# Intel® Rack Scale Design Pod Manager

## API Specification

### Software Version 2.1.3

*May 2017*

*Revision 002*

Document Number: 335455-002

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and noninfringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

This document contains information on products, services, and/or processes in development. All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest forecast, schedule, specifications, and roadmaps.

The products and services described may contain defects or errors known as errata which may cause deviations from published specifications. Current characterized errata are available on request.

Copies of documents that have an order number and are referenced in this document may be obtained by calling 1-800-548-4725 or by visiting *http://www.intel.com/design/literature.htm*.

Intel and the Intel logo are trademarks of Intel Corporation in the United States and other countries.

*Other names and brands may be claimed as the property of others

Copyright © 2017 Intel Corporation. All rights reserved.

# Contents


**1 Introduction ..... 7**
1.1 Scope ..... 7
1.2 Intended audience ..... 7
1.3 Terminology ..... 7
1.4 References ..... 7
**2 PODM API ..... 9**
2.1 PODM API structure and relations ..... 9
2.1.1 PODM API physical resource hierarchy ..... 9
**3 PODM REST API Error Codes ..... 12**
3.1 API error response ..... 12
3.1.1 Example error JSON object ..... 12
3.2 API error codes ..... 13
3.2.1 General error codes ..... 13
3.2.2 PATCH method error codes ..... 13
3.2.3 POST method error codes ..... 14
**4 PODM REST API Definition ..... 15**
4.1 Odata support ..... 15
4.2 Protocol version ..... 15
4.2.1 Operations ..... 15
4.3 Odata service document ..... 15
4.3.1 Operations ..... 16
4.4 Intel® RSD POD manager service root ..... 17
4.4.1 Operations ..... 17
4.5 Chassis collection ..... 18
4.5.1 Operations ..... 18
4.6 Chassis ..... 19
4.6.1 Operations ..... 21
4.7 PowerZone collection ..... 23
4.7.1 Operations ..... 23
4.8 PowerZone ..... 24
4.8.1 Operations ..... 26
4.9 ThermalZone collection ..... 27
4.9.1 Operations ..... 27
4.10 ThermalZone ..... 28
4.10.1 Operations ..... 30
4.11 Storage service collection ..... 31
4.11.1 Operations ..... 32
4.12 Composed node collection ..... 32
4.12.1 Operation ..... 36
4.13 Composed node ..... 40
4.13.1 Operations ..... 43
4.14 PSME and Storage Services resources ..... 48
4.15 Simple Storage collection ..... 49
4.15.1 Operations ..... 49
4.16 Simple storage ..... 50
4.16.1 Operations ..... 51

4.17 Power ... 52
4.17.1 Operations ... 52
4.18 Thermal ... 55
4.18.1 Operations ... 55
**5 Common Property Description ... 58**
5.1 Status ... 58
5.2 Status -> State ... 58
5.3 Status -> Health ... 58
5.4 ComputerSystem.Reset ... 58
5.5 BootSourceOverrideTarget/Supported ... 58
**6 Appendix ... 60**
6.1 Creating new Composed Node - explanation ... 60
6.1.1 Creating Composed Node using JSON template ... 60
6.1.2 Specifying requirements for a Composed Node ... 60
6.1.3 General assumptions for allocation ... 60
6.1.4 Specifying Processor requirements ... 60
6.1.5 Specifying Memory requirements ... 62
6.1.6 Specifying Remote Drive requirements ... 63
6.1.7 Specifying Local Drive requirements ... 64
6.1.8 Specifying Ethernet interface requirements ... 66
6.1.9 Allocation algorithm ... 67

## Figures

Figure 1 PODM REST API hierarchy ... 9
Figure 2 ComposedNodeState changes during assembly process ... 43

## Tables

Table 1 Terminology ... 7
Table 2 Reference documents ... 7
Table 3 Resources and URI ... 9
Table 4 API error response attributes ... 12
Table 5 HTTP error status codes ... 13
Table 6 Chassis collection attributes ... 18
Table 7 Chassis attributes ... 19
Table 8 PowerZone collection attributes ... 23
Table 9 PowerZone attributes ... 24
Table 10 PowerSupply attributes ... 25
Table 11 ThermalZone collection attributes ... 27
Table 12 ThermalZone attributes ... 28
Table 13 Fan attributes ... 29
Table 14 Temperature attributes ... 29
Table 15 Storage service collection attributes ... 31
Table 16 Composed node collection attributes ... 32
Table 17 Composed node Allocation action attributes ... 33
Table 18 Remote master target properties ... 36
Table 19 Composed node attributes ... 40
Table 20 Boot Override update properties ... 46

Table 21 Simple storage collection attributes ..... 49
Table 22 Simple storage attributes ..... 50

# Revision History

| Revision | Description | Date |
|---|---|---|
| 002 | Added FabricSwitch option to local drive section of compose node allocation request. | May 12, 2017 |
| 001 | Initial release. | February 9, 2017 |

§

# 1 Introduction

## 1.1 Scope

This document contains information about the Intel® Rack Scale Design (Intel® RSD) Pod Manager RESTAPI, which was designed and implemented for the Intel® RSD Software v2.1 release.

The interface specified in this document are based on the Distributed Management Task Force's Redfish™ Interface Specification and schema (see dmtf.org) version 2016.3.

## 1.2 Intended audience

The intended audience for this document is designers and engineers working with the Rack Scale Design Software 2.1 release.

## 1.3 Terminology

**Table 1 Terminology**

| Term | Definition |
|---|---|
| BMC | Baseboard Management Controller |
| CIMI | Cloud Infrastructure Management Interface |
| HTTP | Hypertext Transfer Protocol |
| JSON | JavaScript Object Notation |
| NIC | Network Interface Card |
| OCCI | Open Cloud Computing Interface |
| OData | Open Data Protocol |
| OVF | Open Virtualization Format |
| POD | A physical collection of multiple racks |
| PODM | Pod Manager |
| PSME | Pooled System Management Engine |
| REST | Representational state transfer |
| URI | Uniform resource identifier |
| UUID | Universally Unique Identifier |
| XML | Extensible Markup Language |

## 1.4 References

**Table 2 Reference documents**

| Doc ID | Title | Location |
|---|---|---|
| 335451 | Intel® Rack Scale Design Generic Assets Management Interface API Specification | *Intel.com/intelrsd_resources* |
| 335452 | Intel® Rack Scale Design BIOS & BMC Technical Guide | *Intel.com/intelrsd_resources* |
| 335501 | Intel® Rack Scale Design Architecture Specification | *Intel.com/intelrsd_resources* |
| 335454 | Intel® Rack Scale Design Software Reference Kit Getting Started Guide | *Intel.com/intelrsd_resources* |
| 335455 | Intel® Rack Scale Design Pod Manager API Specification | *Intel.com/intelrsd_resources* |
| 335456 | Intel® Rack Scale Design Pod Manager Release Notes | *Intel.com/intelrsd_resources* |
| 335457 | Intel® Rack Scale Design Pod Manager User Guide | *Intel.com/intelrsd_resources* |
| 335458 | Intel® Rack Scale Design PSME REST API Specification | *Intel.com/intelrsd_resources* |
| 335459 | Intel® Rack Scale Design PSME Release Notes | *Intel.com/intelrsd_resources* |
| 335460 | Intel® Rack Scale Design PSME User Guide | *Intel.com/intelrsd_resources* |

| Doc ID | Title | Location |
|---|---|---|
| 335461 | Intel® Rack Scale Design Storage Services API Specification | *Intel.com/intelrsd_resources* |
| 335462 | Intel® Rack Scale Design Rack Management Module (RMM) API Specification | *Intel.com/intelrsd_resources* |
| 335463 | Intel® Rack Scale Design RMM Release Notes | *Intel.com/intelrsd_resources* |
| 335464 | Intel® Rack Scale Design Software RMM User Guide | *Intel.com/intelrsd_resources* |
| DSP0266 | Redfish Scalable Platform Management API Specification | *http://dmtf.org/standards/redfish* |

§

# 2 PODM API

## 2.1 PODM API structure and relations

The PODM REST API provides the REST-based interface that allows full management of the Intel® RSD POD including asset discovery, configuration, and composed node assembly.

### 2.1.1 PODM API physical resource hierarchy

**Figure 1 PODM REST API hierarchy**

**Table 3 Resources and URI**

| Resource | URI |
|---|---|
| Service Root | /redfish/v1 |
| Chassis Collection | /redfish/v1/Chassis |
| Chassis | /redfish/v1/Chassis/{chassisID} |
| Computer System Collection | /redfish/v1/Systems |
| Computer System | /redfish/v1/Systems/{systemID} |
| Processors Collection | /redfish/v1/Systems/{systemID}/Processors |
| Processor | /redfish/v1/Systems/{systemID /Processors/{processorID} |
| Memory Collection | /redfish/v1/Systems/{systemID}/Memory |
| Memory | /redfish/v1/Systems/{systemID}/Memory/{memoryID} |
| Manager Collection | /redfish/v1/Managers |

