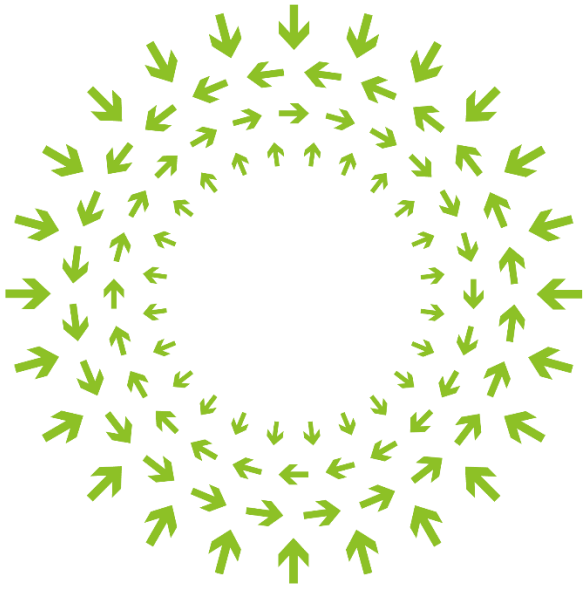


OPEN

Compute Project

Usage Guide and Requirements for the OCP Server Hardware Management API v1.0.0

November 2021

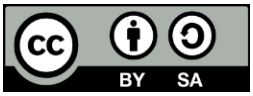
Table of Contents

1. License 4
2. Scope 5
3. Requirements 5
4. Capabilities 5
5. Use Cases 6
   - 5.1. Get the list of systems 6
   - 5.2. Get a system 7
   - 5.3. Get the inventory information 8
   - 5.4. Set the asset tag 8
   - 5.5. Get the location LED 8
   - 5.6. Set the location LED 9
   - 5.7. Get the type of system 9
   - 5.8. Get power state of system 9
   - 5.9. Get status of the computer system 9
   - 5.10. Reset the computer system 10
   - 5.11. Get the revision of the system firmware 10
   - 5.12. Get a summary of the processor information 10
   - 5.13. Get summary of the memory information 11
   - 5.14. Get list of Ethernet interfaces 11
   - 5.15. Get an Ethernet Interface 11
   - 5.16. Get status of Ethernet interface 12
   - 5.17. Get IPv4 address 12
   - 5.18. Set IPv4 Address 13
   - 5.19. Get boot Information 13
   - 5.20. Set the boot device order 14
     - 5.20.1. Default boot order 14
     - 5.20.2. Boot from a UEFI device. 14
     - 5.20.3. Boot from a UEFI HTTP network location. 14
     - 5.20.4. Boot override 14
   - 5.21. Get system log 15
   - 5.22. Get system log entries 15
   - 5.23. Get a system log entry 16
   - 5.24. Clear the system log 16
6. References 16

7. Revision 16

1. License

This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.



NOTWITHSTANDING THE FOREGOING LICENSES, THIS SPECIFICATION IS PROVIDED BY OCP "AS IS" AND OCP EXPRESSLY DISCLAIMS ANY WARRANTIES (EXPRESS, IMPLIED, OR OTHERWISE), INCLUDING IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, FITNESS FOR A PARTICULAR PURPOSE, OR TITLE, RELATED TO THE SPECIFICATION. NOTICE IS HEREBY GIVEN, THAT OTHER RIGHTS NOT GRANTED AS SET FORTH ABOVE, INCLUDING WITHOUT LIMITATION, RIGHTS OF THIRD PARTIES WHO DID NOT EXECUTE THE ABOVE LICENSES, MAY BE IMPLICATED BY THE IMPLEMENTATION OF OR COMPLIANCE WITH THIS SPECIFICATION. OCP IS NOT RESPONSIBLE FOR IDENTIFYING RIGHTS FOR WHICH A LICENSE MAY BE REQUIRED IN ORDER TO IMPLEMENT THIS SPECIFICATION. THE ENTIRE RISK AS TO IMPLEMENTING OR OTHERWISE USING THE SPECIFICATION IS ASSUMED BY YOU. IN NO EVENT WILL OCP BE LIABLE TO YOU FOR ANY MONETARY DAMAGES WITH RESPECT TO ANY CLAIMS RELATED TO, OR ARISING OUT OF YOUR USE OF THIS SPECIFICATION, INCLUDING BUT NOT LIMITED TO ANY LIABILITY FOR LOST PROFITS OR ANY CONSEQUENTIAL, INCIDENTAL, INDIRECT, SPECIAL OR PUNITIVE DAMAGES OF ANY CHARACTER FROM ANY CAUSES OF ACTION OF ANY KIND WITH RESPECT TO THIS SPECIFICATION, WHETHER BASED ON BREACH OF CONTRACT, TORT (INCLUDING NEGLIGENCE), OR OTHERWISE, AND EVEN IF OCP HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

2. Scope

This document references requirements and provide the usage examples for the OCP Baseline Hardware Management API v1.0.1.

