



OPEN

Compute Project

Usage Guide and Requirements for the
OCP Baseline Hardware Management
API v1.0.1

Participants: John Leung

June 2021

Table of Contents

1.	License	3
2.	Scope	4
3.	Requirements	4
4.	Capabilities	4
5.	Use Cases	4
5.1.	Get accounts	4
5.2.	Get sessions	5
5.3.	Get the inventory information	6
5.4.	Set the asset tag	6
5.5.	Get the location LED	6
5.6.	Set the location LED	7
5.7.	Get status of the chassis	7
5.8.	Get the power state	7
5.9.	Get the power consumption	7
5.10.	Get the power limit	8
5.11.	Get the temperature	8
5.12.	Get the temperature	8
5.13.	Obtain fan readings	9
5.14.	Get fan redundancies	9
5.15.	Retrieve the system log	10
5.16.	Clear the system log	11
5.17.	Obtain the revision of firmware for the management controller	11
5.18.	Get status of management controller	11
5.19.	Get network information for management controller	11
5.20.	Reset the management controller	13
6.	References	13
7.	Revision	13

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 Baseline Hardware Management API v1.0.1, the profile is located at –

https://github.com/opencomputeproject/OCP-Profiles/blob/master/OCPBaselineHardwareManagement.v1_0_1.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>.

4. Capabilities

The following use cases and associated resources have been identified to allow BMC interface to provide baseline management capabilities.

Use Case	Manageable Capabilities	Requirement	
Account Management	<ul style="list-style-type: none">• Get accounts	Mandatory	Section 5.1
Session Management	<ul style="list-style-type: none">• Get sessions	Mandatory	Section 5.2
Hardware inventory	<ul style="list-style-type: none">• Get the FRU information• Get and Set the Asset Tag	Mandatory Recommended	Section 5.3 Section 5.4
Hardware location	<ul style="list-style-type: none">• Get the location LED• Set the location LED	Recommended Recommended	Section 5.5 Section 5.6
Status	<ul style="list-style-type: none">• Get status of chassis	Mandatory	Section 5.7
Power	<ul style="list-style-type: none">• Get power state• Get power usage• Get power limit	Mandatory Recommended Recommended	Section 5.8 Section 5.9 Section 5.10
Temperature	<ul style="list-style-type: none">• Get the temperature• Get temperature thresholds	If Impl, Mandatory If Impl, Recom	Section 5.11 Section 5.12
Cooling	<ul style="list-style-type: none">• Get fan speeds• Get fan redundancies	If Impl, Mandatory If Impl, Recom	Section 5.13 Section 5.14
Log	<ul style="list-style-type: none">• Get log entry• Clear the log	Mandatory Recommended	Section 5.15 Section 5.16
Management Controller	<ul style="list-style-type: none">• 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	Section 5.17 Section 5.18 Section 5.19 Section 5.20

5. Use Cases

This section describes how each capability is accomplished by interacting via the Redfish Interface.

5.1. Get accounts

The accounts on the management controller is obtained from the AccountService resource.

```
GET /redfish/v1/AccountService
```

The response message contains the following fragment.

```
{
  "@odata.id": "/redfish/v1/AccountService",
  ...
  "Accounts": { "@odata.id": "/redfish/v1/AccountService/Accounts" },
  "Roles": { "@odata.id": "/redfish/v1/AccountService/Roles" }
}
```

The Roles property specifies the path to the Roles collection resource. The Redfish specification specifies the Admin, Operator and ReadOnly roles be member resource.

The Account resource represents each account on the management controller and the role associated to the account.

```
Get /redfish/v1/AccountService/Accounts/1
```

The following is a fragment of an Account resource.

```
{
  "@odata.id": "/redfish/v1/AccountService/Accounts/1",
  "Name": "User Account",
  "Enabled": true,
  "Password": null,
  "PasswordChangeRequired": false,
  "UserName": "Administrator",
  "RoleId": "Administrator",
  "Locked": false,
  "Links": {
    "Role": { "@odata.id": "/redfish/v1/AccountService/Roles/Administrator" }
  }
}
```

5.2. Get sessions

The sessions on the management controller is obtained from the SessionService resource.

```
GET /redfish/v1/SessionService
```

The response message contains the following fragment.

```
{
  "@odata.id": "/redfish/v1/SessionService",
  "ServiceEnabled": true,
  "SessionTimeout": 30,
  "Sessions": { "@odata.id": "/redfish/v1/SessionService/Sessions" }
}
```

The Sessions property specifies the path to the Sessions collection resource. The Redfish service creates Session resources for individual session that are established. The following is a fragment of a Session resource.

```
{
  "@odata.id": "/redfish/v1/SessionService/Sessions/1234567890ABCDEF",
  "Id": "1234567890ABCDEF",
  "Name": "User Session",
}
```

```
"UserName": "Administrator"
}
```

5.3. Get the inventory information

The hardware inventory for the rack is obtained from the Chassis resource representing each node's hardware.