| Resource | URI |
|---|---|
| Manager | /redfish/v1/Managers/{managerID} |
| Network Protocol | /redfish/v1/Managers/{managerID}/NetworkProtocol |
| Network Interface Collection | /redfish/v1/Systems/{systemID}/EthernetInterfaces<br>/redfish/v1/Managers/{managerID}/EthernetInterfaces |
| Network Interface | /redfish/v1/Systems/{systemID}/EthernetInterfaces/{nicID}<br>/redfish/v1/Managers/{managerID}/EthernetInterfaces/{nicID} |
| Ethernet Switch Collection | /redfish/v1/EthernetSwitches |
| Ethernet Switch | /redfish/v1/EthernetSwitches/{switchID} |
| Ethernet Switch Port Collection | /redfish/v1/EthernetSwitches/{switchID}/Ports |
| ethernet Switch Port | /redfish/v1/EthernetSwitches/{switchID}/Ports/{portID} |
| VLAN Network Interface Collection | /redfish/v1/EthernetSwitches/{switchID}/Ports/{portID}/VLANs<br>/redfish/v1/Systems/{systemID}/EthernetInterfaces/{nicID}/VLANs<br>/redfish/v1/Managers/{managerID}/EthernetInterfaces/{nicID}/VLANs |
| VLAN Network Interface | /redfish/v1/EthernetSwitches/{switchID}/Ports/{portID}/VLANs/{vlanID}<br>/redfish/v1/Systems/{systemID}/EthernetInterfaces/{nicID}/VLANs/{vlanID}<br>/redfish/v1/Managers/{managerID}/EthernetInterfaces/{nicID}/VLANs/{vlanID} |
| Storage Service Collection | /redfish/v1/Services |
| Storage Service | /redfish/v1/Services/{serviceID} |
| Remote Target Collection | /redfish/v1/Services/{serviceID}/Targets |
| Remote Target | /redfish/v1/Services/{serviceID}/Targets/{targetID} |
| Logical Drive Collection | /redfish/v1/Services/{serviceID}/LogicalDrives |
| Logical Drive | /redfish/v1/Services/{serviceID}/LogicalDrives/{driveID} |
| Physical Drive Collection | /redfish/v1/Services/{serviceID}/Drives |
| Physical Drive | /redfish/v1/Services/{serviceID}/Drives/{driveID} |
| Composed Node Collection | /redfish/v1/Nodes |
| Composed Node | /redfish/v1/Nodes/{nodeID} |
| Simple Storage Collection | /redfish/v1/Systems/{system1}/SimpleStorage |
| SimpleStorage | /redfish/v1/Systems/{system1}/SimpleStorage/{storageID} |
| PowerZone Collection | /redfish/v1/Chassis/{chassisID}/PowerZones |
| PowerZone | /redfish/v1/Chassis/{chassisID}/PowerZones/{powerzoneID} |
| ThermalZone Collection | /redfish/v1/Chassis/{chassisID}/ThermalZones |
| ThermalZone | /redfish/v1/Chassis/{chassisID}/ThermalZones/{thermalzoneID} |
| Power | /redfish/v1/Chassis/{chassisID}/Power |
| Thermal | /redfish/v1/Chassis/{chassisID}/Thermal |
| Storage Subsystem Collection | /redfish/v1/Systems/{systemID}/Storage |
| Storage Subsystem | /redfish/v1/Systems/{systemID}/Storage/{storageID} |
| Drives | /redfish/v1/Chassis/{chassisID}/Drives/{driveID} |
| Fabrics collection | /redfish/v1/Fabrics |
| Fabric | /redfish/v1/Fabrics/{fabricID} |
| Fabric Switch collection | /redfish/v1/Fabrics/{fabricID}/Switches |
| Fabric Switch | /redfish/v1/Fabrics/{fabricID}/Switches/{switchID} |
| Fabric Switch Port collection | /redfish/v1/Fabrics/{fabricID}/Switches/{switchID}/Ports |
| Fabric Switch Port | /redfish/v1/Fabrics/{fabricID}/Switches/{switchID}/Ports/{portID} |
| Fabric Zone collection | /redfish/v1/Fabrics/{fabricID}/Zones |
| Fabric Zone | /redfish/v1/Fabrics/{fabricID}/Zones/{zoneID} |
| Endpoint Collection | /redfish/v1/Fabrics/{fabricID}/Endpoints |
| Endpoint | /redfish/v1/Fabrics/{fabricID}/Endpoints/{endpointID} |
| PCIeDevice | /redfish/v1/Chassis/{chassisID}/PCIeDevices/{deviceID} |

| Resource | URI |
|---|---|
| PCIe Device Function | /redfish/v1/Chassis/{chassisID}/PCIeDevices/{deviceID}/Functions/{functionID} |

§

# 3 PODM REST API Error Codes

This chapter describes all error codes that may be returned by the REST calls implemented in the PODM REST API of the Intel® RSD software v1.1 release.

## 3.1 API error response

In the case of an error, PODM REST API responds with an HTTP status code, as defined by the HTTP 1.1 specification and constrained by additional requirements defined in this specification.

HTTP response status codes alone often do not provide enough information to determine the error cause. The PODM REST API returns extended error information as a JSON object with a single property named “error”. The value of this property shall be the JSON object with the properties listed in Table 4.

**Table 4 API error response attributes**

| Attribute | Description |
|---|---|
| `MessageId` | String indicating a specific error or message (not to be confused with the HTTP status code). This code can be used to access a detailed message from a message registry. |
| `Message` | A human readable error message indicating the semantics associated with the error. This shall be the complete message, and not rely on substitution variables. |
| `MessageArgs` ..... | An optional array of strings representing the substitution parameter values for the message. This shall be included in the response if a MessageId is specified for a parameterized message. |
| `Severity` | An optional string representing the severity of the error. |
| `Resolution` | An optional string describing recommended action(s) to take to resolve the error. |
| `RelatedProperties` | An optional array of JSON Pointers defining the specific properties within a JSON payload described by the message. |

### 3.1.1 Example error JSON object

```
{
    "error": {
        "code": "Base.1.0.GeneralError",
        "message": "A general error has occurred. See ExtendedInfo for more
information.",
        "@Message.ExtendedInfo": [
            {
                "@odata.type" :
"/redfish/v1/$metadata#Message.v1_0_0.Message",
                "MessageId": "Base.1.0. MalformedJSON",
                "Message": "The request body submitted was malformed JSON and
could not be parsed by the receiving service",
                "Severity": "Error"
            }
            {
                "@odata.type" :
"/redfish/v1/$metadata#Message.v1_0_0.Message",
                "MessageId": "Base.1.0.PropertyNotWriteable",
                "RelatedProperties": [
                    "#/Name"
                ],
                "Message": "The property Name is a read only property and
cannot be assigned a value",
                "MessageArgs": [
                    "Name"
```

```
                ],
                "Severity": "Warning",
                "Resolution": "Remove the property from the request body and
resubmit the request if the operation failed"
            }
        ]
    }
}
```

## 3.2 API error codes

In general, if an error is not described in any of the following tables, it is to be mapped into an HTTP 500 Internal Error code.

### 3.2.1 General error codes

For a detailed list of error codes, refer to Redfish Scalable Platforms Management API Specification, Section 6.5.2.

**Table 5 HTTP error status codes**

| HTTP Status Code | Description |
|---|---|
| **400 Bad Request** | The request could not be processed because it contains missing or invalid information (such as a validation error on an input field, a missing required value, and so on). An extended error shall be returned in the response body. |
| **404 Not Found** | The request specified a URI of a resource that does not exist. |
| **405 Method Not Allowed** | The HTTP verb specified in the request (e.g., DELETE, GET, HEAD, POST, PUT, PATCH) is not supported for this request URI. The response shall include an Allow header which provides a list of methods that are supported by the resource identified by the Request-URI. |
| **409 Conflict** | A creation or update request cannot be applied given the state of the resource. |
| **500 Internal Server Error** | The server encountered an unexpected condition that prevented it from fulfilling the request. An extended error shall be returned in the response body. |
| **501 Not Implemented** | The server does not (currently) support the functionality required to fulfill the request. This is the appropriate response when the server does not recognize the request method and is not capable of supporting it for any resource. |
| **503 Service Unavailable** | The server is currently unable to handle the request due to temporary overloading or maintenance of the server. |

### 3.2.2 PATCH method error codes

For the PATCH method, the Rack Scale Design service shall conform to IETF RFC 5789.

The service will respond with the following error codes in the cases listed below:

- 400 Bad Request – malformed JSON in request (values not in range, unknown property, etc.)
- 405 Method Not Allowed – resource does not support the PATCH method
- 409 Conflict – update cannot be executed at this moment. User might be able to resolve the conflict and resubmit the request.
- 501 Not Implemented – resource supports PATCH method, but current implementation doesn't (e.g. underlying HW doesn't support such functionality)
- 500 Internal Server Error – all other situations where any of the above codes does not fit (e.g. underlying HW doesn't allow to execute this particular request).

### 3.2.3 POST method error codes

POST method is used to create a new resource (POST request is submitted to the resource collection to which the new resource is to belong) or to initiate an operation on the object (sending POST method to the URI of the action).

The service will respond with the following error codes in the cases listed below:

- 400 Bad Request – malformed JSON in request (values not in range, unknown property, etc.)
- 405 Method Not Allowed – resource does not support the POST method
- 409 Conflict – update cannot be executed at this moment. User might be able to resolve the conflict and resubmit the request.
- 501 Not Implemented – resource supports POST method, but current implementation does not (e.g. underlying HW doesn't support such functionality)
- 500 Internal Server Error – all other situations where any of above codes do not fit (e.g. underlying HW doesn't allow to execute this particular request). Extended error info should provide information why operation failed.

§

# 4 PODM REST API Definition

## 4.1 Odata support

Intel® RSD supports Odata v4.0 as it is defined in the Redfish specification.

All resources within this RESTful API are identified by a unique identifier property named “@odata.id”. Resource Identifiers shall be represented in JSON payloads as uri paths relative to the Redfish Schema portion of the uri. That is, they shall always start with "/redfish/". The resource identifier is the canonical URL for the resource and can be used to retrieve or edit the resource, as appropriate.

## 4.2 Protocol version

The protocol version is separate from the version of the resources or the version of the Redfish Schema supported by them.

Each version of the Redfish protocol is strongly typed. This is accomplished using the URI of the Redfish service in combination with the resource obtained at that URI, called the ServiceRoot.

The root URI for this version of the Redfish protocol shall be "/redfish/v1/".

While the major version of the protocol is represented in the URI, the major version, minor version and errata version of the protocol are represented in the Version property of the ServiceRoot resource, as defined in the Redfish Schema for that resource. The protocol version is a string of the form:

```
MajorVersion.MinorVersion.Errata
```

Where:

- *MajorVersion* = integer: something in the class changed in a backward incompatible way.
- MinorVersion = integer: a minor update. New functionality may have been added but nothing removed. Compatibility will be preserved with previous minorversions.
- Errata = integer: something in the prior version was broken and needed to be fixed.

Any resource discovered through links found by accessing the root service or any service or resource referenced using references from the root service shall conform to the same version of the protocol supported by the root service.

### 4.2.1 Operations

#### 4.2.1.1 GET

Request:

```
GET /redfish
Content-Type: applicaton/json
```

Response:

```
{
        "v1": "/redfish/v1/"
}
```

## 4.3 Odata service document

This service document provides a standard format for enumerating the resources exposed by the service, enabling generic hypermedia-driven OData clients to navigate to the resources of the service.

### 4.3.1 Operations

#### 4.3.1.1 GET

Request:

```
GET /redfish/v1/odata
Content-Type: applicaton/json
```

Response:

```
{
    "@odata.context": "/redfish/v1/$metadata",
    "value": [
        {
            "name": "Service",
            "kind": "Singleton",
            "url": "/redfish/v1/"
        },
        {
            "name": "Systems",
            "kind": "Singleton",
            "url": "/redfish/v1/Systems"
        },
        {
            "name": "Chassis",
            "kind": "Singleton",
            "url": "/redfish/v1/Chassis"
        },
        {
            "name": "Managers",
            "kind": "Singleton",
            "url": "/redfish/v1/Managers"
        },
        {
            "name": "Nodes",
            "kind": "Singleton",
            "url": "/redfish/v1/Nodes"
        },
        {
            "name": "Services",
            "kind": "Singleton",
            "url": "/redfish/v1/Services"
        },
        {
            "name": "EthernetSwitches",
            "kind": "Singleton",
            "url": "/redfish/v1/EthernetSwitches"
        },
        {
            "name": "Fabrics",
            "kind": "Singleton",
            "url": "/redfish/v1/Fabrics"
        }
    ]
}
```

## 4.4 Intel® RSD POD manager service root

Intel® RSD Pod Manager Service Root resource – entry point.

