must be running
/redfish/v1/Systems/1/Storage/MRVL.HA-RAID	For SE9230
/redfish/v1/Systems/1/Storage/MRVL.HA-RAID/Volumes	For SE9230
/redfish/v1/Systems/1/Storage/MRVL.HA-RAID/Volumes/Controller.[controller_num].Volume.[volume_num]	For SE9230
/redfish/v1/Systems/1/Storage/MRVL.HA-RAID/Volumes/Controller.[controller_num].Volume.[volume_num]/Actions/Oem/Volume.DeleteVD	For SE9230, Delete Virtual Drive
/redfish/v1/Systems/1/Storage/MRVL.HA-RAID/Volumes/Controller.[controller_num].Volume.[volume_num]/Actions/Oem/Volume.RebuildVD	For SE9230, Rebuild Virtual Drive
/redfish/v1/Systems/1/Storage/MRVL.HA-RAID/Volumes/Controller.[controller_num].Volume.[volume_num]/Actions/Oem/Volume.ImportVD	For SE9230, Import Virtual Drive
/redfish/v1/Systems/1/Storage/MRVL.HA-RAID/Actions/Oem/Storage.CreateVD	For SE9230, Create Virtual Drive
/redfish/v1/Systems/1/Bios	BIOS current settings; only supports: X11DPT-B, X11DPT-PS, X11DPI, X11DPU

/redfish/v1/Systems/1/Bios/SD	BIOS pending settings; only supports: X11DPT-B, X11DPT-PS, X11DPI, X11DPU
/redfish/v1/Systems/1/Bios/Actions/Bios.ResetBios	Reset BIOS settings to default; only supports: X11DPT-B, X11DPT-PS, X11DPI, X11DPU
/redfish/v1/Systems/1/Bios/Actions/Bios.ChangePassword	Change BIOS booting password; only supports: X11DPT-B, X11DPT-PS, X11DPI, X11DPU
/redfish/v1/Systems/1/SmcNodeManager	View SMC node manager policies
/redfish/v1/Systems/1/SmcNodeManager/Actions/SmcNodeManager.ClearAllPolicies	Clear SMC node manager policies
/redfish/v1/Systems/1/LogServices	System logs
/redfish/v1/Systems/1/LogServices/Log1	System logs
/redfish/v1/Systems/1/LogServices/Log1/Actions/LogService.ClearLog	Clear system management logs
/redfish/v1/Systems/1/LogServices/Log1/Actions/Oem/LogService.ClearAcknowledgements	Clear system log acknowledgements
/redfish/v1/Systems/1/LogServices/Log1/Entries	View system log entries
/redfish/v1/Systems/1/LogServices/Log1/Entries/[log_num]	Log entry details Patch to acknowledge
/redfish/v1/Systems/1/PCIeDevices/NIC[aoc_card_num]	Asset information of AOC NIC cards
/redfish/v1/Systems/1/PCIeDevices/NIC[aoc_card_num]/Functions/[port_num]	Asset information of each AOC's NIC chip
/redfish/v1/Systems/1/PCIeDevices/GPU[gpu_card_num]	Asset information of GPU cards
/redfish/v1/Systems/1/PCIeDevices/GPU[gpu_card_num]/Functions/[gpu_instance_num]	Detail information of GPU cards
/redfish/v1/Systems/1/SecureBoot/	BIOS secureboot settings (Only X11DP supports)
/redfish/v1/Systems/1/SecureBoot/Actions/SecureBoot.ResetKeys	Reset key for secure boot (Only X11DP supports)
/redfish/v1/EventService/Subscriptions	
/redfish/v1/EventService/Subscriptions/[destination_num]	
/redfish/v1/UpdateService/Actions/UpdateService.SimpleUpdate	Only X11DP supports
/redfish/v1/UpdateService/SimpleUpdateActionInfo	Only X11DP supports
/redfish/v1/UpdateService/IPMIConfig	

/redfish/v1/UpdateService/IPMIConfig/Actions/IPMIConfig.Upload	Upload new IPMI configuration file to set BMC
/redfish/v1/UpdateService/IPMIConfig/Actions/IPMIConfig.Download	Download IPMI configuration as a file
/redfish/v1/UpdateService/SSLCert	View current SSL certification info
/redfish/v1/UpdateService/SSLCert/Actions/SSLCert.Upload	Used to upload new SSL certification file
/redfish/v1/UpdateService/SmcFirmwareInventory	Supported on X11 platforms
/redfish/v1/UpdateService/SmcFirmwareInventory/BMC	
/redfish/v1/UpdateService/SmcFirmwareInventory/BMC/Actions/SmcFirmwareInventory.EnterUpdateMode	
/redfish/v1/UpdateService/SmcFirmwareInventory/BMC/Actions/SmcFirmwareInventory.Upload	
/redfish/v1/UpdateService/SmcFirmwareInventory/BMC/Actions/SmcFirmwareInventory.Update	"PreserveCfg":"true", "PreserveSdr":"true", "PreserveSsl":"true"
/redfish/v1/UpdateService/SmcFirmwareInventory/BMC/Actions/SmcFirmwareInventory.Cancel	
/redfish/v1/UpdateService/SmcFirmwareInventory/BIOS	
/redfish/v1/UpdateService/SmcFirmwareInventory/BIOS/Actions/SmcFirmwareInventory.EnterUpdateMode	
/redfish/v1/UpdateService/SmcFirmwareInventory/BIOS/Actions/SmcFirmwareInventory.Upload	
/redfish/v1/UpdateService/SmcFirmwareInventory/BIOS/Actions/SmcFirmwareInventory.Update	"PreserveME":"true", "PreserveNVRAM":"true", "PreserveSMBIOS":"true"
/redfish/v1/UpdateService/SmcFirmwareInventory/BIOS/Actions/SmcFirmwareInventory.Cancel	Only X11DP supports
/redfish/v1/Registries/Base.v1_4_0	
/redfish/v1/Registries/BiosAttributeRegistry.v1_0_0.json	Only X11DP supports
/redfish/v1/JsonSchemas/[variety_of_services]	

Revised API

Old	New
/redfish/v1/AccountService/Roles/Admin	/redfish/v1/AccountService/Roles/Administrator
/redfish/v1/AccountService/Roles/ReadOnlyUser	/redfish/v1/AccountService/Roles/ReadOnly
/redfish/v1/UpdateService/FirmwareInventory	/redfish/v1/UpdateService/SmcFirmwareInventory
/redfish/v1/UpdateService/FirmwareInventory/BMC	/redfish/v1/UpdateService/SmcFirmwareInventory/BMC
/redfish/v1/UpdateService/FirmwareInventory/BMC/Actions/Oem/FirmwareInventory.EnterBMCUpdateMode	/redfish/v1/UpdateService/SmcFirmwareInventory/BMC/Actions/SmcFirmwareInventory.EnterUpdateMode

/redfish/v1/UpdateService/FirmwareInventory/BMC/Actions/Oem/FirmwareInventory.UploadBMC	/redfish/v1/UpdateService/SmcFirmwareInventory/BMC/Actions/SmcFirmwareInventory.Upload
/redfish/v1/UpdateService/FirmwareInventory/BMC/Actions/Oem/FirmwareInventory.UpdateBMC	/redfish/v1/UpdateService/SmcFirmwareInventory/BMC/Actions/SmcFirmwareInventory.Update
/redfish/v1/UpdateService/FirmwareInventory/BMC/Actions/Oem/FirmwareInventory.CancelBMC	/redfish/v1/UpdateService/SmcFirmwareInventory/BMC/Actions/SmcFirmwareInventory.Cancel
/redfish/v1/UpdateService/FirmwareInventory/BIOS	/redfish/v1/UpdateService/SmcFirmwareInventory/BIOS
/redfish/v1/UpdateService/FirmwareInventory/BIOS/Actions/Oem/FirmwareInventory.EnterBIOSUpdateMode	/redfish/v1/UpdateService/SmcFirmwareInventory/BIOS/Actions/SmcFirmwareInventory.EnterUpdateMode
/redfish/v1/UpdateService/FirmwareInventory/BIOS/Actions/Oem/FirmwareInventory.UploadBIOS	/redfish/v1/UpdateService/SmcFirmwareInventory/BIOS/Actions/SmcFirmwareInventory.Upload
/redfish/v1/UpdateService/FirmwareInventory/BIOS/Actions/Oem/FirmwareInventory.UpdateBIOS	/redfish/v1/UpdateService/SmcFirmwareInventory/BIOS/Actions/SmcFirmwareInventory.Update
/redfish/v1/UpdateService/FirmwareInventory/BIOS/Actions/Oem/FirmwareInventory.CancelBIOS	/redfish/v1/UpdateService/SmcFirmwareInventory/BIOS/Actions/SmcFirmwareInventory.Cancel
/redfish/v1/Registries/Base.1.0.0	/redfish/v1/Registries/Base.v1_4_0
/redfish/v1/Registries/BiosAttributeRegistry.1.0.0.json	/redfish/v1/Registries/BiosAttributeRegistry.v1_0_0.json

3 Using RESTful APIs

The user can receive API responses through programming by installing Postman or any other Rest API client application(s).

3.1 Authentication

Redfish supports both "Basic Authentication" and "Redfish Session Login Authentication" (as described below under Session Management). Service does not require a client 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 party authentication service and clients.

Note: Always check the status code once you get a response from the Redfish URL. You can refer to the status code table mentioned above. (All URLs/commands are case sensitive.)

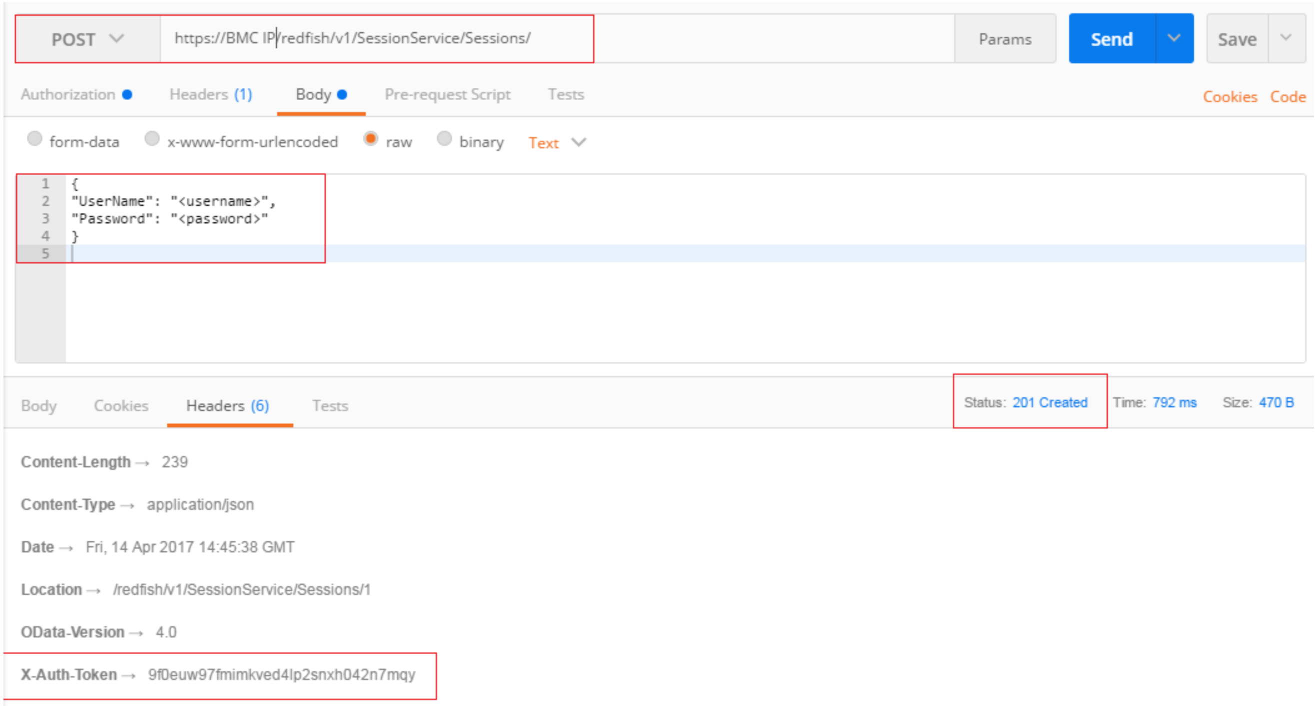
3.1.2 Session Management

Redfish Service uses session management to implement authentication. This includes orphaned session timeouts and a number of simultaneous open sessions.