3. Requirements

As a Redfish-based interface, the required Redfish interface model elements are specified in a profile document. For the Server Hardware Management API v1.0.0, the profile is located at –

https://github.com/opencomputeproject/HWMgmt-OCP-Profiles/blob/master/OCPServerHardwareManagement.v1_0_0.json

The Redfish Interop Validator is an open-source conformance test which reads the profile, executes the tests against an implementation and generates a test report – in text or HTML format.

```
$> python3 RedfishInteropValidator.py profileName --ip host:port
```

The Redfish Interop Validator is located at *https://github.com/DMTF/Redfish-Interop-Validator*.

The Server Hardware Management v1.0.0 profile extends from the Baseline Hardware Management v1.0.1 profile. This extension is specified directly in the profile. This means that the Server profile specifies conformance to the baseline profile in addition to any requirements specified in the server profile, directly.

```
"RequiredProfiles": {
    "OCPBaselineHardwareManagement": {
        "MinVersion": "1.0.1"
    }
},
```

4. Capabilities

The following use cases are enabled by conformance to this Server Hardware Management profile. The Server Hardware Management profile is extended from the Baseline Hardware Management profile.

The following table lists the capabilities provide the baseline profile requirements. These capabilities are described in the "Usage Guide and Requirements for the OCP Baseline Hardware Management Profile v1.0.1" document.

Table 1 - Baseline Capabilities

Use Case	Manageable Capabilities	Requirement
Account Management	• Get accounts	Mandatory
Session Management	• Get sessions	Mandatory
Hardware inventory	• Get the FRU information • Get and Set the Asset Tag	Mandatory Recommended
Hardware location	• Get the location LED • Set the location LED	Recommended Recommended
Status	• Get status of chassis	Mandatory
Power	• Get power state • Get power usage • Get power limit	Mandatory Recommended Recommended
Temperature	• Get the temperature • Get temperature thresholds	If Impl, Mandatory If Impl, Recom

Cooling	• Get fan speeds • Get fan redundancies	If Impl, Mandatory If Impl, Recom
Log	• Get log entry • Clear the log	Mandatory Recommended
Management Controller	• Get version of firmware for mgmt controller • Get status of mgmt controller • Get network information for mgmt controller • Reset the mgmt controller	Mandatory Mandatory Mandatory Mandatory

The following table lists the capabilities provided by the server profile's direct requirements.

Table 2 - Server-specific Capabilities

Use Case	Manageable Capabilities	Rqmt	
Systems	• Get list of systems • Get information on a system	Mandatory Mandatory	Section 5.1 Section 5.2
System inventory	• Get the inventory data • Get the Asset Tag • Set the Asset Tag	Mandatory Mandatory Mandatory	Section 5.3 Section 5.3 Section 5.4
Location	• Get the location LED • Set the location LED	Recommended Recommended	Section 5.5 Section 5.6
System	• Get the type of system • Get the power state of the system • Get status of computer system • Reset the system	Mandatory Mandatory Mandatory Mandatory	Section 5.7 Section 5.8 Section 5.9 Section 5.10
System firmware	• Get version of system firmware	Mandatory	Section 5.11
Processor	• Get summary of processor information	Mandatory	Section 5.12
Memory	• Get summary of memory information	Mandatory	Section 5.13
Ethernet	• Get list of Ethernet interfaces • Get an Ethernet interface • Get status of Ethernet interface • Get IPv4 address • Set IPv4 address	Mandatory Mandatory Mandatory Mandatory Mandatory	Section 5.14 Section 5.15 Section 5.16 Section 5.17 Section 5.18
Boot Information	• Get boot information • Set boot device	Mandatory Mandatory	Section 5.19 Section 5.20
System Log	• Get system log • Get system log entries • Get a system log entry • Clear the log	Mandatory Mandatory Mandatory Mandatory	Section 5.21 Section 5.22 Section 5.23 Section 5.24

5. Use Cases

This section describes how each capability is accomplished by interacting with the Redfish Service.