Properties details available in *ServiceRoot.xml* metadata file.

### 4.4.1 Operations

#### 4.4.1.1 GET

Request:

```
GET /redfish/v1
Content-Type: applicaton/json
```

Response:

```
{
    "@odata.context": "/redfish/v1/$metadata#ServiceRoot.ServiceRoot",
    "@odata.id": "/redfish/v1/",
    "@odata.type": "#ServiceRoot.v1_1_1.ServiceRoot",
    "Id": "RootService",
    "Name": "Root Service",
    "Description": "description-as-string",
    "RedfishVersion": "1.0.2",
    "UUID": "92384634-2938-2342-8820-489239905423",
    "Systems": {
        "@odata.id": "/redfish/v1/Systems"
    },
    "Chassis": {
        "@odata.id": "/redfish/v1/Chassis"
    },
    "Managers": {
        "@odata.id": "/redfish/v1/Managers"
    },
    "EventService": {
        "@odata.id": "/redfish/v1/EventService"
    },
    "Services": {
        "@odata.id": "/redfish/v1/Services"
    },
    "Nodes": {
        "@odata.id": "/redfish/v1/Nodes"
    },
    "EthernetSwitches": {
        "@odata.id": "/redfish/v1/EthernetSwitches"
    },
    "Fabrics": {
      "@odata.id": "/redfish/v1/Fabrics"
    },
    "Oem": {
        "Intel_RackScale": {
            "@odata.type": "#Intel.Oem.ServiceRoot",
            "ApiVersion": "2.0.0",
        }
```

```
    },
   "Links": {}
}
```

#### 4.4.1.2 PUT

Operation is not allowed on this resource.

#### 4.4.1.3 PATCH

Operation is not allowed on this resource.

#### 4.4.1.4 POST

Operation is not allowed on this resource.

#### 4.4.1.5 DELETE

Operation is not allowed on this resource.

## 4.5 Chassis collection

**Table 6 Chassis collection attributes**

| Name | Chassis | | |
|---|---|---|---|
| Type URI | /redfish/v1/Chassis | | |
| **Attribute** | **Type** | **Required** | **Description** |
| `Name` | String | Yes | Name of collection |
| `Members@odata.count` | Number | No | Collection members count |
| `Members` | Array | No | Contains the members of this collection. |

### 4.5.1 Operations

#### 4.5.1.1 GET

Request:

```
GET /redfish/v1/Chassis
Content-Type: application/json
```

Response:

```
{
    "@odata.context": "/redfish/v1/$metadata#Chassis",
    "@odata.id": "/redfish/v1/Chassis",
    "@odata.type": "#ChassisCollection.ChassisCollection",
    "Name": "Chassis Collection",
    "Members@odata.count": 6,
    "Members": [
        {
            "@odata.id": "/redfish/v1/Chassis/Pod"
        },
        {
            "@odata.id": "/redfish/v1/Chassis/Rack1"
        },
```

```
        {
            "@odata.id": "/redfish/v1/Chassis/Drawer1"
        },
        {
            "@odata.id": "/redfish/v1/Chassis/FabricModule1"
        },
        {
            "@odata.id": "/redfish/v1/Chassis/Sled1"
        },
        {
            "@odata.id": "/redfish/v1/Chassis/Blade1"
        }
    ]
}
```

#### 4.5.1.2 PUT

Operation is not allowed on this resource.

#### 4.5.1.3 PATCH

Operation is not allowed on this resource.

#### 4.5.1.4 POST

Operation is not allowed on this resource.

#### 4.5.1.5 DELETE

Operation is not allowed on this resource.

## 4.6 Chassis

This is the schema definition for the Chassis resource. It represents the properties for physical components for any system. This one object is intended to represent racks, rackmount servers, blades, standalone, modular systems, enclosures, and all other containers. The non-cpu/device centric parts of the schema are all accessed either directly or indirectly through this resource.

**Table 7 Chassis attributes**

| **Name** | Chassis | |
|---|---|---|
| **Type URI** | /redfish/v1/Chassis/{chassisId} | |
| **Attribute** | **Type** | **Description** |
| `Id` | String | Resource identifier |
| `Name` | String | Name of service root |
| `Description` | String | Provides a description of this resource and is used for commonality in the schema definitions. |
| `ChassisType` | String | ChassisType shall indicate the physical form factor for the type of chassis.<br>Allowed values:<br>“Pod” - A collection of equipment racks in a large, likely transportable, container<br>“Rack” - An equipment rack, typically a 19-inch wide freestanding unit<br>“Blade” - An enclosed or semi-enclosed, typically vertically-oriented, system chassis which must be plugged into a multi-system chassis to function normally<br>“Enclosure” - A generic term for a chassis that does not fit any other description<br>“RackMount” - A single system chassis designed specifically for mounting in an equipment rack<br>“Expansion” - A chassis which expands the capabilities or capacity of another chassis<br>“Zone” - A logical division or portion of a physical chassis that contains multiple devices or systems that cannot be physically separated |

| | | |
|---|---|---|
| | | "Sled" - An enclosed or semi-enclosed, system chassis which must be plugged into a multi-system chassis to function normally, similar to a blade type chassis<br>"Shelf" - An enclosed or semi-enclosed, typically horizontally-oriented, system chassis which must be plugged into a multi-system chassis to function normally<br>"Drawer" - An enclosed or semi-enclosed, typically horizontally-oriented, system chassis which may be slid into a multi-system chassis<br>"Module" - A small, typically removable, chassis or card which contains devices for a particular subsystem or function<br>"Other" - A chassis that does not fit any of these definitions |
| `Manufacturer` | String | This is the manufacturer of this chassis |
| `Model` | String | This is the model number for the chassis |
| `SKU` | String | This is the SKU for this chassis |
| `SerialNumber` | String | The serial number for this chassis |
| `PartNumber` | String | The part number for this chassis |
| `AssetTag` | String | The user assigned asset tag for this chassis |
| `IndicatorLED` | String | The state of the indicator LED, used to identify the chassis. Allowed values:<br>"Lit"<br>"Blinking"<br>"Off"<br>"Unknown" |
| `Status` | Object | See Section 5.1 for status of resource. |
| `Oem` | Object | Oem extension object<br>"Intel_RackScale" extensions:<br><table><tr><th>Attribute</th><th>Type</th><th>Description</th></tr><tr><td>`Location`</td><td>Object</td><td>Property that shows this chassis ID and its parent. Contains the following properties:<br>**"Id"** – string containing location ID of the chassis<br>**"ParentId"** – string containing location ID of parent chassis</td></tr><tr><td>`RMMPresent`</td><td>Boolean, null</td><td>RMM presence in a rack</td></tr><tr><td>`RackSupportDisaggregatedPowerCooling`</td><td>Boolean, null</td><td>Indicates if Rack support is disaggregated (shared) power and cooling capabilities</td></tr><tr><td>`UUID`</td><td>String, null</td><td>Chassis unique ID</td></tr><tr><td>`GeoTag`</td><td>String, null</td><td>Provides info about the geographical location of this chassis</td></tr></table> |
| `Actions` | Object | Chassis actions |
| `ThermalZones` | Object | A reference to ThermalZones collection |
| `PowerZones` | Object | A reference to PowerZones collection |
| `Power` | Object | A reference to the power properties (power supplies, power policies, sensors) for this chassis.<br>If a Power resource is not available, property should not be included in Chassis response. |
| `Thermal` | Object | A reference to the thermal properties (fans, cooling, sensors) for this chassis.<br>If a Power resource is not available, property should not be included in Chassis response. |
| `Links` | Object | Link sections |
| `ComputerSystems` | Array | An array of references to the computer systems contained in this chassis. This will only reference ComputerSystems that are directly and wholly contained in this chassis |

| | | |
|---|---|---|
| Switches | Array | An array of references to the Ethernet switches contained in this chassis. |
| ManagedBy | Array | An array of references to the managers contained in this chassis |
| ManagersInChassis | Array | An array of references to the managers located in this chassis |
| ContainedBy | Object | The value of this property shall be a reference to the resource that represents the chassis that contains this chassis and shall be of type Chassis. |
| Contains | Array | An array of references to any other chassis that this chassis has in it |
| PoweredBy | Array | An array of ID[s] of resources that power this chassis. Normally the ID will be a chassis or a specific set of powerSupplies |
| CooledBy | Array | An array of ID[s] of resources that cool this chassis. Normally the ID will be a chassis or a specific set of fans. |

## 4.6.1 Operations

### 4.6.1.1 GET

Request:

```
GET /redfish/v1/Chassis/1
Content-Type: applicaton/json
```

Response:


```
{
    "@odata.context": "/redfish/v1/$metadata#Chassis/Members/$entity",
    "@odata.id": "/redfish/v1/Chassis/Rack1",
    "@odata.type": "#Chassis.v1_3_0.Chassis",
    "Id": "Rack1",
    "ChassisType": "RackMount",
    "Name": "name-as-string",
    "Description": "description-as-string",
    "Manufacturer": "Intel Corporation",
    "Model": "model-as-string",
    "SKU": "sku-as-string",
    "SerialNumber": "serial-number-as-string",
    "PartNumber": "part-number-as-string",
    "AssetTag": null,
    "IndicatorLED": "Unknown",
    "Status": {
        "State": "Enabled",
        "Health": "OK"
    },
    "Oem": {
        "Intel_RackScale": {
            "@odata.type": "#Intel.Oem.RackChassis",
            "Location": {
                "Id": "Rack1",
                "ParentId": "Pod1"
            },
            "RMMPresent": true,
            "RackSupportsDisaggregatedPowerCooling": true,
            "UUID": "Unique ID",
            "GeoTag": "54.348103, 18.645172"
        }
    },
```

```
    "ThermalZones": {
        "@odata.id": "/redfish/v1/Chassis/Rack1/ThermalZones"
    },
    "PowerZones": {
        "@odata.id": "/redfish/v1/Chassis/Rack1/PowerZones"
    },
    "Thermal": {
      "@odata.id": "/redfish/v1/Chassis/Rack1/Thermal"
    },
    "Power": {
      "@odata.id": "/redfish/v1/Chassis/Rack1/Power"
    },
    "Links": {
        "@odata.type": "#Chassis.v1_2_0.Links",
        "Contains": [
            {
                "@odata.id": "/redfish/v1/Chassis/Drawer1"
            }
        ],
        "ContainedBy": null,
        "ComputerSystems": [],
        "ManagedBy": [
            {
                "@odata.id": "/redfish/v1/Managers/RMM"
            }
        ],
        "ManagersInChassis": [
            {
                "@odata.id": "/redfish/v1/Managers/RMM"
            }
        ],
        "Oem": {
          "Intel_RackScale": {
            "@odata.type": "#Intel.Oem.ChassisLinks",
            "Switches": []
          }
       }
    }
}
```

#### 4.6.1.2 PUT

Operation is not allowed on this resource.