Step 1: The user can post the following username/password information in the payload field, which will create a new session.

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

Example of applying for Authentication using a Chrome-based app (Advanced Rest Client): The user will receive 201 message code with X-AUTH token created.



POST https://BMC IP/redfish/v1/SessionService/Sessions/

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

Status: 201 Created   Time: 792 ms   Size: 470 B

```
Content-Length → 239
Content-Type → application/json
Date → Fri, 14 Apr 2017 14:45:38 GMT
Location → /redfish/v1/SessionService/Sessions/1
OData-Version → 4.0
X-Auth-Token → 9f0euw97fmimkved4lp2snxh042n7mqy
```

- Users can create a maximum of 16 sessions.
- **Session lifetime**: For Redfish sessions, as long as a client sends 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 30s to 86400s.** If a user is not active in the defined time frame, then the token will be rendered invalid. Users can always patch “SessionTimeout” value if needed.

  Example: [PATCH] https://BMC IP/redfish/v1/SessionService Payload: {"SessionTimeout": 50}
- **Session termination or logout**: A Redfish session is terminated when the client logs out. This is accomplished by performing a DELETE to the session resource identified by the link returned in the location header either when the session was created or if the Session ID is returned in the response data. The ability to DELETE a session by specifying the session resource ID allows an administrator with sufficient privilege to terminate other users’ sessions from a different session.

Example: [DELETE] https://IP/redfish/v1/SessionService/Sessions/2(num) ->Send->Status Code: 200 OK

Log in	Log out
Operation: POST	Operation: DELETE
URI: redfish/v1/SessionService/Sessions/	URI: redfish/v1/SessionService/Sessions/(num)
Request headers: Content-Type: application/json	Request headers: Content-Type: application/json
Request body: {"UserName":"UserName","Password":"Password"}	Requestbody: NONE
Response: 201 created	Response: 200 OK
X-Auth Token header displays Location and session ID ex: Location: /redfish/v1/SessionService/Sessions/5	

Step 2: The response will include an X-Auth-token header with a session token and a location header.
Parse X-Auth token value to get an API response.
Note: The user can apply basic authentication as well.

3.2 Account Service

The user can perform the following operations under /redfish/v1/AccountService.
Methods supported: Get/Post/Patch/Delete

3.2.1 Create User

The user can create a new account using the following API and payload. The user can also delete respective accounts.

[POST]: redfish/v1/AccountService/Accounts/

Payload:

```
{
"UserName":"User_Name",
"Password":"User_Password",
"RoleId":"role_id",        *// Admin, Operator, ReadOnlyUser
"Enabled":true
}
```

The user can also verify assigned privileges for different roles (ADMIN/Operator/Readonlyuser) under redfish/v1/AccountService/Roles.

3.2.2 User Lockout Configuration

[PATCH] redfish/v1/AccountService

Payload:

```
{"AuthFailureLoggingThreshold": 5,
"AccountLockoutThreshold": 2,
"AccountLockoutDuration": 300,
"AccountLockoutCounterResetAfter": 300}
```

3.3 Event Service

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

3.3.1 Add Subscription

Add a subscription to inform Redfish who will receive this event.

[POST]: https://IP/redfish/v1/EventService/Subscriptions/

Payload:

```
{"Destination":"http://www.dnsname.com/Destination1",
"Context":"user1_test",
"EventTypes":["Alert","StatusChange"],
"Protocol":"Redfish"}
```

Destination: Value shall contain a URI to the destination where the events will be sent.
Context: Value is a client-supplied string that is stored with the event destination subscription.
Protocol: This property shall contain the protocol type that the event will use for sending the event to the destination. A value of Redfish shall be used to indicate that the event type shall adhere to that defined in the Redfish specification.
EventTypes: Allowable values

- "StatusChange"
- "ResourceUpdated"
- "ResourceAdded"
- "ResourceRemoved"
- "Alert"

3.3.2 Test Event Subscription

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

[POST]: https://IP/redfish/v1/EventService/Actions/EventService.SendTestEvent

Payload:

```
{"EventType":"Alert"}
```

Users need to implement a RESTful event listener that can receive HTTP or HTTPS POST data that describes the Redfish event format. It can also subscribe to multiple services.

Refer to the Redfish-Event-Listener project page at GitHub to test Event Subscriptions or setup a Redfish Event Listener.

Sample data from Redfish Event Listener:

```
Time:Tue Feb 12 16:49:28 2019 Count:1
Host IP:(‘BMC_IP’, 38486)
Event Details:{'@odata.context': '/redfish/v1/$metadata#EventService/Members/Events/58',
'@odata.id': '/redfish/v1/EventService/Events/58', '@odata.type': '#EventService.v1_0_0.Event', 'Id':
'58', 'Name': 'Event Array', 'Events': [{'EventType': 'Alert', 'Severity': 'OK', 'EventTimestamp': '2019/02/13
00:49:04', 'Message': 'Submit Test Event', 'MessageArgs': ['/redfish/v1/EventService/Actions'],
'MessageId': '0', 'OriginOfCondition': {'@odata.id': '/redfish/v1/EventService'}, 'Context': 'Public'}]}

Time:Tue Feb 12 16:52:24 2019 Count:2
Host IP:('BMC_IP', 38500)
Event Details:{'@odata.context': '/redfish/v1/$metadata#EventService/Members/Events/59',
'@odata.id': '/redfish/v1/EventService/Events/59', '@odata.type': '#EventService.v1_0_0.Event', 'Id':
'59', 'Name': 'Event Array', 'Events': [{'EventType': 'Alert', 'Severity': 'Info', 'EventTimestamp':
'2019/02/13 00:52:00', 'Message': 'Web login was successful.', 'MessageArgs': [''], 'MessageId':
'Alert.1.0.LoginWeb', 'OriginOfCondition': {}, 'Context': 'Public'}]}
```

3.3.3 View All Subscriptions

To see all subscriptions:

[GET]: https://IP/redfish/v1/EventService/Subscriptions/

3.3.4 Delete a Subscription

The user can delete subscription using the Delete request method.

[DELETE]: https://IP/redfish/v1/EventService/Subscriptions/(num)

3.4 Registries

/redfish/v1/Registries/Base.v1_4_0

Registry defines the base messages for Redfish. It represents properties for the registries themselves. The Message ID is formed per the Redfish specification. It consists of the RegistryPrefix concatenated with the version concatenated with the unique identifier for the message registry entry.

3.5 Jsonschema

/redfish/v1/JsonSchemas

The JSON Schema File resource describes the location (URI) of a particular Redfish schema definition being implemented or referenced by a Redfish service.

4 UpdateService

4.1 Update SSL Certificate and Key

Description: Update SSL certificate and key for secure web server connection.

[POST]: https://{BMC_IP}/redfish/v1/UpdateService/SSLCert/Actions/SSLCert.Upload

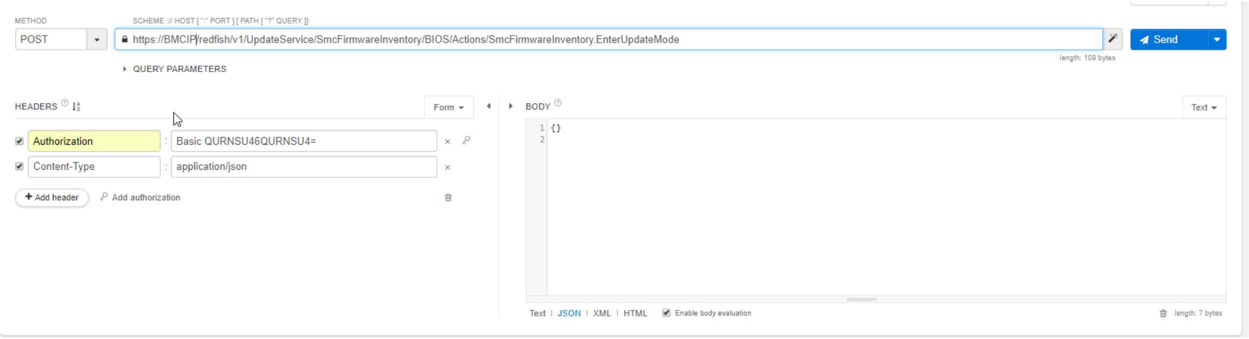
1. Change the type to “form-data”.
2. Select cert_file and key_file as keys and browse respective files to upload-> send.

4.2 BIOS Update

Description: Update BIOS through Redfish API. In the current implementation, the content type must be “multipart/form-data” while uploading the BIOS image.

4.2.1 Enter BIOS update mode by posting the following request and expect to receive a “Successfully Completed Request” response.
[POST]:
https://$BMC_IP/redfish/v1/UpdateService/SmcFirmwareInventory/BIOS/Actions/SmcFirmwareInventory.EnterUpdateMode

Note: The following screenshots are from the Restlet Chrome-based app.