5.1. Get the list of systems

The systems managed by the Redfish Service is obtained from the Systems resource.

```
GET /redfish/v1/Systems
```

The response message contains the following fragment.

```
{
      "@odata.id": "/redfish/v1/Systems",
      "Name": "Computer System Collection",
      "Members@odata.count": 4,
      "Members": [
          { "@odata.id": "/redfish/v1/Systems/System-3" },
          { "@odata.id": "/redfish/v1/Systems/System-4" },
          { "@odata.id": "/redfish/v1/Systems/System-2" },
          { "@odata.id": "/redfish/v1/Systems/System-1" }
     }
      "@odata.type": "#ComputerSystemCollection.ComputerSystemCollection",
}
```

5.2. Get a system

A system managed by the Redfish Service is obtained from the System resource.

```
GET /redfish/v1/Systems/{id}
```

The response message contains the following fragment. The fragment only contains the required properties as specified in the Server profile. The other properties can be found in the schema files and new properties may be added in a Redfish schema update, which occurs 3-4 times a year. A snapshot is provided in the appendix.

```
{
      "@odata.id": "/redfish/v1/Systems/1",
      "SystemType": "Physical",
      "AssetTag": "free form asset tag",
      "Manufacturer": "Manufacturer Name",
      "Model": "Model Name",
      "SKU": "",
      "SerialNumber": "2M220100SL",
      "PartNumber": "",
      "UUID": "00000000-0000-0000-0000-000000000000",
      "Status": {
            "State": "Enabled",
            "Health": "OK"
      },
      "IndicatorLED": "Off",
      "PowerState": "On",
      "Boot": {
            "BootSourceOverrideEnabled": "Once",
            "BootSourceOverrideMode": "UEFI",
            "BootSourceOverrideTarget": "Pxe",
            "UefiTargetBootSourceOverride": "",
      },
      "BiosVersion": "P79 v1.00 (09/20/2013)",
      "ProcessorSummary": {
            "Count": 8,
            "Model": "Multi-Core Intel(R) Xeon(R) processor 7xxx Series",
      },
      "MemorySummary": {
            "TotalSystemMemoryGiB": 16
```

```
    },
    "LogServices": { … },
    "Links": {
        "Chassis":          [ { "@odata.id": "/redfish/v1/Chassis/1" } ],
        "ManagedBy":        [ { "@odata.id": "/redfish/v1/Managers/1" } ]
    },
    "Actions": {
        "#ComputerSystem.Reset": {
            "target": "/redfish/v1/Systems/1/Actions/ComputerSystem.Reset",
            "@Redfish.ActionInfo": "/redfish/v1/Systems/1/ResetActionInfo"
        }
    }
}
```

Redfish models a node as a physical chassis and the logical computer system. The relationship of the system to the chassis which hosts the system is specified in the Links.Chassis property. The relationship of the system to the manager which manages the system (i.e., management controller) is specified in the Links.ManagedBy property.

5.3. Get the inventory information

The inventory information for the system is obtained by retrieving the System resource.

```
GET /redfish/v1/Systems/{id}
```

The response message contains the following fragment. The AssetTag properties is a client writeable property.

```
{
    "@odata.id": "/redfish/v1/System/1",
     "AssetTag": "free form asset tag",
     "Manufacturer": "Manufacturer Name",
     "Model": "Model Name",
     "SKU": "",
     "SerialNumber": "2M220100SL",
     "PartNumber": "",
     "UUID": "00000000-0000-0000-0000-000000000000"
}
```

5.4. Set the asset tag

The asset tag of system is set by modifying the System resource.

```
PATCH /redfish/v1/System/(id}
```

The PATCH request includes the following message.

```
{
    "AssetTag": “989846353530048”
}
```

On successful completion, the response message contains the System resource.