#### 4.6.1.3 PATCH

Following properties can be updated by a PATCH operation:

| Attribute | Type | Required | Description |
|---|---|---|---|
| `AssetTag` | String | No | The user assigned asset tag for this chassis. |
| `Oem->Intel_RackScale->Location->Id` | String | No | The user assigned location id for this chassis. It can be changed only for a Rack Chassis. |

```
PATCH /redfish/v1/Chassis/Rack1
Content-Type: application/json
```

```
{
    "AssetTag": "Rack#1",
    "Oem": {
        "Intel_RackScale": {
            "Location": {
                "Id": "1234"
            }
        }
    }
}
```

Response

```
HTTP/1.1 204 No Content
```

or

```
HTTP/1.1 200 OK
```

with full resource representation.

#### 4.6.1.4 POST

Operation is not allowed on this resource.

#### 4.6.1.5 DELETE

Operation is not allowed on this resource.

## 4.7 PowerZone collection

PowerZone collection resource.

**Table 8 PowerZone collection attributes**

| Name | PowerZone Collection | |
|---|---|---|
| Type URI | /redfish/v1/Chassis/{chassisID}/PowerZones | |
| **Attribute** | **Type** | **Description** |
| `Name` | String | Name of collection |
| `Members@odata.count` | Number | Collection members count |
| `Members` | Array | Contains the members of this collection. |

### 4.7.1 Operations

#### 4.7.1.1 GET

Request:

```
GET /redfish/v1/Chassis/Rack1/PowerZones
Content-Type: application/json
```

Response:

```
{
    "@odata.context":
"/redfish/v1/$metadata#PowerZoneCollection.PowerZoneCollection",
    "@odata.id": "/redfish/v1/Chassis/Rack1/PowerZones",
    "@odata.type": "#PowerZoneCollection.PowerZoneCollection",
    "Name": "Power Zones Collection",
```

```
    "Members@odata.count": 1,
    "Members": [
        {
            "@odata.id": "/redfish/v1/Chassis/Rack1/PowerZones/Power1"
        }
    ]
}
```

#### 4.7.1.2 PUT

Operation is not allowed on this resource.

#### 4.7.1.3 PATCH

Operation is not allowed on this resource.

#### 4.7.1.4 POST

Operation is not allowed on this resource.

#### 4.7.1.5 DELETE

Operation is not allowed on this resource.

## 4.8 PowerZone

This resource shall be used to represent a power zone resource for an Intel® RSD implementation. It contains Power Supplies and Power Control information.

**Table 9 PowerZone attributes**

| **Name** | PowerZone | |
|---|---|---|
| **Type URI** | /redfish/v1/Chassis/{chassisId}/PowerZones/{zoneID} | |
| **Attribute** | **Type** | **Description** |
| `Id` | String | Resource identifier |
| `Name` | String | Name of service root |
| `Description` | String | Provides a description of this resource and is used for commonality in the schema definitions |
| `Status` | Object | See Section 5.1 for status of resource. |
| `RackLocation` | Object | Zone physical location. |

| **Attribute** | **Type** | **Description** |
|---|---|---|
| `RackUnits` | String | Rack.Units: indicates the rack unit type.<br>“RU”: rack unit, each RU 44.5mm<br>“OU”: openU, each OU 48mm |
| `XLocation` | Number | The horizontal location within uLocation, from left to right(1.. MAXIMUM)<br>0 indicate not available. |
| `ULocation` | Number | The index of the top-most U of the component, from top to bottom(1..MAXIMUM)<br>0 indicate not available |
| `UHeight` | Number | The height of managed zone, e.g. 8 for 8U, 16 for 16U |

| | | |
|---|---|---|
| MaxPSUsSupported | Number | The maximum number of Power Supply Units supported by PowerZone. |
| Presence | String | Indicates the aggregated Power Supply Unit presence information<br>Aggregated Power Supply Unit presence format:<br>Length of string indicate total slot of Power Supply Units in PowerZone.<br>For each byte the string:<br>"1" means present<br>"0" means not present |
| NumberOfPSUsPresent | Number | Indicates the number of existing Power Supply Units in PowerZone. |
| PowerConsumedWatts | Number | The total power consumption of PowerZone, sum of trays' power consumption |
| PowerOutputWatts | Number | The total power production of PowerZone, sum of PSUs' output |
| PowerCapacityWatts | Number | The maximum power capacity supported by PowerZone. |
| PowerSupplies | Array | Details of the power supplies associated with this system or device. Detailed description in Table 10 |
| Actions | Object | Actions that can be executed on zone |
| Links | Object | Links to resources related to this resource. |

**Table 10 PowerSupply attributes**

| Attribute | Type | Description |
|---|---|---|
| Name | String | Power Supply name |
| PowerCapacityWatts | Number | The maximum capacity of this Power Supply |
| LastPowerOutputWatts | Number | The average power output of this Power Supply |
| Manufacturer | String | The manufacturer of this Power Supply |
| ModelNumber | String | The model number for this Power Supply |
| FirmwareRevision | String | The firmware version for this Power Supply |
| SerialNumber | String | The serial number for this Power Supply |
| PartNumber | String | The part number for this Power Supply |
| Status | Object | Status of resource please see Section 5.1 |
| RackLocation | Object | |

| Attribute | Type | Description |
|---|---|---|
| RackUnits | String | Rack.Units: indicates the rack unit type.<br>"RU": rack unit, each RU 44.5mm<br>"OU": openU, each OU 48mm |
| XLocation | Number | The horizontal location within uLocation, from left to right(1.. MAXIMUM)<br>0 indicate not available. |
| ULocation | Number | The index of the top-most U of the component, from top to bottom(1..MAXIMUM)<br>0 indicate not available |
| UHeight | Number | The height of component, e.g. 8 for 8U, 16 for 16U |

### 4.8.1 Operations

#### 4.8.1.1 GET

Request:

```
GET /redfish/v1/Chassis/Rack1/PowerZones/Power1
Content-Type: applicaton/json
```

Response:


```
{
    "@odata.context":
"/redfish/v1/$metadata#Chassis/Rack/PowerZones/Members/$entity",
    "@odata.id": "/redfish/v1/Chassis/Rack1/PowerZones/1",
    "@odata.type": "PowerZone.v1_0_0.PowerZone",
    "Id": "1",
    "Name": "power zone 1",
    "Description": "power zone 1",
    "Status": {
        "State": "Enabled",
        "Health": "OK",
        "HealthRollup": "OK"
    },
    "RackLocation": {
        "RackUnits": "OU",
        "XLocation": 0,
        "ULocation": 1,
        "UHeight": 8
    },
    "MaxPSUsSupported": 6,
    "Presence": "111111",
    "NumberOfPSUsPresent": 6,
    "PowerConsumedWatts": 2000,
    "PowerOutputWatts": 2000,
    "PowerCapacityWatts": 3000,
    "PowerSupplies": [
        {
            "Name": "Power supply 1",
            "Status": {
                "State": "Enabled",
                "Health": "OK",
                "HealthRollup": "OK"
            },
            "RackLocation": {
                "RackUnits": "OU",
                "XLocation": 0,
                "ULocation": 1,
                "UHeight": 8
            },
            "SerialNumber": "",
            "Manufacturer": "",
            "ModelNumber": "",
            "PartNumber": "",
            "FirmwareRevision": "",
            "PowerCapacityWatts": 300,
            "LastPowerOutputWatts": 48
```

```
        }
    ],
    "Actions": {},
    "Links": {}
}
```

#### 4.8.1.2 PUT

Operation is not allowed on this resource.

#### 4.8.1.3 PATCH

Operation is not allowed on this resource.

#### 4.8.1.4 POST

Operation is not allowed on this resource.

#### 4.8.1.5 DELETE

Operation is not allowed on this resource.

## 4.9 ThermalZone collection

ThermalZone collection resource.

**Table 11 ThermalZone collection attributes**

| Name | ThermalZone Collection | |
|---|---|---|
| Type URI | /redfish/v1/Chassis/{chassisID}/ThermalZones | |
| **Attribute** | **Type** | **Description** |
| `Name` | String | Name of collection |
| `Members@odata.count` | Number | Collection members count |
| `Members` | Array | Contains the members of this collection. |

### 4.9.1 Operations

#### 4.9.1.1 GET

Request:

```
GET /redfish/v1/Chassis/Rack1/ThermalZones
Content-Type: application/json
```

Response:

```
{
    "@odata.context":
"/redfish/v1/$metadata#ThermalZoneCollection.ThermalZoneCollection",
    "@odata.id": "/redfish/v1/Chassis/Rack1/ThermalZones",
    "@odata.type": "#ThermalZoneCollection.ThermalZoneCollection",
    "Name": "Thermal Zones Collection",
    "Members@odata.count": 1,
    "Members": [
        {
            "@odata.id": "/redfish/v1/Chassis/Rack1/ThermalZones/Thermal1"
        }
```

```
    ]
}
```

#### 4.9.1.2 PUT

Operation is not allowed on this resource.

#### 4.9.1.3 PATCH

Operation is not allowed on this resource.

#### 4.9.1.4 POST

Operation is not allowed on this resource.

#### 4.9.1.5 DELETE

Operation is not allowed on this resource.

## 4.10 ThermalZone

This resource shall be used to represent a thermal zone resource for an Intel® RSD implementation. It contains Fans and temperature information.