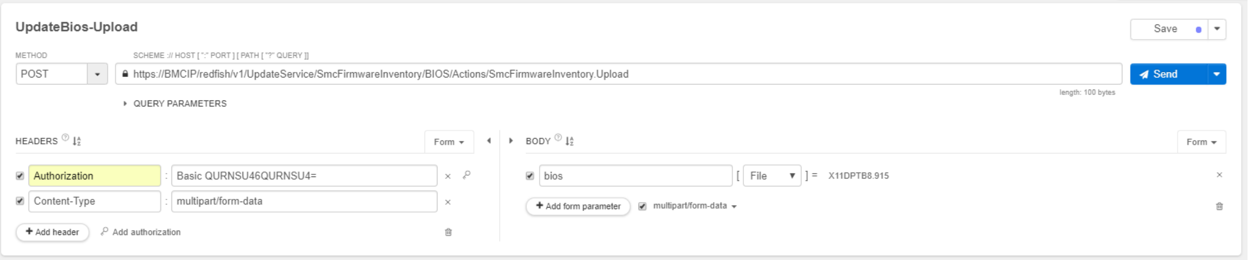


POST https://BMCIP/redfish/v1/UpdateService/SmcFirmwareInventory/BIOS/Actions/SmcFirmwareInventory.EnterUpdateMode

☑ Authorization : Basic QURNSU46QURNSU4=
☑ Content-Type : application/json

```
{}
```

4.2.2 Upload the BIOS image by posting the following request and expect to receive a “Successfully Completed Request” response. The content type must be “multipart/form-data”.
[POST]:
https://$BMC_IP/redfish/v1/UpdateService/SmcFirmwareInventory/BIOS/Actions/SmcFirmwareInventory.Upload



UpdateBios-Upload

METHOD: POST

SCHEME :// HOST [":" PORT] [PATH ["?" QUERY]]: https://BMCIP/redfish/v1/UpdateService/SmcFirmwareInventory/BIOS/Actions/SmcFirmwareInventory.Upload

length: 100 bytes

HEADERS

☑ Authorization : Basic QURNSU46QURNSU4=

☑ Content-Type : multipart/form-data

BODY

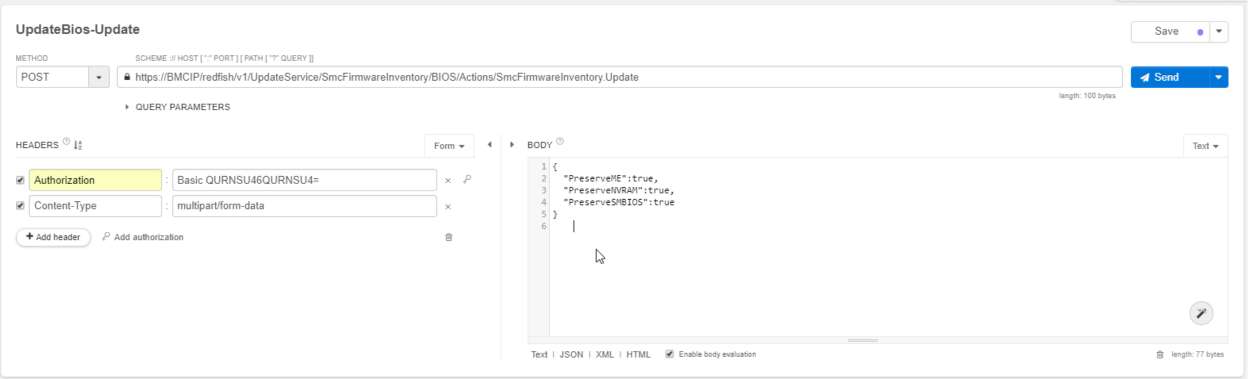
☑ bios [File] = X11DPTB8.915

☑ multipart/form-data

4.2.3 Update BIOS by posting the following request with the following payload and expect to receive a "Successfully Completed Request" response.

[POST]: https://$BMC_IP/redfish/v1/UpdateService/SmcFirmwareInventory/BIOS/Actions/SmcFirmwareInventory.Update

Payload:
{ "PreserveME":true,
 "PreserveNVRAM":true,
 "PreserveSMBIOS":true }



UpdateBios-Update

METHOD: POST

SCHEME :// HOST [":" PORT] [PATH ["?" QUERY]]: https://BMCIP/redfish/v1/UpdateService/SmcFirmwareInventory/BIOS/Actions/SmcFirmwareInventory.Update

length: 100 bytes

HEADERS

☑ Authorization : Basic QURNSU46QURNSU4=

☑ Content-Type : multipart/form-data

BODY

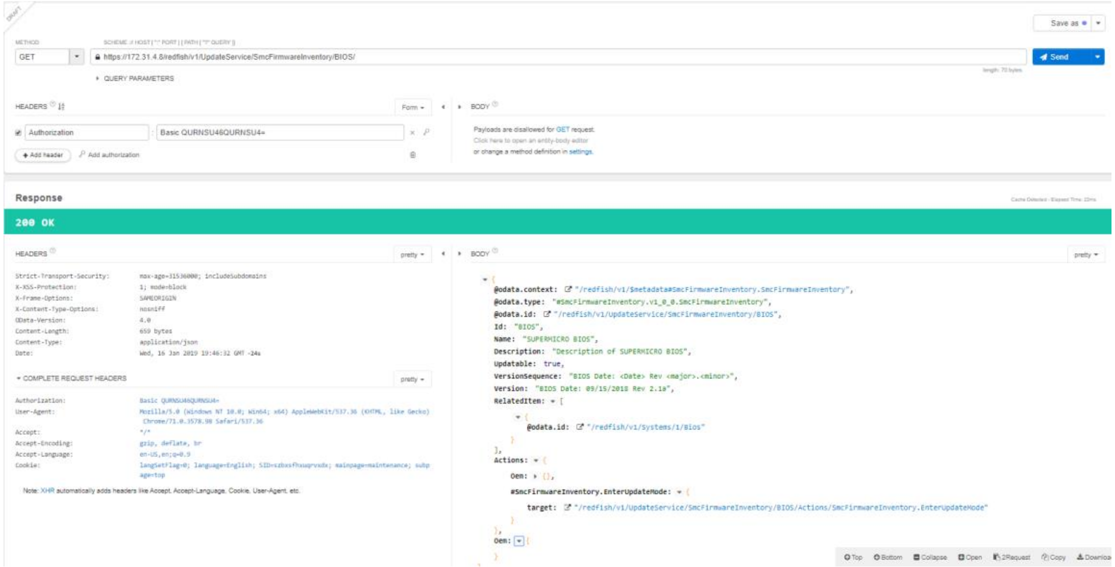
```
{
  "PreserveME":true,
  "PreserveNVRAM":true,
  "PreserveSMBIOS":true
}
```

☑ Enable body evaluation

length: 77 bytes

Check the BIOS update status by issuing the following request with the GET method and expect to receive a response with the BIOS information.

[POST]: https://$BMC_IP/redfish/v1/UpdateService/SmcFirmwareInventory/BIOS/



METHOD: GET

SCHEME :// HOST [":" PORT] [PATH ["?" QUERY]]: https://172.31.4.8/redfish/v1/UpdateService/SmcFirmwareInventory/BIOS/

HEADERS

☑ Authorization : Basic QURNSU46QURNSU4=

Add header   Add authorization

BODY

Payloads are disallowed for GET request.
Click here to open an entity-body editor
or change a method definition in settings.

Response

200 OK

HEADERS

```
Strict-Transport-Security:    max-age=31536000; includeSubdomains
X-XSS-Protection:             1; mode=block
X-Frame-Options:              SAMEORIGIN
X-Content-Type-Options:       nosniff
OData-Version:                4.0
Content-Length:               659 bytes
Content-Type:                 application/json
Date:                         Wed, 16 Jan 2019 19:46:32 GMT -24s
```

COMPLETE REQUEST HEADERS

```
Authorization:     Basic QURNSU46QURNSU4=
User-Agent:        Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
                    Chrome/71.0.3578.98 Safari/537.36
Accept:            */*
Accept-Encoding:   gzip, deflate, br
Accept-Language:   en-US,en;q=0.9
Cookie:            langSetFlag=0; language=English; SID=szbxsfhxuqrvxdx; mainpage=maintenance; subp
                   age=top
```

Note: XHR automatically adds headers like Accept, Accept-Language, Cookie, User-Agent, etc.

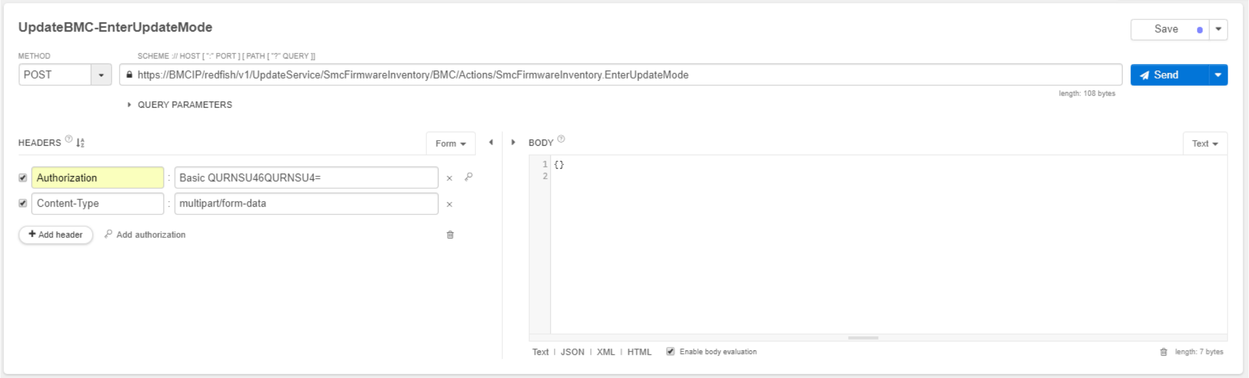
BODY

```
{
  @odata.context: "/redfish/v1/$metadata#SmcFirmwareInventory.SmcFirmwareInventory",
  @odata.type: "#SmcFirmwareInventory.v1_0_0.SmcFirmwareInventory",
  @odata.id: "/redfish/v1/UpdateService/SmcFirmwareInventory/BIOS",
  Id: "BIOS",
  Name: "SUPERMICRO BIOS",
  Description: "Description of SUPERMICRO BIOS",
  Updatable: true,
  VersionSequence: "BIOS Date: <Date> Rev <major>.<minor>",
  Version: "BIOS Date: 09/15/2018 Rev 2.1a",
  RelatedItem: [
    {
      @odata.id: "/redfish/v1/Systems/1/Bios"
    }
  ],
  Actions: {
    Oem: {},
    #SmcFirmwareInventory.EnterUpdateMode: {
      target: "/redfish/v1/UpdateService/SmcFirmwareInventory/BIOS/Actions/SmcFirmwareInventory.EnterUpdateMode"
    }
  },
  Oem: {
  }
}
```

4.3 BMC Firmware Update

Description: Update BMC firmware through the Redfish API. In the current implementation, the content type must be "multipart/form-data" while uploading the BMC image.