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

The response message contains the following fragment. The response contains the hardware inventory properties for manufacturer, model, SKU, serial number, and part number. The AssetTag properties is a client writeable property.

```
{
  "@odata.id": "/redfish/v1/Chassis/Ch-1",
  "Id": "Node1",
  ...
  "ChassisType": "Node",
  "Manufacturer": "Contoso"
  "Model": "RackScale_Rack",
  "SKU": "...",
  "SerialNumber": "...",
  "PartNumber": "...",
  "AssetTag": null,
  "IndicatorLED": "Off"
}
```

5.4. Set the asset tag

The hardware inventory for the rack is obtained from the Chassis resource representing each node's hardware.

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

The PATCH request includes the following message.

```
{
  "AssetTag": "989846353530048"
}
```

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

5.5. Get the location LED

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

```
GET /redfish/v1/Chassis/Ch-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 status of the chassis

Redfish models a node as its physical chassis and the logical computer system. The relationship between the two resource and specified by references.

The status and health the chassis aspect is obtained by retrieving the Chassis resource.

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

The following message is the response. The Status property contains the state and health of the chassis.

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

5.8. Get the power state

The power state is obtained from the Chassis resource.

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

The response message contains the following fragment. The response contains the PowerState property.

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

5.9. Get the power consumption

The power consumption is obtained from the Chassis resource.

```
GET /redfish/v1/Chassis/Ch-1/Power
```

The response message contains the following fragment. The PowerControl property contains the PowerConsumedWatts PowerCapacityWatts properties.

```
{
  "@odata.id": "/redfish/v1/Chassis/Ch-1/Power",
  "PowerControl": {
    "PowerConsumedWatts": "215",
    "PowerCapacityWatts": "230"
  }
}
```

5.10. Get the power limit

The power limit is obtained from the Chassis resource.

```
GET /redfish/v1/Chassis/Ch-1/Power
```

The response message contains the following fragment. The PowerControl property contains the LimitInWatts and LimitException properties.

```
{
  "@odata.id": "/redfish/v1/Chassis/Ch-1/Power",
  "PowerControl": {
    "PowerLimit": {
      "LimitInWatts": "220",
      "LimitException": "230"
    }
  }
}
```

5.11. Get the temperature

The temperature is obtained from the Thermal resource which is subordinate to Chassis resource.

```
GET /redfish/v1/Chassis/Chassis_1/Thermal
```

The response message contains the following fragment. One of the elements in the Temperatures array property. The ReadingCelsius property contains the value of temperature is required. The threshold properties are optional.

```
{
  "@odata.id": "/redfish/v1/Chassis/Ch-1/Thermal",
  "Temperatures": [
    {
      "ReadingCelsius": 21
    }
  ]
}
```

5.12. Get the temperature thresholds

The temperature thresholds are obtained from the Thermal resource which is subordinate to Chassis resource.

```
GET /redfish/v1/Chassis/Chassis_1/Thermal
```

The response message contains the following fragment. One of the elements in the Temperatures array property. The ReadingCelsius property contains the value of temperature is required. The threshold properties are optional.

```
{
  "@odata.id": "/redfish/v1/Chassis/Ch-1/Thermal",
```



```

    "Temperatures": [
      {
        "UpperThresholdFatal": "45",
        "UpperThresholdCritical": "40",
        "UpperThresholdNonCritical": "35"
      }
    ]
  }
}

```

5.13. Obtain fan readings

The fan speeds are obtained from the

of a node is obtained from the Thermal resource subordinate to Chassis resource which represents node's chassis.

```
GET /redfish/v1/Chassis/Ch-1/Thermal
```

The response contains the following fragment. Within the Fans array property, each array member has a Reading and ReadingUnits property.

```

{
  "@odata.id": "/redfish/v1/Chassis/Ch-1/Thermal",
  "Fans": [
    {
      "Name": ""
      "Reading": 300
      "ReadingUnits": "RPM"
    }
  ]
}

```

5.14. Get fan redundancies

Fans which are configured in a redundancy set should be available via the resource model.

The fans redundancy structure is obtained from Thermal resource.

```
GET /redfish/v1/Chassis/Ch-1/Thermal
```

The response message contains the following fragment. The Redundancy array property contains as list of the redundancies. The redundancy contains a RedundancySet property which contains the members of the redundancy set.