**Table 12 ThermalZone attributes**

| **Name** | ThermalZone | |
|---|---|---|
| **Type URI** | /redfish/v1/Chassis/{chassisId}/ThermalZones/{zoneID} | |
| **Attribute** | **Type** | **Description** |
| `Id` | String | Resource identifier |
| `Name` | String | Name of service root |
| `Description` | String | Provides a description of this resource and is used for commonality in the schema definitions |
| `Status` | Object | See Section 5.1 for status of resource. |
| `RackLocation` | Object | Zone physical location.<br>**Attribute** / **Type** / **Description**<br>`RackUnits` / String / Rack.Units: indicates the rack unit type.<br>"RU": rack unit, each RU 44.5mm<br>"OU": openU, each OU 48mm<br>`XLocation` / Number / The horizontal location within uLocation, from left to right(1.. MAXIMUM)<br>0 indicate not available.<br>`ULocation` / Number / The index of the top-most U of the component, from top to bottom(1..MAXIMUM)<br>0 indicate not available<br>`UHeight` / Number / The height of managed zone, e.g. 8 for 8U, 16 for 16U |
| `Presence` | String | Indicates the aggregated Power Supply Unit presence information<br>Aggregated Power Supply Unit presence format:<br>Length of string indicate total slot of Power Supply Units in PowerZone.<br>For each byte the string:<br>"1" means present |

| | | |
|---|---|---|
| | | "0" means not present |
| DesiredSpeedPWM | Number | The desired FAN speed in current ThermalZone present in PWM unit |
| DesiredSpeedRPM | Number | The desired FAN speed in current ThermalZone present in RPM unit |
| MaxFansSupported | Number | Number of maximum fans that can be installed in a given Thermal Zone |
| NumberOfFansPresent | Number | The existing number of fans in current ThermalZone. |
| VolumetricAirflow | Number | Rack Level PTAS Telemetry – Volumetric airflow in current ThermalZone. |
| Temperatures | Array | Array of temperature sensors. Detailed description in Table 14 |
| Fans | Array | Details of the fans associated with this thermal zone. Detailed description in Table 10 |
| Actions | Object | Actions that can be executed on thermal zone |
| Links | Object | Links to resources related to this resource. |

**Table 13 Fan attributes**

| Attribute | Type | Description |
|---|---|---|
| Name | String | Power Control Function name |
| ReadingRPM | Number | Fan RPM reading |
| Status | Object | Status of resource please see Section 5.1 |
| RackLocation | Object | Fan physical location.<br>**Attribute** / **Type** / **Description**<br>RackUnits / String / Rack.Units: indicates the rack unit type.<br>"RU": rack unit, each RU 44.5mm<br>"OU": openU, each OU 48mm<br>XLocation / Number / The horizontal location within uLocation, from left to right(1.. MAXIMUM)<br>0 indicate not available.<br>ULocation / Number / The index of the top-most U of the component, from top to bottom(1..MAXIMUM)<br>0 indicate not available<br>UHeight / Number / The height of managed fun, e.g. 8 for 8U, 16 for 16U |

**Table 14 Temperature attributes**

| Attribute | Type | Description |
|---|---|---|
| Name | String | Power Supply name |
| ReadingCelsius | Number | Current value of the temperature sensor's reading |
| PhysicalContext | String | Describes the area or device to which this temperature measurement applies:<br>"Intake" - The intake point of the chassis<br>"Exhaust" - The exhaust point of the chassis<br>"Backplane" - A backplane within the chassis<br>"PowerSupply" – A power supply<br>"SystemBoard" - The system board (PCB)<br>"ComputeBay" - Within a compute bay<br>"PowerSupplyBay" - Within a power supply bay |

| Attribute | Type | Description |
|---|---|---|
| Status | Object | See Section 5.1 for status of resource. |

## 4.10.1 Operations

### 4.10.1.1 GET

Request:

```
GET /redfish/v1/Chassis/Rack1/ThermalZones/Thermal1
Content-Type: applicaton/json
```

Response:


```
{
    "@odata.context":
"/redfish/v1/$metadata#Chassis/Rack/ThermalZones/Members/$entity",
    "@odata.type": "ThermalZone.v1_0_0.ThermalZone",
    "@odata.id": "/redfish/v1/Chassis/Rack1/ThermalZones/1",
    "Id": "1",
    "Name": "thermal zone 1",
    "Description": "thermal zone 1",
    "RackLocation": {
        "RackUnits": "OU",
        "XLocation": 0,
        "ULocation": 1,
        "UHeight": 8
    },
    "Presence": "111100",
    "DesiredSpeedPWM": 50,
    "DesiredSpeedRPM": 3000,
    "MaxFansSupported": 6,
    "NumberOfFansPresent": 6,
    "VolumetricAirflow": 80,
    "Temperatures": [
        {
            "Name": "Inlet Temperature",
            "Status": {
                "State": "Enabled",
                "Health": "OK",
                "HealthRollup": null
            },
            "ReadingCelsius": 21,
            "PhysicalContext": "Intake"
        },
        {
            "Name": "Outlet Temperature",
            "Status": {
                "State": "Enabled",
                "Health": "OK",
                "HealthRollup": null
            },
            "ReadingCelsius": 35,
            "PhysicalContext": "Exhaust"
```

```
        }
    ],
    "Status": {
        "State": "Enabled",
        "Health": "OK",
        "HealthRollup": null
    },
    "Fans": [
        {
            "Name": "Fan 1",
            "Status": {
                "State": "Enabled",
                "Health": "OK",
                "HealthRollup": null
            },
            "ReadingRPM": 0,
            "RackLocation": {
                "RackUnits": "OU",
                "XLocation": 0,
                "ULocation": 1,
                "UHeight": 8
            }
        }
    ],
    "Actions": {},
    "Links": {} }
```


#### 4.10.1.2 PUT

Operation is not allowed on this resource.

#### 4.10.1.3 PATCH

Operation is not allowed on this resource.

#### 4.10.1.4 POST

Operation is not allowed on this resource.

#### 4.10.1.5 DELETE

Operation is not allowed on this resource.

## 4.11 Storage service collection

Intel® RSD storage service collection resource – provides a collection of available storage services. Table 15 lists the attributes.

**Table 15 Storage service collection attributes**

| **Name** | Storage service | | |
|---|---|---|---|
| **Type URI** | /redfish/v1/Services | | |
| **Attribute** | **Type** | **Mandatory** | **Description** |
| `Name` | String | Yes | Name of service collection. |
| `Members@odata.count` | Number | No | Collection members count |
| `Members` | Array | No | Contains the members of this collection. |

### 4.11.1 Operations

#### 4.11.1.1 GET

Request:

```
GET /redfish/v1/Services
Content-Type: application/json
```

Response:

```
{
    "@odata.context": "/redfish/v1/$metadata#Services",
    "@odata.id": "/redfish/v1/Services",
    "@odata.type": "#StorageServiceCollection.StorageServiceCollection",
    "Name": "Storage Services Collection",
    "Description": "Collection of Storage Services",
    "Members@odata.count": 1,
    "Members": [
        {
            "@odata.id": "/redfish/v1/Services/RSS1"
        }
    ]
}
```

#### 4.11.1.2 PUT

Operation is not allowed on this resource.

#### 4.11.1.3 PATCH

Operation is not allowed on this resource.

#### 4.11.1.4 POST

Operation is not allowed on this resource.

#### 4.11.1.5 DELETE

Operation is not allowed on this resource.

## 4.12 Composed node collection

Intel® RSD Composed Node collection resource – provides a collection of all logical nodes. Table 16 lists the attributes.

**Table 16 Composed node collection attributes**

| **Name** | Composed node collection | | |
|---|---|---|---|
| **Type URI** | /redfish/v1/Nodes | | |
| **Attribute** | **Type** | **Mandatory** | **Description** |
| `Name` | String | Yes | Name of collection |
| `Members@odata.count` | Number | No | Collection members count |
| `Members` | Array | No | Contains the members of this collection. |
| `Actions` | Object | No | Actions available:<br>**Allocate** – this action is the first mandatory step to create a composed node. In response to this action proper resources will be found and allocated for node composition. Node resource will be created and the URL (link) of this node will be returned. |

| | | | |
|---|---|---|---|
| | | | To allocate a Composed Node using PODM REST API, it is necessary to create a JSON template describing requested resources.<br>The JSON template may contain various details concerning resources to be used in the Composed Node. All JSON template elements are optional, but shouldn't be mutually exclusive. It is possible to supply PODM with a JSON template containing no specific requirements (e.g. {} – a pair of empty curly braces in the HTTP request body) thus allowing PODM to propose a Composed Node containing resources chosen arbitrarily by PODM.<br>The format of the JSON template (action payload) is described in Table 17.<br><br>For more information about node allocation and assembly, refer to the PODM_Allocation_Guide document. |

**Table 17 Composed node Allocation action attributes**

| Attribute | Type | Mandatory | Description |
|---|---|---|---|
| `Name` | String | No | Name of composed node.<br>Note: Because ComposedNode is a Redfish resource - it's Name field is mandatory, so an attempt to directly set a **null** value results in an expected error. PODM will set a default name for a newly created ComposedNode resource only upon not supplying the Name attribute. |
| `Description` | String | No | Description of composed node |
| `Processors` | Array | No | Array of requirements for processor for composed node. Each processor requirement may contain one or more optional attributes: |

| Attribute | Type | Description |
|---|---|---|
| Model | String | Processor model that should be used for composed node (exact model name) |
| TotalCores | Number | Minimum number of processor cores |
| AchievableSpeedMHz | Number | Minimum achievable processor operating frequency. |
| InstructionSet | String | Processor supported instruction set.<br>“x86” – x86 32-bit<br>“x86-64” – x86 64-bit<br>“IA-64” – Intel IA-64<br>“ARM-A32” – ARM* 32-bit<br>“ARM-A64” – ARM 64-bit<br>“MIPS32” – MIPS 32-bit<br>“MIPS64” – MIPS 64-bit<br>“OEM” – OEM-defined |
| Resource | Object | Reference to a particular processor that should be used in composed node |
| Chassis | Object | Link to chassis object within this processor should be contained. |
| Brand | String | Brand of CPU that should be used to allocate node. Allowable values:<br>Xeon family: E3, E5, E7 |

| Attribute | Type | Mandatory | Description |
|---|---|---|---|
| | | | |
| | | | |
| | | | SoC/Atom family: X3 (Avoton), X5 (Broadwell-DE), X7<br>Core family: I3, I5, I7<br>“Unknown” – processor doesn’t fit to any above categories |
| | | | Capabilities \| Array \| Array of strings describing processor capabilities (like reported in /proc/cpuinfo flags), such as:<br>“sse” - Streaming SIMD Extensions<br>“avx” - Advanced Vector Extensions |
| Memory | Array | No | Array of requirements for memory for composed node. |

| Attribute | Type | Description |
|---|---|---|
| CapacityMiB | Number | Minimum memory capacity requested for composed node |
| MemoryDeviceType | String | Type details of DIMM:<br>“DDR”<br>“DDR2”<br>“DDR3”<br>“DDR4”<br>“DDR4 SDRAM”<br>“DDR4E SDRAM”<br>“LPDDR4SDRAM”<br>“DDR3 SDRAM”<br>“LPDDR3 SDRAM”<br>“DDR2 SDRAM”<br>“DDR2 SDRAM-FB-DIMM”<br>“DDR2SDRAM-FB-DIMM PROBE”<br>“DDR SGRAM”<br>“DDR SDRAM”<br>“ROM”<br>“SDRAM”<br>“EDO”<br>“FastPageMode”<br>“PipelinedNibble” |
| SpeedMHz | Number | Minimum supported memory speed. |
| Manufacturer | String | Requested memory manufacturer. |
| DataWidthBits | Number | Requested memory data width in bits. |
| Resource | Object | Reference to a particular memory module that should be used in composed node |
| Chassis | Object | Link to chassis object within this memory DIMM should be contained. |