4.3.1 Enter BMC update mode by posting the following request and expect to receive a "Successfully Completed Request" response.
[POST]:
https://$BMC_IP/redfish/v1/UpdateService/SmcFirmwareInventory/BMC/Actions/SmcFirmwareInventory.EnterUpdateMode



UpdateBMC-EnterUpdateMode

METHOD: POST

SCHEME :// HOST [":" PORT] [PATH ["?" QUERY]]: https://BMCIP/redfish/v1/UpdateService/SmcFirmwareInventory/BMC/Actions/SmcFirmwareInventory.EnterUpdateMode

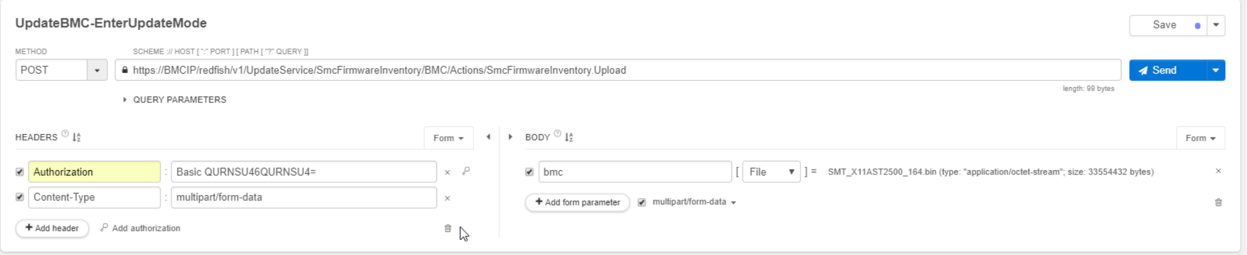
HEADERS

☑ Authorization : Basic QURNSU46QURNSU4=
☑ Content-Type : multipart/form-data

BODY

```
{}
```

4.3.2 Upload the BMC image by issuing the following request with the POST method and expect to receive a "Successfully Completed Request" response. The content type must be "multipart/form-data".
[POST]:
https://$BMC_IP/redfish/v1/UpdateService/SmcFirmwareInventory/BMC/Actions/SmcFirmwareInventory.Upload



UpdateBMC-EnterUpdateMode

METHOD: POST

SCHEME :// HOST [":" PORT] [PATH ["?" QUERY]]

https://BMCIP/redfish/v1/UpdateService/SmcFirmwareInventory/BMC/Actions/SmcFirmwareInventory.Upload

length: 99 bytes

QUERY PARAMETERS

HEADERS

☑	Authorization	Basic QURNSU46QURNSU4=
☑	Content-Type	multipart/form-data

BODY

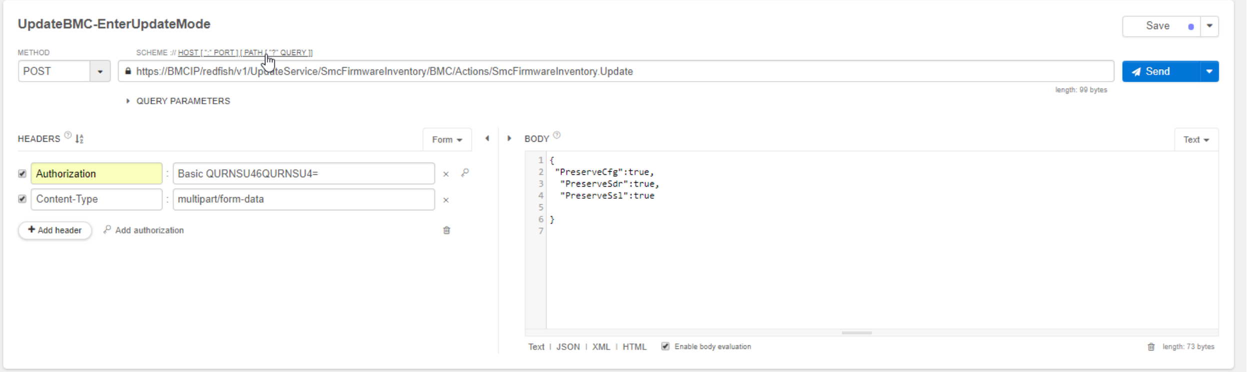
☑ bmc [File ▾] = SMT_X11AST2500_164.bin (type: "application/octet-stream"; size: 33554432 bytes)

☑ multipart/form-data

4.3.2 Update the BMC image by issuing the following request with the POST method and expect to receive a “Successfully Completed Request” response. The content type must be “multipart/form-data”.

[POST]:
https://$BMC_IP/redfish/v1/UpdateService/SmcFirmwareInventory/BMC/Actions/SmcFirmwareInventory.Update

Payload:
{ "PreserveCfg":true,
 "PreserveSdr":true,
 "PreserveSsl":true }



UpdateBMC-EnterUpdateMode

METHOD: POST

SCHEME :// HOST [":" PORT] [PATH ["?" QUERY]]

https://BMCIP/redfish/v1/UpdateService/SmcFirmwareInventory/BMC/Actions/SmcFirmwareInventory.Update

length: 99 bytes

QUERY PARAMETERS

HEADERS

☑	Authorization	Basic QURNSU46QURNSU4=
☑	Content-Type	multipart/form-data

BODY

```
{
 "PreserveCfg":true,
  "PreserveSdr":true,
  "PreserveSsl":true

}

```

Text | JSON | XML | HTML ☑ Enable body evaluation

length: 73 bytes

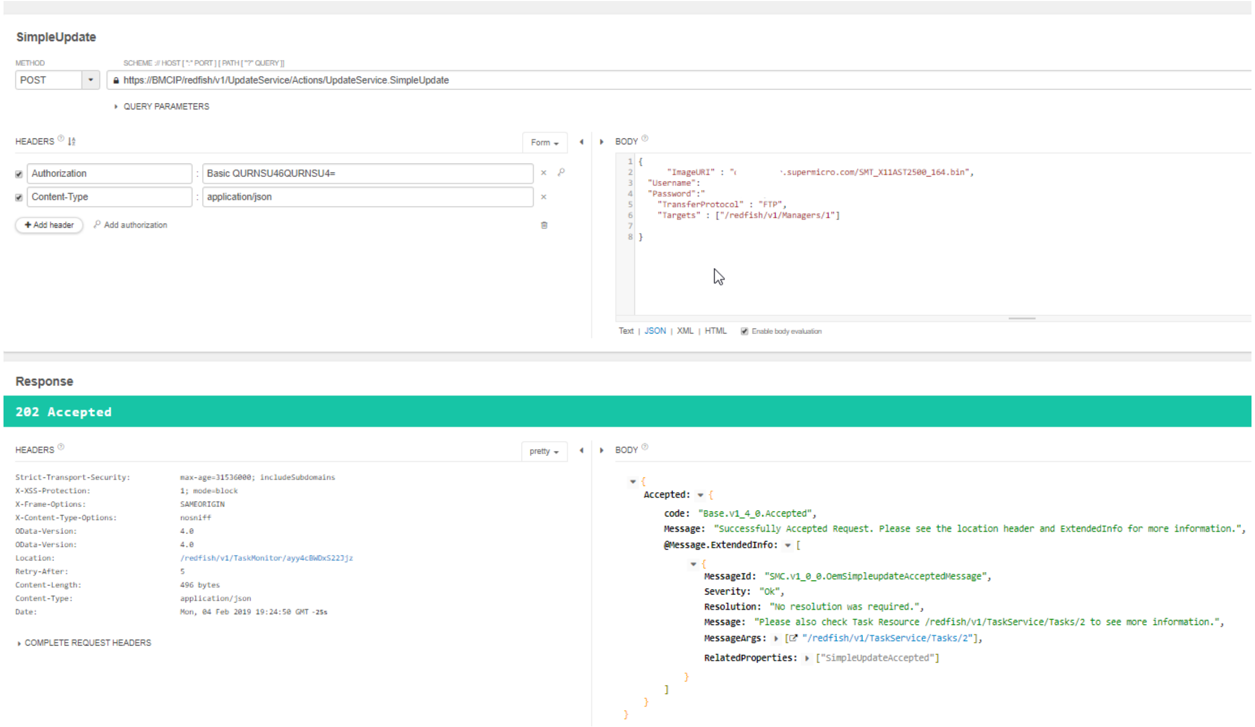
4.4 Simple Update

This action shall perform an update of installed software component(s) as contained within a software image file located at a URI referenced by the ImageURI parameter.

The user is required to prepare FTP, HTTP, or HTTPs file server to put BMC or BIOS firmware image file in.

[POST]: /redfish/v1/UpdateService/Actions/UpdateService.SimpleUpdate
Payload:
{ "ImageURI": "<file ip>/<path and image file name>",
 "TransferProtocol": "FTP",
"Targets": ["/redfish/v1/Managers/1"]
}

Target value:
For BIOS Update use "/redfish/v1/System/1"
For BMC Update use "/redfish/v1/Managers/1"



SimpleUpdate

METHOD: POST

SCHEME :// HOST [":" PORT] [PATH ["?" QUERY]]: https://BMCIP/redfish/v1/UpdateService/Actions/UpdateService.SimpleUpdate

QUERY PARAMETERS

HEADERS

☑ Authorization : Basic QURNSU46QURNSU4=
☑ Content-Type : application/json

+ Add header   Add authorization

BODY

```
{
     "ImageURI" : "        .supermicro.com/SMT_X11AST2500_164.bin",
  "Username":
  "Password":"
    "TransferProtocol" : "FTP",
    "Targets" : ["/redfish/v1/Managers/1"]

}
```

Text | JSON | XML | HTML   ☑ Enable body evaluation

Response

202 Accepted

HEADERS

```
Strict-Transport-Security:    max-age=31536000; includeSubdomains
X-XSS-Protection:             1; mode=block
X-Frame-Options:              SAMEORIGIN
X-Content-Type-Options:       nosniff
OData-Version:                4.0
OData-Version:                4.0
Location:                     /redfish/v1/TaskMonitor/ayy4cBWDxS22Jjz
Retry-After:                  5
Content-Length:               496 bytes
Content-Type:                 application/json
Date:                         Mon, 04 Feb 2019 19:24:50 GMT -25s
```

COMPLETE REQUEST HEADERS

BODY

```
{
  Accepted: {
     code: "Base.v1_4_0.Accepted",
     Message: "Successfully Accepted Request. Please see the location header and ExtendedInfo for more information.",
     @Message.ExtendedInfo: [
        {
          MessageId: "SMC.v1_0_0.OemSimpleupdateAcceptedMessage",
          Severity: "Ok",
          Resolution: "No resolution was required.",
          Message: "Please also check Task Resource /redfish/v1/TaskService/Tasks/2 to see more information.",
          MessageArgs: ["/redfish/v1/TaskService/Tasks/2"],
          RelatedProperties: ["SimpleUpdateAccepted"]
        }
     ]
  }
}
```

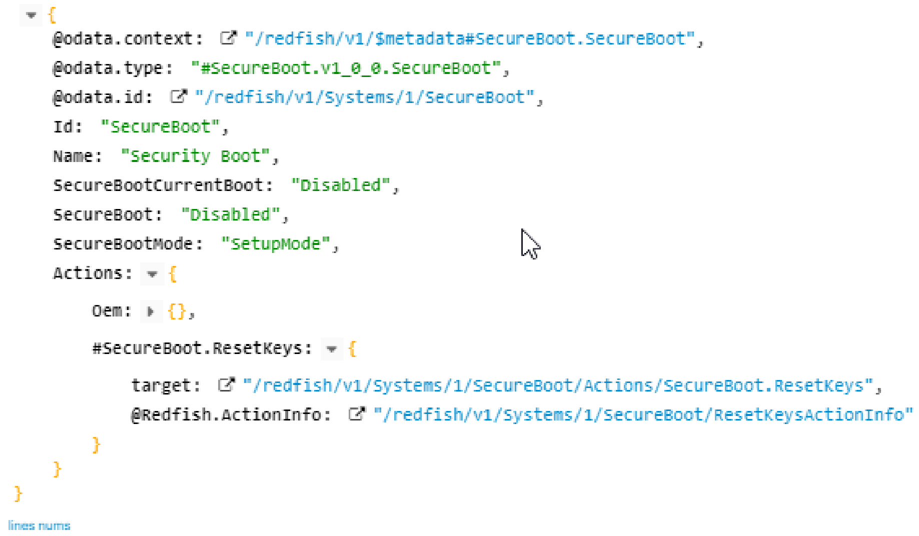
5 Secure Boot

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 supported BIOS for this function.