```

{
  "@odata.id": "/redfish/v1/Chassis/Ch-1/Thermal",
  "Fans": [
    {
      "@odata.id": "/redfish/v1/Chassis/Ch-1/Thermal#/Fans/0",
      "MemberId": "0",
      "Reading": 300
      "ReadingUnits": "RPM"
    },
    {
      "@odata.id": "/redfish/v1/Chassis/Ch-1/Thermal#/Fans/1",
      "MemberId": "1",
      "Reading": 300
      "ReadingUnits": "RPM"
    }
  ]
}

```

```

    }
  ],
  "Redundancy": [
    {
      "@odata.id": "/redfish/v1/Chassis/Ch-1/Thermal#/Redundancy/0",
      "MemberId": "0",
      "RedundancySet": [
        { "@odata.id": "/redfish/v1/Chassis/Ch-1/Thermal#/Fans/0" },
        { "@odata.id": "/redfish/v1/Chassis/Ch-1/Thermal#/Fans/1" }
      ],
      "Mode": "N+m",
      "Status": {
        "State": "Disabled",
        "Health": "OK"
      },
      "MinNumNeeded": 1,
      "MaxNumSupported": 2
    }
  ]
}

```

5.15. Retrieve the system log

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

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

The response message contains the following fragment. The value of the Entries property is the collection resource for the entries in 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"
  }
}

```

The client can get each 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",
  "Message": "Temperature threshold exceeded",
}

```

5.16. Clear the system log

The System's log is retrieved is obtained by retrieving the Log resource which represent the node's log.

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

5.17. Obtain the revision of firmware for the management controller

The version of firmware for the management controller is obtained by retrieving the Manager resource which represents the management controller of interest.

```
GET /redfish/v1/Managers/BMC_1
```

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

```
{
  "@odata.id": "/redfish/v1/Managers/BMC",
  "FirmwareVersion": "1.00"
}
```

5.18. Get status of management controller

The status of the management controller is obtained by retrieving the Manager resource.

```
GET /redfish/v1/Managers/BMC
```

The response message contains the following fragment. The Status property contains the State and Health properties of the manager.

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

5.19. Get network information for management controller

The network information for the management controller is obtained by retrieving the EthernetInterface resource.

```
GET /redfish/v1/Managers/BMC/EthernetInterface
```

The response message contains the following fragment.

```
{
  "@odata.id": "/redfish/v1/Managers/BMC/EthernetInterface",
  "Status": {
    "Health": "Enabled",
    "State": "OK"
  },
  "MacAddress": "1E:C3:DE:6F:1E:24",
  "SpeedMbps": 100,
  "InterfaceEnabled": true,
  "LinkStatus": "LinkUp",
  "HostName": "MyHostName",
}
```

```

"FQDN": "MyHostName.MyDomainName.com",
"IPv4Addresses": [
  {
    "Address": "192.168.0.10",
    "SubnetMask": "255.255.252.0",
    "AddressOrigin": "DHCP",
    "Gateway": "192.168.0.1"
  }
],
"NameServers": [
  "192.168.200.10",
  "192.168.150.1",
  "fc00:1234:100::2500"
]
}

```

The response message may contain properties from the following fragment.

```

{
  "@odata.id": "/redfish/v1/Managers/BMC/EthernetInterface",
  "StaticNameServers": [
    "192.168.150.1",
    "fc00:1234:200:2500"
  ],
  "DHCPv4": {
    "DHCPEnabled": true,
    "UseDNSServers": true,
    "UseGateway": true,
    "UseNTPServers": false,
    "UseStaticRoutes": true,
    "UseDomainName": true,
    "FallbackAddress": "Static"
  },
  "IPv4StaticAddresses": [
    {
      "Address": "192.168.88.130",
      "SubnetMask": "255.255.0.0",
      "Gateway": "192.168.0.1"
    }
  ],
  "DHCPv6": {
    "OperatingMode": "Stateful",
    "UseDNSServers": true,
    "UseDomainName": false,
    "UseNTPServers": false,
    "UseRapidCommit": false
  },
  "IPv6Addresses": [
    {
      "Address": "2001:1:3:5::100",
      "PrefixLength": 64,
      "AddressOrigin": "DHCPv6",
      "AddressState": "Preferred"
    }
  ]
}

```

```

],
"IPv6AddressPolicyTable": [
  {
    "Prefix": "::1/128",
    "Precedence": 50,
    "Label": 0
  }
],
"IPv6StaticAddresses": [
  {
    "Address": "fc00:1234::a:b:c:d",
    "PrefixLength": 64
  }
],
"IPv6StaticDefaultGateways": [
  {
    "Address": "fe80::fe15:b4ff:fe97:90cd",
    "PrefixLength": 64
  }
]
}

```

5.20. Reset the management controller

The management controller is reset by performing a POST action.

```
POST /redfish/v1/Manager/BMC/Actions/Manager.Reset
```

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

```

{
  "ResetType": "ForceRestart"
}

```

6. References

[1] "Redfish API Specification"
<https://www.dmtf.org/dsp/DSP0266>

7. Revision

Revision	Date	Description
1.0.0	6/15/2021	Final draft - contribution