| Attribute | Type | Mandatory | Description |
|---|---|---|---|
| RemoteDrives | Array | No | Array of requirements for remote drives that should be created/connected to composed node.<br>**Attribute** / **Type** / **Description**<br>CapacityGiB / Number / Minimum drive capacity requested for composed node<br>iSCSIAddress / String / Defines TargetIQN of RemoteTarget to be set for new Remote Target (should be unique in PODM).<br>Note: Master property is required for creating new target.<br>Master / Object / Defines master logical volume that should be taken to create new remote target. It contains properties described in Table 18. |
| LocalDrives | Array | No | Array of requirements for local drives for composed node.<br>**Attribute** / **Type** / **Description**<br>CapacityGiB / Number / Minimum drive capacity requested for composed node<br>Type / String / Drive type<br>“HDD”<br>“SSD”<br>MinRPM / Number / Minimum rotation speed of requested drive.<br>SerialNumber / String / Serial number of requested drive<br>Interface / String / Interface of requested drive:<br>“SAS”<br>“SATA”<br>“NVMe”<br>Resource / Object / Reference to particular local drive that should be used in composed node<br>Chassis / Object / Link to chassis object within this drive should be contained.<br>FabricSwitch / Boolean / Determine if local drive should be connected using fabric switch or local connected. |
| EthernetInterfaces | Array | No | Array of requirements for ethernet interfaces of composed node<br>**Attribute** / **Type** / **Description**<br>SpeedMbps / Number / Minimum speed of Ethernet interface requested for composed node |

| Attribute | Type | Mandatory | Description | | |
|---|---|---|---|---|---|
| | | | VLANs | Array | Array of VLANs that should be configured on connected switch port for composed node for given Ethernet interface in the following format:<br>**VLANId** – number indicating VLAN Id<br>**Tagged** – Boolean value describing if given VLAN is tagged.<br>**Deprecated**: This is going to be removed in future releases of RSD. Equivalent functionality will be provided via CRUD operation on Ethernet Switch port VLANs. |
| | | | PrimaryVLAN | Number | Primary VLAN ID that should be set for a given Ethernet Interface.<br>**Deprecated**: This is going to be removed in future releases of RSD. Equivalent functionality will be provided via CRUD operation on Ethernet Switch port. |
| | | | Resource | Object | Reference to a particular Ethernet interface that should be used in composed node. |
| | | | Chassis | Object | Link to chassis object within this network interface should be contained. |
| `TotalSystemMemory` | Number | No | Minimum total amount of memory in composed node. | | |
| `TotalSystemCoreCount` | Number | No | Minimum total processor core count in composed node. | | |

**Table 18 Remote master target properties**

| Attribute | Type | Description |
|---|---|---|
| `Type` | String | Type of replication of master drive:<br>Clone – volume should be cloned<br>Snapshot – Copy on Write should be created from indicated volume |
| `Resource` | Object | Reference to logical volume that should be used as master for replication. |

## 4.12.1 Operation

### 4.12.1.1 GET

Request:

```
GET /redfish/v1/Nodes
```

```
Content-Type: application/json
```

Response:

```
{
    "@odata.context": "/redfish/v1/$metadata#Nodes",
    "@odata.id": "/redfish/v1/Nodes",
    "@odata.type": "#ComposedNodeCollection.ComposedNodeCollection",
    "Name": "Composed Nodes Collection",
    "Members@odata.count": 1,
    "Members": [
        {
            "@odata.id": "/redfish/v1/Nodes/Node1"
        }
    ],
    "Actions": {
        "#ComposedNodesCollection.Allocate": {
            "target": "/redfish/v1/Nodes/Actions/Allocate"
        }
    }
}
```

#### 4.12.1.2 PUT

Operation is not allowed on this resource.

#### 4.12.1.3 PATCH

Operation is not allowed on this resource.

#### 4.12.1.4 POST

***Note:*** Currently a user can request allocation of a single node with a single request. Node components—CPU, memory, local storage, network interface—must be located on a single physical blade. Remote storage can be located anywhere in the Pod.

***Note:*** See JSON example below. For more details, see Section 6.1: *Creating new Composed Node - explanation.*

Request:

```
POST /redfish/v1/Nodes/Actions/Allocate
Content-Type: application/json
{
        "Name": "My first composed node",
        "Description": "Test node",
        "Processors": [{
                "Model": "Multi-Core Intel(R) Xeon(R) processor 7xxx Series",
                "TotalCores": 2,
                "AchievableSpeedMHz": 2000,
                "InstructionSet": "x86",
                "Oem": {
                        "Brand": "E5",
                        "Capabilities": [
                                "sse"
                        ],

                },
                "Resource": {
                        "@odata.id":
"/redfish/v1/Systems/System1/Processors/CPU1"
                }
        }],
```

```
        "Memory": [{
                "CapacityMiB": 16000,
                "MemoryDeviceType": "DDR3",
                "SpeedMHz": 1600,
                "Manufacturer": "Intel",
                "DataWidthBits": 64,
                "Resource": {
                        "@odata.id": "/redfish/v1/Systems/System1/Memory/Dimm1"
                },
                "Chassis": {
                        "@odata.id": "/redfish/v1/Chassis/Rack1"
                }
        }],
        "RemoteDrives": [{
                "CapacityGiB": 80,
                "iSCSIAddress": "iqn.oem.com:fedora21",
                "Master": {
                        "Type": "Snapshot",
                        "Resource": {
                                "@odata.id":
"/redfish/v1/Services/RSS1/LogicalDrives/sda1"
                        }
                }
        }],
        "LocalDrives": [{
                "CapacityGiB": 500,
                "Type": "HDD",
                "MinRPM": 5400,
                "SerialNumber": "12345678",
                "Interface": "SATA",
                "Resource": {
                        "@odata.id": "redfish/v1/Chassis/Blade1/Drives/Disk1"
                },
                "FabricSwitch": false
        }],
        "EthernetInterfaces": [{
                "SpeedMbps": 1000,
                "PrimaryVLAN": 100,
                "VLANs": [{
                        "VLANId": 100,
                        "Tagged": false
                }],
                "Resource": {
                        "@odata.id":
"/redfish/v1/Systems/System1/EthernetInterfaces/LAN1"
                }
        }],
        "Oem": {
        },
        "TotalMemoryCapacityMiB": 32000,
        "TotalProcessorCoreCount": 2
}
```


Response:

```
HTTP/1.1 201 Created
```

```
Location: http://<IP>:<Port>/redfish/v1/Nodes/2
```

#### 4.12.1.5 DELETE

Operation is not allowed on this resource.

## 4.13 Composed node

Composed node resource – provides detailed information about an assembled logical node identified by {nodeID}. Table 19 lists the attributes.

**Table 19 Composed node attributes**

| **Name** | Composed node | | |
|---|---|---|---|
| **Type URI** | /redfish/v1/Nodes/{nodeID} | | |
| **Attribute** | **Type** | **Mandatory** | **Description** |
| `Id` | String | Yes | Provides an ID of this resource |
| `Name` | String | Yes | Name of composed node |
| `Description` | String | No | User provided node description |
| `UUID` | String | No | UUID of computer system used as a base for this node. |
| `PowerState` | String (enum) | No | This is the current power state of the node<br>“On” - The system is powered on<br>“Off” - The system is powered off, although some components may continue to have AUX power such as management controller<br>“PoweringOn” - A temporary state between Off and On. This temporary state can be very short.<br>“PoweringOff” - A temporary state between On and Off. The power off action can take time while the OS is in the shutdown process. |
| `Processors` | Object | Yes | **Name** / **Type** / **Mandatory** / **Description**<br>`Count` / Number / No / Number of CPUs<br>`Model` / String, Null / No / Basic information about processor model<br>`Status` / Object / No / See Section 5.1 for status of resource. |
| `Memory` | Object | Yes | **Name** / **Type** / **Mandatory** / **Description**<br>`TotalSystemMemoryGiB` / Number / No / Amount of installed memory in GiB<br>`Status` / Object / No / See Section 5.1 for status of resource. |
| `Status` | Object, null | No | See Section 5.1 for status of resource. |
| `ComposedNodeState` | String (enum) | Yes | Current state of assembly process for this node.<br>• *Allocating*: Allocating resources for node is in progress. Next state can be *Allocated* or *Failed*<br>• *Allocated*: Node resources has been allocated, but assembly not started yet. After `ComposedNode.Assemble` action state will progress to *Assembling*.<br>• *Assembling*: Assembly process initiated, but not finished yet. When assembly is done it will change into *Assembled*.<br>• *Assembled*: Node successfully assembled.<br>• *Failed*: Allocation or assembly process failed, or in runtime one of composing components was removed or transitioned in error state. |
| `Boot` | Object | No | **Name** / **Type** / **Required** / **Description**<br>`BootSourceOverrideEnabled` / String, Null / No / State of the Boot Source Override feature. Proper values:<br>“Disabled”<br>“Once”<br>“Continuous” |

| Name | Composed node | | |
|---|---|---|---|
| Type URI | /redfish/v1/Nodes/{nodeID} | | |
| Attribute | Type | Mandatory | Description |

| | | | |
|---|---|---|---|
| BootSourceOverrideTarget | String, Null | No | The current boot source to be used at next boot instead of the normal boot device, if BootSourceOverrideEnabled is true. |
| BootSourceOverrideTarget@Redfish.AllowableValues | Array | No | Array of supported boot sources. See: **Error! Reference ource not found.** |
| BootSourceOverrideMode | String, Null | No | The BIOS Boot Mode (either Legacy or UEFI) to be used when BootSourceOverrideTarget boot source is booted from. |
| BootSourceOverrideMode@Redfish.AllowableValues | Array | No | Array of supported boot modes |

| Attribute | Type | Mandatory | Description |
|---|---|---|---|
| Oem | Object, Null | No | OEM defined object |
| Links | Object | No | Link section: |

| Name | Type | Required | Description |
|---|---|---|---|
| ComputerSystem | Object, null | Yes | Reference to ComputerSystem resource used to compose this node. |
| Processors | Array | No | Array of references to Processor resources |
| Memory | Array | No | Array of references to Memory resources |
| RemoteDrives | Array | No | An array of references to the remote storage drives |
| LocalDrives | Array | No | An array of references to the computer system local storage drives |
| EthernetInterfaces | Array | No | Array of links to Ethernet Interface collection associated with this Composed Node |
| ManagedBy | Array | No | An array of references to Managers responsible for this Composed Node |

| Attribute | Type | Mandatory | Description |
|---|---|---|---|
| Actions | Object | Yes | Actions available for this node:<br>• *Reset* action with the following values:<br>○ On - Turn the system on<br>○ ForceOff - Turn the system off immediately (non-graceful) shutdown<br>○ GracefulRestart - Perform a graceful system shutdown followed by a restart of the system<br>○ ForceRestart - Perform an immediate (non-graceful) shutdown, followed by a restart of the system<br>○ Nmi - Generate a Diagnostic Interrupt (usually an NMI on x86 systems) to cease normal operations, perform diagnostic actions and typically halt the system.<br>○ ForceOn - Turn the system on immediately<br>○ PushPowerButton - Simulate the pressing of the physical power button on this system |

| Name | Composed node | | |
|---|---|---|---|
| Type URI | /redfish/v1/Nodes/{nodeID} | | |
| Attribute | Type | Mandatory | Description |
| | | | ○ GracefulShutdown – initiate a soft-shutdown of OS via ACPI<br><br>• ***Assemble***: Doesn't consume any parameters. Second step of creating a composed node (after Allocate Action on Nodes collection). This action will trigger PODM to start assembly of the composed node process – initiate `ComposedNodeState change from Allocated state into Assembling state`<br>After finished assembling, composed node will stay in Off PowerState. To change its state, one needs to execute Reset action with "On" parameter (make sure that Boot is set propertly).<br>• **AttachEndpoint**: This action allows to attach endpoint (functional device) like PCI drives on PNC switch, from available pool to composed node. Action can be performed when Composed Node is *Assembled* or *Failed.* One of the following parameters are required:<br>○ Resource – link to endpoint (realized by PCI device function). Must be connected to PNC switch servicing ComputerSystem hosting this ComposedNode<br>○ CapacityGiB – requested capacity of PNC drive. PODM will find available drive in the pool and connect it to this node.<br>**Note: Currently only PCI NVMe drives are supported by this action.**<br>• **DetachEndpoint**: Action used to detach already connected PNC endpoint (functional device). In case of storage drive, if *EraseOnDetach* parameter of drive is set to *"true"*, this drive will be Secure Erased before returning to the pool. Action can be performed when Composed Node is *Assembled* or *Failed.* It takes one argument:<br>○ Resource – link to PNC endpoint realized by PCI device function that needs to be detached. |