5.1 To Enable Redfish Secure Boot Refer to Below APIs

5.1.1 /redfish/v1/Systems/1/SecureBoot



```
{
    @odata.context: "/redfish/v1/$metadata#SecureBoot.SecureBoot",
    @odata.type: "#SecureBoot.v1_0_0.SecureBoot",
    @odata.id: "/redfish/v1/Systems/1/SecureBoot",
    Id: "SecureBoot",
    Name: "Security Boot",
    SecureBootCurrentBoot: "Disabled",
    SecureBoot: "Disabled",
    SecureBootMode: "SetupMode",
    Actions: {
        Oem: {},
        #SecureBoot.ResetKeys: {
            target: "/redfish/v1/Systems/1/SecureBoot/Actions/SecureBoot.ResetKeys",
            @Redfish.ActionInfo: "/redfish/v1/Systems/1/SecureBoot/ResetKeysActionInfo"
        }
    }
}
```

Enable SecureBoot
[PATCH]: /redfish/v1/Systems/1/SecureBoot
Payload:
{"SecureBoot":"Enabled"}

Check in Bios/SD.
Check in BIOS setup menu.

ResetKeyTypes
[POST]: /redfish/v1/Systems/1/SecureBoot
{"ResetKeysType":"DeleteAllKeys"}

ResetKeysType Allowable Values:
"ResetAllKeysToDefault",
"DeleteAllKeys",
"DeletePK"

5.1.2 /redfish/v1/Systems/1/Bios

Set the 3 attributes below to BIOS to enable Secure Boot.
[POST]: /redfish/v1/Systems/1/Bios
Payload:
{"SecureBoot":"Enabled",
"SecureBootMode":"User",
"ResetKeysType":"Delete PK Key"}

SecureBoot Allowable Values:
"Enabled”,
"Disabled”
SecureBootMode allowable values:
"Setup",
"User",
"Audit",
"Deployed"

ResetKeyType allowable values:
"Disabled",
"Reset all keys to default",
"Delete all keys",
"Delete PK key"

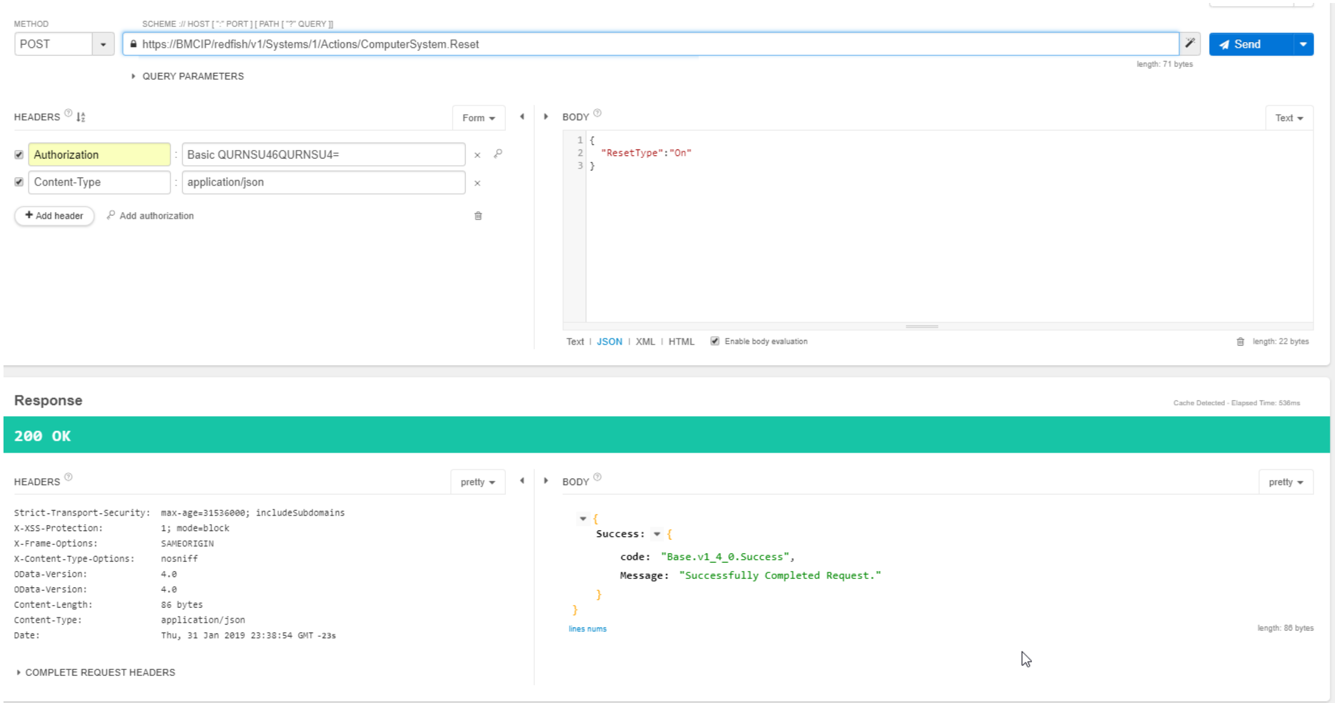
6 Examples

Users can integrate current APIs into their software and applications in order to receive all services provided by Redfish APIs.

6.1 System Reset

[POST]: https://$BMC_IP/redfish/v1/Systems/1/Actions/ComputerSystem.Reset

ResetType: AllowableValues
:[
"On",
"ForceOff",
"GracefulShutdown",
"GracefulRestart",
"ForceRestart",
"Nmi",
"ForceOn"
]



POST https://BMCIP/redfish/v1/Systems/1/Actions/ComputerSystem.Reset

☑ Authorization : Basic QURNSU46QURNSU4=
☑ Content-Type : application/json

```
{
  "ResetType":"On"
}
```

200 OK

```
Strict-Transport-Security:  max-age=31536000; includeSubdomains
X-XSS-Protection:           1; mode=block
X-Frame-Options:            SAMEORIGIN
X-Content-Type-Options:     nosniff
OData-Version:              4.0
OData-Version:              4.0
Content-Length:             86 bytes
Content-Type:               application/json
Date:                       Thu, 31 Jan 2019 23:38:54 GMT -23s
```

```
{
  Success: {
      code: "Base.v1_4_0.Success",
      Message: "Successfully Completed Request."
  }
}
```

6.2 Configure the Boot Order in System BIOS

Description: Change system boot order using Redfish.

1) **BootSourceOverrideEnabled:** Describes the state of the Boot Source Override feature.
   - **Disabled**: The system will boot normally.
   - **Once**: The system will boot (one time) to the Boot Source OverrideTarget.
   - **Continuous**: The system will boot to the target specified in the Boot SourceOverrideTarget until this property is set to Disabled.
2) **BootSourceOverrideMode:** The BIOS Boot Mode (either Legacy or UEFI) to be used when BootSourceOverrideTarget boot source is booted from.
   - **Legacy:** The system will boot in non-UEFI boot mode to the Boot Source Override Target.
   - **UEFI:** The system will boot in UEFI boot mode to the Boot Source Override Target.
3) **BootSourceOverrideTarget:** The current boot source to be used at next boot instead of the normal boot device if BootSourceOverrideEnabled is true.

Below values are allowed to set BootSourceOverrideTarget

when BootSourceOverRideMode: UEFI

- PXE
- CD
- USB
- HDD

when BootSourceOverRideMode: Legacy

- None
- PXE
- CD
- Floppy
- USB
- HDD

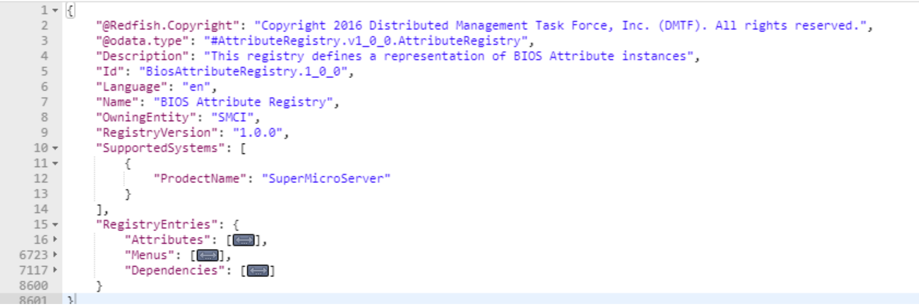
Example: Change BootSourceOverrideTarget to BiosSetup.

```
[PATCH]: redfish/v1/Systems/1
{  "Boot":{
 "BootSourceOverrideEnabled":"Once",
 "BootSourceOverrideMode":"Legacy",
 "BootSourceOverrideTarget": "BiosSetup"
 }
}
```

6.3 BIOS Configurations: Configure BIOS over Redfish

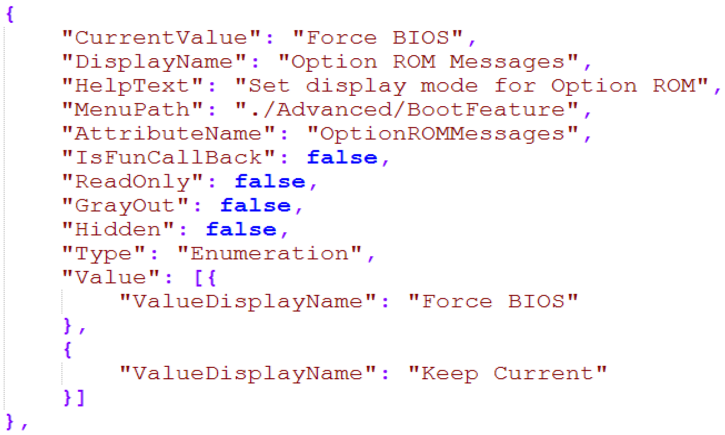
BIOS registry will show Menu of Key (Menus), Keys (Attributes), and Keys' dependencies (Dependencies).

[GET]: https://$BMC_IP/redfish/v1/Registries/BiosAttributeRegistry.v1_0_0



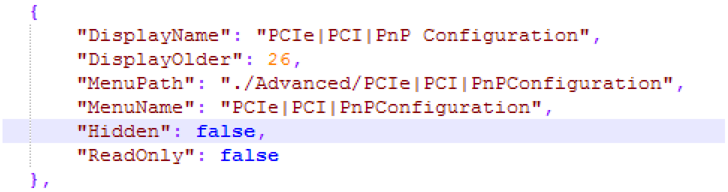
```
{
    "@Redfish.Copyright": "Copyright 2016 Distributed Management Task Force, Inc. (DMTF). All rights reserved.",
    "@odata.type": "#AttributeRegistry.v1_0_0.AttributeRegistry",
    "Description": "This registry defines a representation of BIOS Attribute instances",
    "Id": "BiosAttributeRegistry.1_0_0",
    "Language": "en",
    "Name": "BIOS Attribute Registry",
    "OwningEntity": "SMCI",
    "RegistryVersion": "1.0.0",
    "SupportedSystems": [
        {
            "ProdectName": "SuperMicroServer"
        }
    ],
    "RegistryEntries": {
        "Attributes": [↔],
        "Menus": [↔],
        "Dependencies": [↔]
    }
}
```

Attributes: containing the attributes and their possible values.