5.5. Get the location LED

The state of the location LED is obtained by retrieving the System resource.

```
GET /redfish/v1/System/1
```

The response message contains one of the following two fragments.

```
{
      "IndicatorLED": "Lit"
}
```

Or

```
{
      "LocationIndicatorActive": True
}
```

5.6. Set the location LED

The state of the location LED is set by setting the IndicatorLED or the LocationIndicatorActive property in the Chassis resource.

```
PATCH /redfish/v1/Chassis/Ch-1
```

The PATCH request includes one of the following two messages, with corresponds to the property returned in the GET request.

```
{
      "IndicatorLED": "Lit"
}
```

Or

```
{
      "LocationIndicatorActive": True
}
```

5.7. Get the type of system

The type of the system is obtained by retrieving the System resource.

```
GET /redfish/v1/System/{id}
```

The response message contains the following fragment.

```
{
      "@odata.id": "/redfish/v1/System/1",
      "SystemType": "Physical"
}
```

The possible values for the SystemType property are: "Physical", "Virtual", "OS", "PhysicallyPartitioned", "VirtuallyPartitioned" and "Composed".

5.8. Get power state of system

The power state of the computer system is obtained by retrieving the System resource.

```
GET /redfish/v1/System/CS-1
```

The response message contains the following fragment.

```
{
      "@odata.id": "/redfish/v1/System/CS-1",
      "PowerState": "On"
}
```

5.9. Get status of the computer system

The status and health the computer system aspect is obtained by retrieving the System resource.

```
GET /redfish/v1/System/CS-1
```

The response message contains the following fragment.

```
{
	"@odata.id": "/redfish/v1/System/1",
	"Status": {
		"State": "Enabled",
		"Health": "OK"
	}
}
```

5.10. Reset the computer system

The system is reset by performing a POST action.

```
POST /redfish/v1/System/{id}/Actions/Manager.Reset
```

The POST request includes the following message. The ResetType property contains type of reset to perform.

```
{
	"ResetType": "ForceRestart"
}
```

The possible values for the ResetType property are "ForceOff", "On", and "ForceRestart".

No response message is provided.

5.11. Get the revision of the system firmware

The version of system firmware (AKA BIOS) on a computer system is obtained by retrieving the System resource.

```
GET /redfish/v1/Systems/{id}
```

The response contains the following fragment. The information of interest is the value of the BiosVersion property.

```
{
	"@odata.id": "/redfish/v1/System/CS_1",
	"BiosVersion": "P79 v1.00 (09/20/2013)"
}
```

5.12. Get a summary of the processor information

The status of the processors is obtained by retrieving the System resource.

```
GET /redfish/v1/System/CS-1
```

The response message contains the following fragment. Within the ProcessorSummary property, the Status property contains the State and Health of the processors.

```
{
	"@odata.id": "/redfish/v1/System/CS-1",
	"ProcessorSummary": {
		"Count": 8,
		"LogicalProcessorCount": 256,
		"Model": "Multi-Core Intel(R) Xeon(R) processor 7xxx Series",
```

```
            "Status": {
                  "State": "Enabled",
                  "Health": "OK",
                  "HealthRollup": "OK"
            },
}
```

More details and health of the individual processors can be found by inspecting the individual processor resources in the Processors collection resource.

5.13. Get summary of the memory information

The status of the memory is obtained by retrieving the System resource.

```
GET /redfish/v1/System/CS-1
```

The response message contains the following fragment. Within the MemorySummary property, the Status property contains the State and Health of the memory.

```
{
      "@odata.id": "/redfish/v1/System/CS-1",
      "MemorySummary": {
          "TotalSystemMemoryGiB": 16,
            "MemoryMirroring": "System",
            "Status": {
                  "State": "Enabled",
                  "Health": "OK",
                  "HealthRollup": "OK"
            }
      }
}
```

More details and health of the individual processors can be found by inspecting the individual processor resources in the Memory collection resource.

5.14. Get list of Ethernet interfaces

The list of Ethernet interfaces is obtained by retrieving the EthernetInterfaces resource on the System of interest.

```
GET /redfish/v1/System/{id}/EthernetInterfaces
```

The response message contains the following fragment.

```
{
      "@odata.id": "/redfish/v1/Systems/1/EthernetInterfaces",
      "Members@odata.count": 1,
      "Members": [
            { "@odata.id": "/redfish/v1/Systems/1/EthernetInterfaces/1" }
      ]
}
```

5.15. Get an Ethernet Interface

The Ethernet interface is obtained by retrieving the EthernetInterface resource on the System of interest.

```
GET /redfish/v1/Systems/{id}/EthernetInterfaces/{id}
```