**Figure 2 ComposedNodeState changes during assembly process**

## 4.13.1 Operations

### 4.13.1.1 GET

Request:

```
GET /redfish/v1/Nodes/{nodeID}
Content-Type: application/json
```

Response:

```
{
```

```
    "@odata.context": "/redfish/v1/$metadata#Nodes/Members/$entity",
    "@odata.id": "/redfish/v1/Nodes/Node1",
    "@odata.type": "#ComposedNode.1.1.0.ComposedNode",
    "Id": "Node1",
    "Name": "Composed Node",
    "Description": "Node #1",
    "UUID": "00000000-0000-0000-0000-000000000000 - the same as Computer
System",
    "PowerState": "On",
    "Status": {
        "State": "Enabled",
        "Health": "OK",
        "HealthRollup": "OK"
    },
    "Processors": {
        "Count": 2,
        "Model": "Multi-Core Intel(R) Xeon(R) processor 7xxx Series",
        "Status": {
            "State": "Enabled",
            "Health": "OK"
        }
    },
    "Memory": {
        "TotalSystemMemoryGiB": 32,
        "Status": {
            "State": "Enabled",
            "Health": "OK"
        }
    },
    "ComposedNodeState": "Allocated",
    "Boot": {
        "BootSourceOverrideEnabled": "Disabled",
        "BootSourceOverrideTarget": "None",
        "BootSourceOverrideTarget@Redfish.AllowableValues": [
            "None",
            "Pxe",
            "Hdd",
            "RemoteDrive"
        ],
    "BootSourceOverrideMode": "Legacy",
    "BootSourceOverrideMode@Redfish.AllowableValues":  ["Legacy",
      "UEFI"]
    },
    "Oem": {},
    "Links": {
        "ComputerSystem": {
            "@odata.id": "/redfish/v1/Systems/System1"
        },
        "Processors": [
            {
                "@odata.id": "/redfish/v1/Systems/System1/Processors/CPU1"
            }
        ],
        "Memory": [
            {
```

```
                "@odata.id": "/redfish/v1/Systems/System1/Memory/Dimm1"
            }
        ],
        "EthernetInterfaces": [
            {
                "@odata.id":
"/redfish/v1/Systems/System1/EthernetInterfaces/LAN1"
            }
        ],
        "LocalDrives": [
            {
                "@odata.id": "/redfish/v1/Chassis/Blade1/Drives/1"
            }
        ],
        "RemoteDrives": [
            {
                "@odata.id": "/redfish/v1/Services/RSS1/Targets/target1"
            }
        ],
        "ManagedBy": [
            {
                "@odata.id": "/redfish/v1/Managers/PODM"
            }
        ],
        "Oem": {}
    },
    "Actions": {
        "#ComposedNode.Reset": {
            "target": "/redfish/v1/Nodes/Node1/Actions/ComposedNode.Reset",
            "ResetType@Redfish.AllowableValues": [
                "On",
                "ForceOff",
                "GracefulRestart",
                "ForceRestart",
                "Nmi",
                "ForceOn",
                "PushPowerButton",
                "GracefulShutdown"
            ]
        },
        "#ComposedNode.Assemble": {
            "target": "/redfish/v1/Nodes/Node1/Actions/ComposedNode.Assemble"
        },
    "#ComposedNode.AttachEndpoint": {
      "target":
"/redfish/v1/Nodes/Node1/Actions/ComposedNode.AttachEndpoint",
      "Resource@Redfish.AllowableValues": [

{"@odata.id":"/redfish/v1/Chassis/PCIeSwitchChassis/Drives/Disk.Bay.1"},

{"@odata.id":"/redfish/v1/Chassis/PCIeSwitchChassis/Drives/Disk.Bay.2"}
      ]
    },
    "#ComposedNode.DetachEndpoint": {
```

```
      "target":
"/redfish/v1/Nodes/Node1/Actions/ComposedNode.DetachEndpoint",
      "Resource@Redfish.AllowableValues": [

{"@odata.id":"/redfish/v1/Chassis/PCIeSwitchChassis/Drives/Disk.Bay.3"}
      ]
    }
    }
}
```

#### 4.13.1.2 PUT

Operation is not allowed on this resource.

#### 4.13.1.3 PATCH

The following properties can be updated by PATCH operation:

| Attribute | Type | Required | Description |
|---|---|---|---|
| Boot | Object | No | Boot override properties, details in Table 20. |

The following table describes "Boot" properties that can be patched:

**Table 20 Boot Override update properties**

| Attribute | Type | Required | Description |
|---|---|---|---|
| BootSourceOverrideEnabled | String | No | Describes the state of the Boot Source Override feature. Allowed values:<br>"Disabled" - The system will boot as normal<br>"Once" - On its next boot cycle, the system will boot (one time) to the Boot Source Override Target<br>"Continuous" - The system will boot to the target specified in the BootSourceOverrideTarget until this property is set to Disabled |
| BootSourceOverrideTarget | String | No | The current boot source to be used at next boot instead of the normal boot device, if BootSourceOverrideEnabled is true.<br>Supported values:<br>"None" - Boot from the normal boot device<br>"Pxe" - Boot from the Pre-Boot EXecution (PXE) environment<br>"Hdd" - Boot from a hard drive |
| BootSourceOverrideMode | String | No | The BIOS Boot Mode (either Legacy or UEFI) to be used when BootSourceOverrideTarget boot source is booted from:<br>"Legacy" - The system will boot in non-UEFI boot mode to the Boot Source Override Target<br>"UEFI" - The system will boot in UEFI boot mode to the Boot Source Override Target |

**Note**: Boot property is representing only override values, not current boot source configured on system. To ensure that correct boot source/mode is applied, it is recommended to send PATCH to boot property after node assembly, before powering it on.

```
PATCH /redfish/v1/Nodes/Node1
Content-Type: application/json
{
        "Boot": {
                "BootSourceOverrideEnabled": "Once",
```

```
                "BootSourceOverrideTarget": "Pxe",
                "BootSourceOverrideMode": "Legacy"
        }
}
```

Response:

```
HTTP/1.1 204 No Content
```

Or:

```
HTTP/1.1 200 OK
```

with full resource representation.

#### 4.13.1.4 POST

Reset Action

Request:

```
POST /redfish/v1/Nodes/1/Actions/ComposedNode.Reset
Content-Type: application/json
{
        "ResetType": "On"
}
```

Response:

```
HTTP/1.1 204 No Content
```

Assemble action

Request:

```
POST /redfish/v1/Nodes/1/Actions/ComposedNode.Assemble
```

Response:

```
HTTP/1.1 204 No Content
```

AttachEndpoint action (1) – attaching specific endpoint (PNC drive) to existing composed node.

Request:

```
POST /redfish/v1/Nodes/1/Actions/ComposedNode.AttachEndpoint
Content-Type: application/json
{
    "Resource": {
        "@odata.id":"/redfish/v1/Chassis/PCIeSwitchChassis/Drives/Disk.Bay.1"
    }
}
```

Response:

```
HTTP/1.1 204 No Content
```

AttachEndpoint action (2) – attaching endpoint meeting criteria provided in request body to existing composed node.

Request:

```
POST /redfish/v1/Nodes/1/Actions/ComposedNode.AttachEndpoint
Content-Type: application/json
{
```

```
    "CapacityGiB": 40
}
```

Response:

```
HTTP/1.1 204 No Content
```

DetachDrive action

Request:

```
POST /redfish/v1/Nodes/1/Actions/ComposedNode.DetachEndpoint
Content-Type: application/json
{
    "Resource": {
       "@odata.id":"/redfish/v1/Chassis/PCIeSwitchChassis/Drives/Disk.Bay.3"
    }
}
```

Response:

```
HTTP/1.1 204 No Content
```

#### 4.13.1.5 DELETE

Upon deletion (disassembly) of Composed Node, several actions are performed:

- Force off request is sent to Computer System.
- All VLANs (except for reserved ones – see *Reserved VLANs*) are removed from associated Ethernet switch ports associated with Computer System's Ethernet Interfaces.
- All PCIe* devices connected to Node via PCIe switch are detached
- All Drives attached via PCIe Switch with property EraseOnDetach set to *true* are secure erased.

DELETE action removes components reservation (deallocation) and puts them back in the resource pool.

Request:

```
DELETE /redfish/v1/Nodes/1
```

Response:

```
HTTP/1.1 204 No Content
```

## 4.14 PSME and Storage Services resources

PODM supports PSME and Storage Services resources. The table below describes which resources and their operations are included as a part of PODM API.

| Resource Name | Supported Operations | | | | |
|---|---|---|---|---|---|
| | GET | PATCH | POST | DELETE | Actions |
| Computer System | X | X | | | X |
| Processor | X | | | | |
| Memory | X | | | | |
| Storage | X | | | | |
| Drive | X | X | | | X |

| Resource Name | Supported Operations | | | | |
|---|---|---|---|---|---|
| | GET | PATCH | POST | DELETE | Actions |
| Network Interface | X | | | | |
| VLAN | X | | X | X | |
| Manager | X | | | | |
| Network Protocol | X | | | | |
| Ethernet Switch | X | | | | |
| Ethernet Switch Port | X | X | X | X | |
| Ethernet Switch ACL | | | | | |
| Ethernet Switch ACL rules | | | | | |
| Ethernet Switch Port static MACs | | | | | |
| Fabric | X | | | | |
| Zone | X | X | | | |
| PCIe Device | X | X | | | |
| Fabric Switch | X | | | | |
| PCIe Port | X | | | | |
| PCIe Function | X | | | | |
| Endpoint | X | | | | |
| Storage Service | X | | | | |
| Remote Target | X | X | | | |
| Logical Drive | X | | | | |
| Physical Drive | X | | | | |
| EventService | X | | | | |
| EventSubscription | X | | X | X | |
| NetworkInterface | X | | | | |
| NetworkDeviceFunction | X | X | | | |

See the PSME and Storage Services specifications for resource definitions.