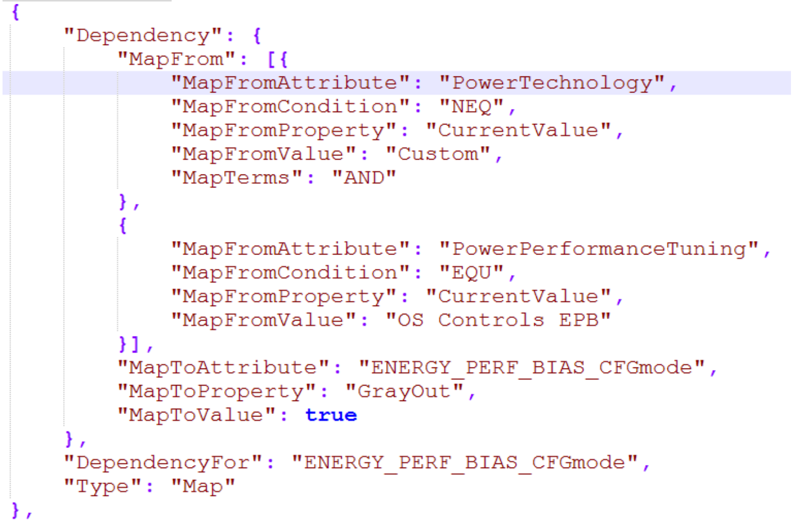
```
{
    "CurrentValue": "Force BIOS",
    "DisplayName": "Option ROM Messages",
    "HelpText": "Set display mode for Option ROM",
    "MenuPath": "./Advanced/BootFeature",
    "AttributeName": "OptionROMMessages",
    "IsFunCallBack": false,
    "ReadOnly": false,
    "GrayOut": false,
    "Hidden": false,
    "Type": "Enumeration",
    "Value": [{
        "ValueDisplayName": "Force BIOS"
    },
    {
        "ValueDisplayName": "Keep Current"
    }]
},
```

Menu: containing the attributes menus and their hierarchy.



```
{
    "DisplayName": "PCIe|PCI|PnP Configuration",
    "DisplayOlder": 26,
    "MenuPath": "./Advanced/PCIe|PCI|PnPConfiguration",
    "MenuName": "PCIe|PCI|PnPConfiguration",
    "Hidden": false,
    "ReadOnly": false
},
```

Dependencies: a list of dependencies of attributes on this component.



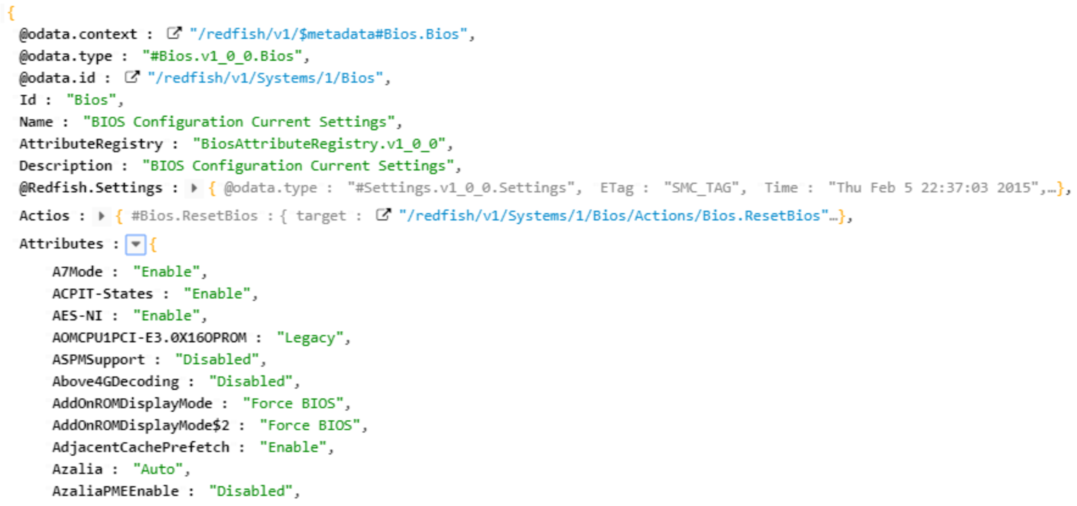
```
{
    "Dependency": {
        "MapFrom": [{
            "MapFromAttribute": "PowerTechnology",
            "MapFromCondition": "NEQ",
            "MapFromProperty": "CurrentValue",
            "MapFromValue": "Custom",
            "MapTerms": "AND"
        },
        {
            "MapFromAttribute": "PowerPerformanceTuning",
            "MapFromCondition": "EQU",
            "MapFromProperty": "CurrentValue",
            "MapFromValue": "OS Controls EPB"
        }],
        "MapToAttribute": "ENERGY_PERF_BIAS_CFGmode",
        "MapToProperty": "GrayOut",
        "MapToValue": true
    },
    "DependencyFor": "ENERGY_PERF_BIAS_CFGmode",
    "Type": "Map"
},
```

Example: If (PowerTechnology’s CurrentValue != “Custom” **AND**
PowerPerformanceTuning’s CurrentValue == “OS Controls EPB”)
ENERGY_PERF_BIAS_CFGmode’s GrayOut = true

Modify attributes:

The user can GET current setting and PATCH desired settings.

[PATCH]: https://$BMC_IP/redfish/v1/Systems/1/Bios

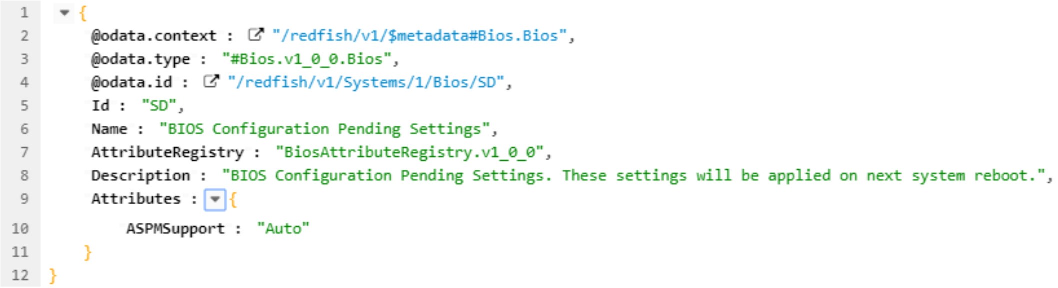


```
{
 @odata.context :  "/redfish/v1/$metadata#Bios.Bios",
 @odata.type :  "#Bios.v1_0_0.Bios",
 @odata.id :  "/redfish/v1/Systems/1/Bios",
 Id :  "Bios",
 Name :  "BIOS Configuration Current Settings",
 AttributeRegistry :  "BiosAttributeRegistry.v1_0_0",
 Description :  "BIOS Configuration Current Settings",
 @Redfish.Settings : ▸ { @odata.type :  "#Settings.v1_0_0.Settings",  ETag :  "SMC_TAG",  Time :  "Thu Feb 5 22:37:03 2015",…},
 Actios : ▸ { #Bios.ResetBios : { target :  "/redfish/v1/Systems/1/Bios/Actions/Bios.ResetBios"…},
 Attributes : ▾{
     A7Mode :  "Enable",
     ACPIT-States :  "Enable",
     AES-NI :  "Enable",
     AOMCPU1PCI-E3.0X16OPROM :  "Legacy",
     ASPMSupport :  "Disabled",
     Above4GDecoding :  "Disabled",
     AddOnROMDisplayMode :  "Force BIOS",
     AddOnROMDisplayMode$2 :  "Force BIOS",
     AdjacentCachePrefetch :  "Enable",
     Azalia :  "Auto",
     AzaliaPMEEnable :  "Disabled",
```

View pending settings:

The user can view any pending setting after PATCH. After PATCH, the user needs to reset the system to apply values to BIOS.

[GET]: https://$BMC_IP/redfish/v1/Systems/1/Bios/SD



```
▾ {
     @odata.context :  "/redfish/v1/$metadata#Bios.Bios",
     @odata.type :  "#Bios.v1_0_0.Bios",
     @odata.id :  "/redfish/v1/Systems/1/Bios/SD",
     Id :  "SD",
     Name :  "BIOS Configuration Pending Settings",
     AttributeRegistry :  "BiosAttributeRegistry.v1_0_0",
     Description :  "BIOS Configuration Pending Settings. These settings will be applied on next system reboot.",
     Attributes : ▾{
         ASPMSupport :  "Auto"
     }
 }
```

BIOS Reset:

POST a reset of the BIOS attributes to default values. After POST, the user needs to reset the system to apply values to BIOS.

[POST]: https://$BMC_IP /redfish/v1/Systems/1/Bios/Actions/Bios.ResetBios"

Change BIOS booting password:

After POST, the user needs to reset the system to apply values to BIOS.

[POST]: https://$BMC_IP /redfish/v1/Systems/1/Bios/Actions/Bios.ChangePassword"
Payload: {
"PasswordName":"AdminPassword"/"UserPassword",
"OldPassword":"",
 "NewPassword":"ADMIN"}

Note: Please use supported BIOS for this function.

6.4 RAID Management Reference Examples

Create LSI3108 Volume	URL: ${BMC_IP}/redfish/v1/Systems/1/Storage/HA-RAID/Actions/Oem/Storage.CreateVolume
	Method: post
	Example Body: {
	"ControllerId":0,
	"Raid": "RAID0",
	"Span": 1,
	"PhysicalDrives":["HA-RAID.0.Disk.0", "HA-RAID.0.Disk.1"],
	"UsePercentage":100,
	"LogicalDriveCount":1,
	"StripSizePerDDF":"256K",
	"LdReadPolicy":"NoReadAhead",
	"LdWritePolicy":"WriteBack",
	"LdIOPolicy":"DirectIO",
	"AccessPolicy":"ReadWrite",
	"DiskCachePolicy":"Unchanged",
	"InitState":"NoInit"
	}

Locate physical HDD	URL: ${BMC_IP}/redfish/v1/Chassis/HA-RAID. [contoller_num].StorageEnclosure. [enclosure_num]/Drives/Disk.Bay. [disk_num]/Volume.Indicate
	Method: post
	Example Body: {
	"Active":"true"
	}

Locate logical volume HDD	URL: ${BMC_IP}/redfish/v1/Systems/1/Storage/HA-RAID/HA-RAID. [contoller_num].Volumes/[volume_num]/Actions/OEM/Volume.Indicate
	Method: post
	Example Body: {
	"Active":"true"
	}

Delete logical volume	URL: ${BMC_IP}/redfish/v1/Systems/1/Storage/HA-RAID/HA-RAID. [contoller_num].Volumes/[volume_num]/Actions/OEM/Volume.Delete
	Method: post
	Example Body: {
	}

Clear all logical volumes	URL: ${BMC_IP}/redfish/v1/Systems/1/Storage/HA-RAID/Storage.ClearVolumes
	Method: post
	Example Body: {
	"ControllerId":0
	}