The response message contains the following fragment. The fragment only contains the properties specified in the Server profile. Since the EthernetInterface resource may get new properties during in Redfish schema update release, a snapshot is provided in the appendix.

```
{
      "@odata.id": "/redfish/v1/Systems/1/EthernetInterfaces/1",
      "Status": {
            "State": "Enabled",
            "Health": "OK"
      },
      "InterfaceEnabled": true,
      "MACAddress": "1E:C3:DE:6F:1E:24",
       "LinkStatus": "Linkup",
      "SpeedMbps": 100,
      "HostName": "MyHostName",
      "FQDN": "MyHostName.MyDomainName.com",
      "NameServers": [ …    ],
      "IPv4Addresses": [
            {
                  "Address": "192.168.0.10",
                  "SubnetMask": "255.255.252.0",
                  "AddressOrigin": "Static",
                  "Gateway": "192.168.0.1"
            }
     ],
      "@Redfish.Settings": {
            "@odata.type": "#Settings.v1_0_0.Settings",
            "SettingsObject": {
                  "@odata.id": "/redfish/v1/Managers/1/EthernetInterfaces/1/SD"
            }
      }
}
```

5.16. Get status of Ethernet interface

The status of an Ethernet interface is obtained by retrieving the EthernetInterface resource.

```
GET /redfish/v1/System/{id}/EthernetInterfaces/{id}
```

The response message contains the following fragment.

```
{
      "@odata.id": "/redfish/v1/Systems/1/EthernetInterfaces/1",
      "Status": {
            "State": "Enabled",
            "Health": "OK"
      },
      "InterfaceEnabled": true,
      "LinkStatus": "Linkup"
}
```

5.17. Get IPv4 address

The IPv4 address are obtained by retrieving the EthernetInterface resource.

```
GET /redfish/v1/System/{id}/EthernetInterfaces/{id}
```

The response message contains the following fragment.

```
{
      "@odata.id": "/redfish/v1/Systems/1/EthernetInterfaces/1",
      "IPv4Addresses": [
            {
                  "Address": "192.168.0.10",
                  "SubnetMask": "255.255.252.0",
                  "AddressOrigin": "Static",
                  "Gateway": "192.168.0.1"
            }
     ]
}
```

5.18. Set IPv4 Address

The IPv4 address is set by modifying the Settings resource which is subordinate to the EthernetInterface resource. In the fragment below, the @Redfish.Settings property indicates the location of the settings resource as being subordinate. A HTTP POST to the path will apply the values of the settings resource.

```
{
      "@odata.id": "/redfish/v1/Systems/1/EthernetInterfaces/1",
      "@Redfish.Settings": {
            "SettingsObject": {
                  "@odata.id": "/redfish/v1/Managers/1/EthernetInterfaces/1/SD"
            }
      }
}
```

The Settings resource, itself, represents the future intended state of a resource. The Settings resource contains a subset of properties of the EthernetInterface resource. The Settings resource may contain a @Redfish.SettingsApplyTime to indicate when the settings shall be applied.

```
{
     "@Redfish.SettingsApplyTime": ".."
     "IPv4Addresses": [
            {
                  "Address": "192.170.0.15",
                  "SubnetMask": "255.255.252.0",
                  "AddressOrigin": "Static",
                  "Gateway": "192.170.0.1"
            }
     ]
}
```

Once the properties within the settings resource have the desired values, a POST to the settings resource applies the settings.

```
POST /redfish/v1/Systems/{id}/EthernetInterfaces/{id}/SD
```

5.19. Get boot Information

The boot information is obtained by retrieving the System resource.

```
GET /redfish/v1/Systems/{id}
```

The response message contains the following fragment.

```
{
    "@odata.id": "/redfish/v1/Systems/1",
    "Boot": {
        "AliasBootOrder": [ "Hdd", "CD" ],
        "BootSourceOverrideEnabled": "Once",
        "BootSourceOverrideMode": "UEFI",
        "BootSourceOverrideTarget": "Usb",
        "UefiTargetBootSourceOverride": ""
    }
}
```

5.20. Set the boot device order

The boot order is set by modifying the System resource.

```
PATCH /redfish/v1/Systems/{id}
```

The PATCH request includes the following message.

5.20.1. Default boot order

This message sets the boot order to the hard disk (Hdd), then the USB fob (Usb).

```
{
    "Boot": {
        "AliasBootOrder": [ "Hdd", "Usb" ]
    }
}
```

5.20.2. Boot from a UEFI device.

This message sets the boot order to boot from the device specified as the UEFI target.

```
{
    "Boot": {
        "AliasBootOrder": [ "UefiTarget" ]
        "UefiTargetBootSourceOverride": "<UEFI device path>"
    }
}
```

5.20.3. Boot from a UEFI HTTP network location.

This message sets the boot order to boot from an UEFI HTTP network location.

```
{
    "Boot": {
        "AliasBootOrder": [ "UefiHttp" ],
        "HttpBootUri": "<URI bootpath"
    }
}
```

5.20.4. Boot override

The following message instructs the system to boot once from the USB in UEFI BIOS boot mode.

```
{
      "Boot": {
            "BootSourceOverrideEnabled": "Once",
            "BootSourceOverrideMode": "Legacy",
            "BootSourceOverrideTarget": "Usb",
      }
}
```

5.21. Get system log

The System's log is obtained retrieving the Log resource which represent the system log.

```
GET /redfish/v1/Systems/CS-1/LogServices/Log
```

The response message contains the following fragment. The Entries resource contains the entries of the log.

```
{
      "@odata.id": "/redfish/v1/Systems/CS-1/LogServices/Log",
      "Name": "System Log",
     . . .
      "Entries": {
            "@odata.id": "/redfish/v1/Systems/CS-1/LogServices/Log/Entries"
      }
}
```

5.22. Get system log entries

The entries of a system log are obtained by retrieving each entry of the log.

```
GET /redfish/v1/Systems/CS-1/LogService/Log/Entries
```

The following fragment is

```
{
      "@odata.id": "/redfish/v1/Systems/1/LogServices/Log1/Entries",
      "Members@odata.count": 2,
      "Members": [
            {
                  "@odata.id": "/redfish/v1/Systems/1/LogServices/Log1/Entries/1",
                  "EntryType": "SEL",
                  "Severity": "Critical",
                  "Created": "2012-03-07T14:44:00Z",
                  "EntryCode": "Upper Critical - going high",
                  "SensorType": "Temperature",
                  "SensorNumber": 1,
                  "Message": "Temperature threshold exceeded",
                  "MessageId": "0x592A28",
                  "Links": {
                        "OriginOfCondition": { "@odata.id": "/redfish/v1/Chassis/1/Thermal" }
                  }
            },
            {
                  "@odata.id": "/redfish/v1/Systems/1/LogServices/Log1/Entries/2",
                  "EntryType": "SEL",
                  "Severity": "Critical",
```

```
                "Created": "2012-03-07T14:45:00Z",
                "EntryCode": "Upper Critical - going high",
                "SensorType": "Temperature",
                "SensorNumber": 2,
                "Message": "Temperature threshold exceeded",
                "MessageId": "0x592E28",
                "Links": {
                    "OriginOfCondition": { "@odata.id": "/redfish/v1/Chassis/1/Thermal" }
                }
            }
    ]
}
```

5.23. Get a system log entry

A system log entry is obtained by retrieving a specific entry of the log.

```
GET /redfish/v1/Systems/CS-1/LogService/Log/Entries/1
```

The following fragment is

```
{
    "@odata.id": "/redfish/v1/Systems/1/LogServices/Log1/Entries/1",
    "EntryType": "SEL",
    "Severity": "Critical",
    "Created": "2012-03-07T14:44:00Z",
    "EntryCode": "Upper Critical - going high",
    "SensorType": "Temperature",
    "SensorNumber": 1,
    "Message": "Temperature threshold exceeded",
    "MessageId": "0x592A28",
    "Links": {
        "OriginOfCondition": { "@odata.id": "/redfish/v1/Chassis/1/Thermal" }
    }
}
```

5.24. Clear the system log

The system log is cleared by POST'ing to the ClearLog action for

```
POST /redfish/v1/Systems/CS-1/LogService/Log/Actions/LogService.ClearLog
```

6. References

[1] Usage Guide and Requirements for the OCP Baseline Hardware Management Profile v1.0.1

[2] “Redfish API Specification”
https://www.dmtf.org/dsp/DSP0266

7. Revision

Revision	Date	Description
1.0.0	11/16/2020	Initial draft
1.0.0 draft 2	1/18/2021	Use same format as Baseline Usage Guide.