Save HA-Raid controller config	URL: ${BMC_IP} /redfish/v1/Systems/1/Storage/HA-RAID/Actions/Oem/HARAIDController.Save
	Method: patch
	Example Body: {
	"ControllerId":0,
	"BIOSBootMode":"PauseOnError",
	"JBODMode":"Enable"
	}

View Marvell Drive details	URL: ${BMC_IP}/redfish/v1/Systems/1/Storage/MRVL.HA-RAID **Method: get**

Create Virtual Drive for Marvell	URL: ${BMC_IP}/redfish/v1/Systems/1/Storage/MRVL.HA-RAID/Actions/Oem/Storage.CreateVD
	Method: post
	Example Body: {
	"PD":["MRVL.HA-RAID.0.StorageModule/Drives/Disk.Bay.0","MRVL.HA-RAID.0.StorageModule/Drives/Disk.Bay.1"],
	"RaidLevel":"RAID1",
	"StripeBlock":"64K",
	"VDName":"SuperDrive"
	}

Delete Virtual Drive for Marvell	URL: ${BMC_IP} redfish/v1/Systems/1/Storage/MRVL.HA-RAID/Volumes/Controller.0.Volume.0/Actions/Oem/Volume.DeleteVD **Method: post** Example Body: {}
Rebuild Virtual Drive for Marvell	URL: ${BMC_IP} redfish/v1/Systems/1/Storage/MRVL.HA-RAID/Volumes/Controller.0.Volume.0/Actions/Oem/Volume.RebuildVD **Method: post** Example Body: {}
Import Virtual Drive for Marvell	Usage: Insert/import VD and register its UUID to Marvell firmware. Wait for the next system power-on, when this UUID will be registered to Marvell firmware. URL: ${BMC_IP} redfish/v1/Systems/1/Storage/MRVL.HA-RAID/Volumes/Controller.0.Volume.0/Actions/Oem/Volume.RebuildVD **Method: post** Example Body: {}

6.5 SMTP

After applying the configurations, generate any system event to check if an email alert is received.

A: SMTP SSL authentication Disabled:

[PATCH]: redfish/v1/Managers/1/SMTP

Payload:

```
{"SmtpServer":"mailserver_ip or mailserver_name",
"SmtpPortNumber":"server_port",
"SmtpUserName":"",
"SmtpPassword":"",
"SmtpSenderAddress":"sender_email_address"
}
```

B: SMTP SSL authentication Enabled:

[PATCH]: redfish/v1/Managers/1/SMTP

Payload:

```
{"SmtpSSLEnabled": true,
"SmtpServer":"mailserver_ip or mailserver_name",
"SmtpPortNumber":"server_port",
"SmtpUserName":"user_name",
"SmtpPassword":"user_password",
"SmtpSenderAddress":"sender_email_address"
}
```

6.6 FanMode

[PATCH]: redfish/v1/Managers/1/ FanMode

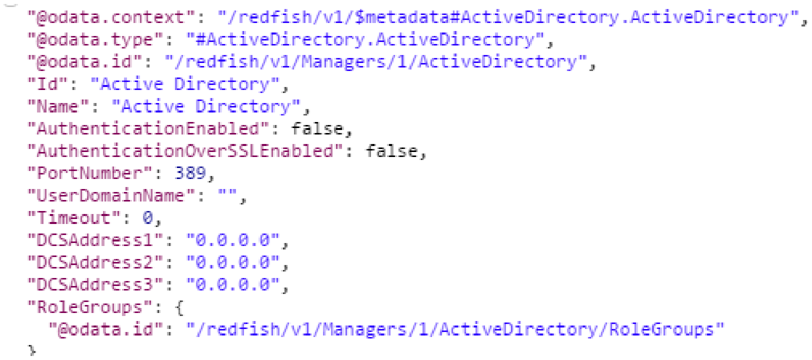
Payload: {"Mode": "FullSpeed"}

Mode Allowable Values:

{"Standard", "FullSpeed", "PUE2", "HeavyIO"}

6.7 Active Directory

The user can PATCH the following properties in order to configure ActiveDirectory.



```
  "@odata.context": "/redfish/v1/$metadata#ActiveDirectory.ActiveDirectory",
  "@odata.type": "#ActiveDirectory.ActiveDirectory",
  "@odata.id": "/redfish/v1/Managers/1/ActiveDirectory",
  "Id": "Active Directory",
  "Name": "Active Directory",
  "AuthenticationEnabled": false,
  "AuthenticationOverSSLEnabled": false,
  "PortNumber": 389,
  "UserDomainName": "",
  "Timeout": 0,
  "DCSAddress1": "0.0.0.0",
  "DCSAddress2": "0.0.0.0",
  "DCSAddress3": "0.0.0.0",
  "RoleGroups": {
    "@odata.id": "/redfish/v1/Managers/1/ActiveDirectory/RoleGroups"
  }
```

[GET]: redfish/v1/Managers/1/ActiveDirectory

[POST]: "redfish/v1/Managers/1/ActiveDirectory/RoleGroups"

Payload:

```
{"RoleGroupName":"xxx",
"RoleGroupDomain":"xxx",
"RoleGroupPrivilege":"Operator"}
```

[GET]/[PATCH]/[DELETE]: "redfish/v1/Managers/1/ActiveDirectory/RoleGroups/ [number]"

6.8 Get/Set iKVM Mouse Mode

It is implemented under redfish/v1/Managers/1/MouseMode.

Methods supported: Get/Patch

Allowable values: "Absolute", "Relative", "Single"

6.9 NTP

Methods supported: Get/Patch

[PATCH]: redfish/v1/Managers/1/NTP

Payload:

```
{"NTPEnable":true,
 "PrimaryNTPServer":"127.0.0.1",
 "SecondaryNTPServer":"localhost",
 "DaylightSavingTime":false}
```

6.10 RADIUS

Methods supported: Get/Patch

[PATCH]: redfish/v1/Managers/1/RADIUS.

Payload:

```
{"RadiusEnabled":true,
 "RadiusServerIP":"127.0.0.1",
 "RadiusPortNumber":1812,
 "RadiusSecret":"SECRET"}
```

6.11 LDAP

Methods supported: Get/Patch

[PATCH]: redfish/v1/Managers/1/LDAP

Payload:

```
{"LDAPEnabled":true,
 "LDAPAuthOverSSL":true,
 "LDAPPortNumber": 389,
"LDAPServerIP" : "0.0.0.0",
 "LDAPPassword" : "password",
 "LDAPDN": "",
 "LDAPSearchbase" : ""}
```

6.12 Snooping

[GET]: https://$BMC_IP/redfish/v1/Managers/1/Snooping

6.13 IP Access Control

It is implemented under redfish/v1/Managers/1/IPAccessControl.

Methods supported: Get/Patch/Post

[PATCH]: https://$BMC_IP/redfish/v1/Managers/1/IPAccessControl

Payload: {"ServiceEnabled": true}

[POST]: https://$BMC_IP/redfish/v1/Managers/1/IPAccessControl/FilterRule
Payload: {“Address": "10.136.176.0", "PrefixLength": 24, "Policy": "Accept"}
[PATCH]: https://$BMC_IP/redfish/v1/Managers/1/IPAccessControl/FilterRule/1
Payload: {“Address": "10.136.176.0", "PrefixLength": 24,"Policy": "Drop"}

6.14 SMCRAKP

Methods supported: Get/Patch
[PATCH]: redfish/v1/Managers/1/SMCRAKP
Payload: {"Mode":"Enabled"}

6.15 SNMP

Methods supported: Get/Patch
[PATCH]: https://$BMC_IP/redfish/v1/Managers/1/SNMP
Payload:
{"SnmpEnabled":true}

[PATCH]: https://$BMC_IP/redfish/v1/Managers/1/SNMP/SNMPv2
Payload: {"Snmpv2Enabled":true,"ROCommunity":"rtest","RWCommunity":"wtest"}

[PATCH]: https://$BMC_IP/redfish/v1/Managers/1/SNMP/SNMPv3
Payload: {"Snmpv3Enabled":true,"UserName":"administrator","AuthProtocol":"SHA1", "PrivateProtocol":"DES", "AuthKey":"Test1234", "PrivateKey":"Test1234"}

6.16 Syslog

Methods supported: Get/Patch
[PATCH]: redfish/v1/Managers/1/Syslog
Payload: {"EnableSyslog": true,"SyslogPortNumber": 514,"SyslogServer": "10.136.176.16"}

6.17 Chassis Intrusion

Methods supported: Get/Patch
[PATCH]: /redfish/v1/Chassis/1
Payload: {"PhysicalSecurity":{"IntrusionSensor": "Normal"}}

6.18 IKVM

Launch HTML5 iKVM using Redfish.
[GET]: {BMC_IP}/redfish/v1/Managers/1/IKVM

Use response property, “URI”, above to prepend “https://${BMC_IP}” and paste this complete URL in browser to render HTML5 iKVM.
Example of launching URL: https://{BMC_IP}/redfish/Kk1D4UVATDja0Jw.IKVM

6.19 KCS Channel Control

This feature allows the user to secure their environment by giving appropriate privilege to access the KCS interface.

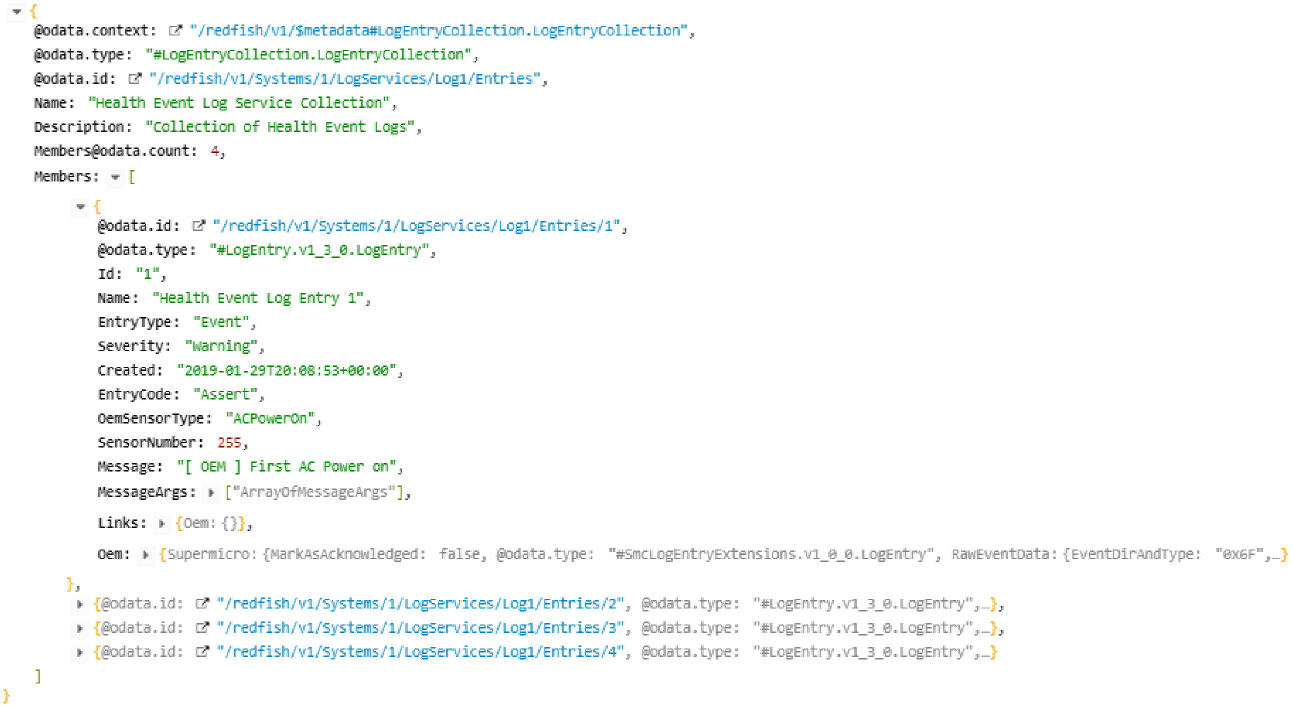
[Administrator]: Any user accessing KCS interface will be able to do all the operations that the Administrator user can do.
[Operator]: Any user accessing the KCS interface will be able to do all the operations that a user with Operator privilege can do.
[User]: Any user accessing the KCS interface will be able to do all the operations that a user with User privilege can do.
[Callback]: This may be considered the lowest privilege level. Only commands necessary to support initiating a Callback are allowed.

[PATCH]: redfish/v1/Managers/1/KCSInterface
Payload: {"Privilege": "Administrator"}

6.20 Acknowledge Event

Description: Acknowledge event using Redfish.
It is implemented under redfish/v1/Systems/1/LogServices.
Methods supported: Get/Patch

- View events
  https://{BMC_IP}/redfish/v1/Systems/1/LogServices/Log1/Entries



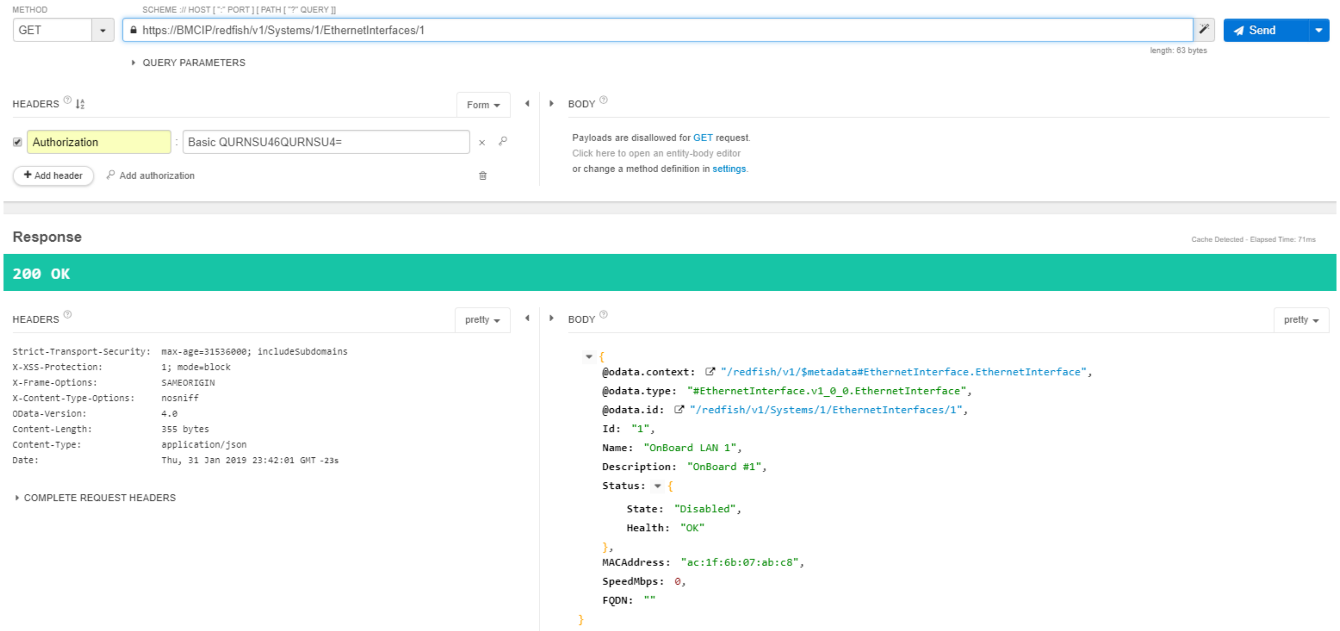
```
{
  @odata.context: "/redfish/v1/$metadata#LogEntryCollection.LogEntryCollection",
  @odata.type: "#LogEntryCollection.LogEntryCollection",
  @odata.id: "/redfish/v1/Systems/1/LogServices/Log1/Entries",
  Name: "Health Event Log Service Collection",
  Description: "Collection of Health Event Logs",
  Members@odata.count: 4,
  Members: [
      {
        @odata.id: "/redfish/v1/Systems/1/LogServices/Log1/Entries/1",
        @odata.type: "#LogEntry.v1_3_0.LogEntry",
        Id: "1",
        Name: "Health Event Log Entry 1",
        EntryType: "Event",
        Severity: "Warning",
        Created: "2019-01-29T20:08:53+00:00",
        EntryCode: "Assert",
        OemSensorType: "ACPowerOn",
        SensorNumber: 255,
        Message: "[ OEM ] First AC Power on",
        MessageArgs: ["ArrayOfMessageArgs"],
        Links: {Oem: {}},
        Oem: {Supermicro: {MarkAsAcknowledged: false, @odata.type: "#SmcLogEntryExtensions.v1_0_0.LogEntry", RawEventData: {EventDirAndType: "0x6F",…}
      },
      {@odata.id: "/redfish/v1/Systems/1/LogServices/Log1/Entries/2", @odata.type: "#LogEntry.v1_3_0.LogEntry",…},
      {@odata.id: "/redfish/v1/Systems/1/LogServices/Log1/Entries/3", @odata.type: "#LogEntry.v1_3_0.LogEntry",…},
      {@odata.id: "/redfish/v1/Systems/1/LogServices/Log1/Entries/4", @odata.type: "#LogEntry.v1_3_0.LogEntry",…}
  ]
}
```

- Acknowledge event: https://{BMC_IP}}/redfish/v1/Systems/1/LogServices/Log1/Entries/1/ [PATCH]:

```
{
   "Oem":{
      "Supermicro":{
          "MarkAsAcknowledged": true
       }
    }
}
```

6.21 Getting MAC Address from System NICs

https://{BMC_IP}/redfish/v1/Systems/1/EthernetInterfaces/1



METHOD: GET

https://BMCIP/redfish/v1/Systems/1/EthernetInterfaces/1

length: 63 bytes

☑ Authorization : Basic QURNSU46QURNSU4=

Payloads are disallowed for GET request.
Click here to open an entity-body editor
or change a method definition in settings.

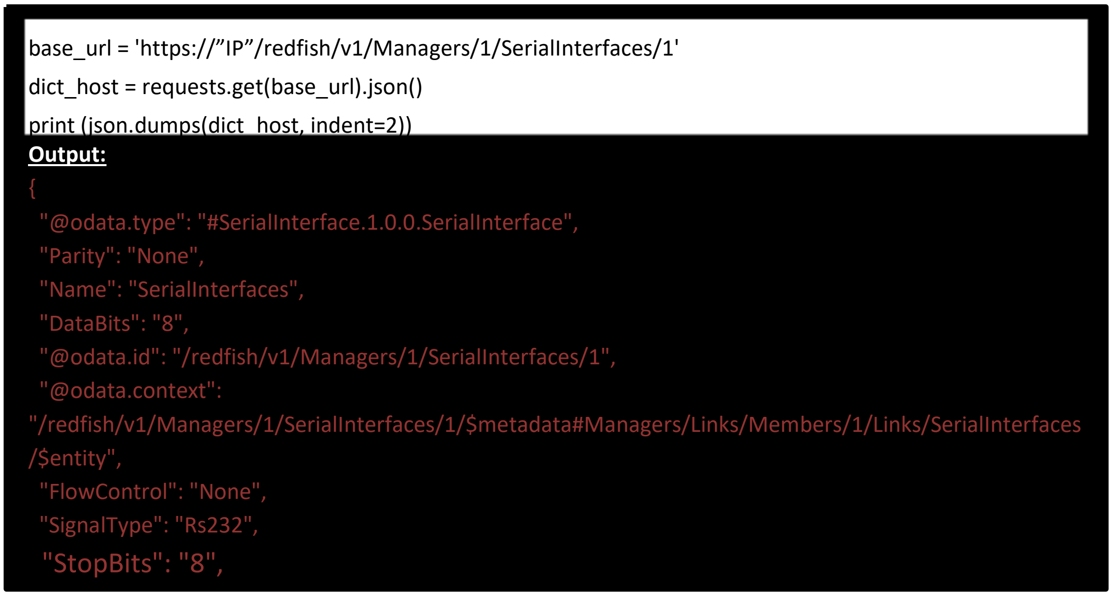
Cache Detected - Elapsed Time: 71ms

200 OK

```
Strict-Transport-Security:  max-age=31536000; includeSubdomains
X-XSS-Protection:           1; mode=block
X-Frame-Options:            SAMEORIGIN
X-Content-Type-Options:     nosniff
OData-Version:              4.0
Content-Length:             355 bytes
Content-Type:               application/json
Date:                       Thu, 31 Jan 2019 23:42:01 GMT -23s
```

```
{
  @odata.context: "/redfish/v1/$metadata#EthernetInterface.EthernetInterface",
  @odata.type: "#EthernetInterface.v1_0_0.EthernetInterface",
  @odata.id: "/redfish/v1/Systems/1/EthernetInterfaces/1",
  Id: "1",
  Name: "OnBoard LAN 1",
  Description: "OnBoard #1",
  Status: {
      State: "Disabled",
      Health: "OK"
  },
  MACAddress: "ac:1f:6b:07:ab:c8",
  SpeedMbps: 0,
  FQDN: ""
}
```

6.22 Python Code for Redfish API Response



```
base_url = 'https://"IP"/redfish/v1/Managers/1/SerialInterfaces/1'
dict_host = requests.get(base_url).json()
print (json.dumps(dict_host, indent=2))
```

Output:

```
{
  "@odata.type": "#SerialInterface.1.0.0.SerialInterface",
  "Parity": "None",
  "Name": "SerialInterfaces",
  "DataBits": "8",
  "@odata.id": "/redfish/v1/Managers/1/SerialInterfaces/1",
  "@odata.context":
"/redfish/v1/Managers/1/SerialInterfaces/1/$metadata#Managers/Links/Members/1/Links/SerialInterfaces
/$entity",
  "FlowControl": "None",
  "SignalType": "Rs232",
  "StopBits": "8",
```

7 Reference Links

Supermicro Redfish:
https://www.supermicro.com/solutions/Redfish.cfm
Supermicro YouTube:
https://www.youtube.com/watch?v=anppU663kUs
DMTF Redfish:
http://www.dmtf.org/standards/redfish
http://redfish.dmtf.org/
Mockups:
http://redfish.dmtf.org/redfish/v1
Contact:
Supermicro Technical Support
support@supermicro.com