## 4.15 Simple Storage collection

**Table 21 Simple storage collection attributes**

| Name | Simple storage | | |
|---|---|---|---|
| Type URI | /redfish/v1/Systems/System1/SimpleStorage | | |
| Attribute | Type | Required | Description |
| `Name` | String | Yes | Name of collection |
| `Members@odata.count` | Number | No | Collection members count |
| `Members` | Array | No | Contains the members of this collection. |

### 4.15.1 Operations

#### 4.15.1.1 GET

Request:

```
GET /redfish/v1/Systems/System1/SimpleStorage
```

```
Content-Type: application/json
```

Response:

```
{
    "@odata.context":
"/redfish/v1/$metadata#SimpleStorageCollection.SimpleStorageCollection",
    "@odata.id": "/redfish/v1/Systems/System1/SimpleStorage",
    "@odata.type": "#SimpleStorageCollection.SimpleStorageCollection",
    "Name": "Simple Storage Collection",
    "Members@odata.count": 1,
    "Members": [
        {
            "@odata.id": "/redfish/v1/Systems/System1/SimpleStorage/Storage1"
        }
    ]}
```

#### 4.15.1.2 PUT

Operation is not allowed on this resource.

#### 4.15.1.3 PATCH

Operation is not allowed on this resource.

#### 4.15.1.4 POST

Operation is not allowed on this resource.

#### 4.15.1.5 DELETE

Operation is not allowed on this resource.

## 4.16 Simple storage

Simple storage devices associated with this system.

**Table 22 Simple storage attributes**

| **Name** | Simple storage | |
|---|---|---|
| **Type URI** | /redfish/v1/Systems/System1/SimpleStorage/{storageID} | |
| **Attribute** | **Type** | **Description** |
| Id | String | Resource identifier |
| Name | String | Name of service root |
| Description | | Provides a description of this resource and is used for commonality in the schema definitions |
| UefiDevicePath | String | The UEFI device path used to access this storage controller. |
| Devices | Object | The storage devices associated with this resource |

| **Attribute** | **Type** | **Required** | **Description** |
|---|---|---|---|
| Name | String | No | The name of the resource |
| Manufacturer | String, null | No | The name of the manufacturer of this device |
| Model | String, null | No | The product model number of this device |
| Status | Object, null | No | See Section 5.1 for status of resource. |
| CapacityBytes | Number, null | No | The size of the storage device |
| Oem | Object | No | Oem defined object |

| | | |
|---|---|---|
| Status | Object | See Section 5.1 for status of resource. |

| | | |
|---|---|---|
| Oem | Object | Oem extension object |

### 4.16.1 Operations

#### 4.16.1.1 GET

Request:

```
GET /redfish/v1/Systems/System1/SimpleStorage/Storage1
Content-Type: applicaton/json
```

Response:

```
{
    "@odata.context": "/redfish/v1/$metadata#SimpleStorage.SimpleStorage",
    "@odata.id": "/redfish/v1/Systems/System1/SimpleStorage/Storage1",
    "@odata.type": "#SimpleStorage.v1_1_0.SimpleStorage",
    "Id": "Storage1",
    "Name": "Simple Storage Controller",
    "Description": "System SATA",
    "UefiDevicePath": "UEFI Device Path",
    "Status": {
        "State": "Enabled",
        "Health": "OK",
        "HealthRollup": "OK"
    },
    "Devices": [
        {
            "@odata.type": "#SimpleStorage.v1_1_0.Device",
            "Name": "Drive 1",
            "Manufacturer": "ACME",
            "Model": "Drive Model string",
           "CapacityBytes": 322122547200,
            "Status": {
                "State": "Enabled",
                "Health": "OK"
            }
        },
        {
            "@odata.type": "#SimpleStorage.v1_1_0.Device",
            "Name": "Drive 2",
            "Manufacturer": "SuperDuperSSD",
            "Model": "Drive Model string",
           "CapacityBytes": 68719476736,
            "Status": {
                "State": "Enabled",
                "Health": "OK"
            }
        },
        {
            "Name": "Drive 3",
            "Status": {
                "State": "Absent"
            }
        },
```

```
        {
            "Name": "Drive 4",
            "Status": {
                "State": "Absent"
            }
        }
    ]
}
```

#### 4.16.1.2 PUT

Operation is not allowed on this resource.

#### 4.16.1.3 PATCH

Operation is not allowed on this resource.

#### 4.16.1.4 POST

Operation is not allowed on this resource.

#### 4.16.1.5 DELETE

Operation is not allowed on this resource.

## 4.17 Power

Power metrics resource. It represents the properties for Power Consumption and Power Limiting.

Detailed info about this resource properties can be obtained from metadata file: *Power.xml*

### 4.17.1 Operations

#### 4.17.1.1 GET

Request:

```
GET /redfish/v1/Chassis/Rack1/Power
Content-Type: applicaton/json
```

Response:

```
{
    "@odata.context": "/redfish/v1/$metadata#Power.Power",
    "@odata.id": "/redfish/v1/Chassis/Rack1/Power",
    "@odata.type": "#Power.v1_1_0.Power",
    "Id": "Power",
    "Name": "Power",
    "Description": "PowerSubsystem",
    "PowerControl": [
        {
            "@odata.id": "/redfish/v1/Chassis/Rack1/Power#/PowerControl/0",
            "MemberId": "0",
               "Name": "System Power Control",
            "PowerConsumedWatts": 8000,
            "PowerRequestedWatts": 8500,
            "PowerAvailableWatts": 8500,
            "PowerCapacityWatts": 10000,
            "PowerAllocatedWatts": 8500,
            "PowerMetrics": {
```

```
                "IntervalInMin": 30,
                "MinConsumedWatts": 7500,
                "MaxConsumedWatts": 8200,
                "AverageConsumedWatts": 8000
            },
            "PowerLimit": {
                "LimitInWatts": 9000,
                "LimitException": "LogEventOnly",
                "CorrectionInMs": 42
               },
            "RelatedItem": [
                {"@odata.id": "/redfish/v1/Chassis/Drawer1"},
                {"@odata.id": "/redfish/v1/Systems/System1"}
            ],
             "Status": {
                   "State": "Enabled",
                   "Health": "OK",
                "HealthRollup": "OK"
               },
               "Oem": {}
        }
    ],
    "Voltages": [
        {
            "@odata.id": "/redfish/v1/Chassis/Rack1/Power#/Voltages/0",
            "MemberId": "0",
            "Name": "VRM1 Voltage",
            "SensorNumber": 11,
            "Status": {
                "State": "Enabled",
                "Health": "OK"
            },
            "ReadingVolts": 12,
            "UpperThresholdNonCritical": 12.5,
            "UpperThresholdCritical": 13,
            "UpperThresholdFatal": 15,
            "LowerThresholdNonCritical": 11.5,
            "LowerThresholdCritical": 11,
            "LowerThresholdFatal": 10,
            "MinReadingRange": 0,
            "MaxReadingRange": 20,
            "PhysicalContext": "VoltageRegulator",
            "RelatedItem": [
                {"@odata.id": "/redfish/v1/Systems/System1" }
            ]
        }
    ],
    "PowerSupplies": [
        {
            "@odata.id": "/redfish/v1/Chassis/Rack1/Power#/PowerSupplies/0",
            "MemberId": "0",
            "Name": "Power Supply Bay 1",
            "Status": {
                "State": "Enabled",
                "Health": "Warning"
```

```
            },
            "Oem": {},
            "PowerSupplyType": "DC",
            "LineInputVoltageType": "DCNeg48V",
            "LineInputVoltage": -48,
            "PowerCapacityWatts": 400,
            "LastPowerOutputWatts": 192,
            "Model": "499253-B21",
            "Manufacturer": "ManufacturerName",
            "FirmwareVersion": "1.00",
            "SerialNumber": "1z0000001",
            "PartNumber": "1z0000001A3a",
            "SparePartNumber": "0000001A3a",
               "InputRanges": [
               {
                   "InputType": "DC",
                   "MinimumVoltage": -47,
                   "MaximumVoltage": -49,
                   "OutputWattage": 400,
                   "MinimumFrequencyHz": 50,
                   "MaximumFrequencyHz": 60,
                   "Oem": {}
              }
            ],
            "IndicatorLED": "Off",
            "RelatedItem": [
              { "@odata.id": "/redfish/v1/Chassis/Rack1" }
            ],
            "Redundancy": [
              { "@odata.id": "/redfish/v1/Chassis/1/Power#/Redundancy/0" }
            ]
        }
    ],
    "Redundancy": [
        {
            "@odata.id": "/redfish/v1/Chassis/Rack1/Power#/Redundancy/0",
            "MemberId": "0",
            "Name": "PowerSupply Redundancy Group 1",
            "Mode": "Failover",
            "MaxNumSupported": 2,
            "MinNumNeeded": 1,
            "RedundancySet": [
                { "@odata.id": "/redfish/v1/Chassis/1/Power#/PowerSupplies/0"
}
            ],
            "Status": {
                "State": "Offline",
                "Health": "OK"
            }
        }
    ],
    "Oem": {}
}
```

#### 4.17.1.2 PUT

Operation is not allowed on this resource.

#### 4.17.1.3 PATCH

Operation is not allowed on this resource.

#### 4.17.1.4 POST

Operation is not allowed on this resource.

#### 4.17.1.5 DELETE

Operation is not allowed on this resource.

## 4.18 Thermal

Thermal metrics resource. It represents the properties for Temperature and Cooling.

Detailed info about this resource properties can be obtained from metadata file: *Thermal.xml*

### 4.18.1 Operations

#### 4.18.1.1 GET

Request:

```
GET /redfish/v1/Chassis/Rack1/Thermal
Content-Type: applicaton/json
```

Response:


```
{
    "@odata.context": "/redfish/v1/$metadata#Thermal.Thermal",
    "@odata.id": "/redfish/v1/Chassis/Rack1/Thermal",
    "@odata.type": "#Thermal.v1_1_0.Thermal",
    "Id": "Thermal",
    "Name": "Thermal",
    "Description": "Thermal Subsystem",
    "Temperatures": [
        {
            "@odata.id": "/redfish/v1/Chassis/Rack1/Thermal#/Temperatures/0",
            "MemberId": "0",
            "Name": "Drawer inlet Temp",
            "SensorNumber": 42,
            "Status": {
                "State": "Enabled",
                "Health": "OK"
            },
            "ReadingCelsius": 21,
            "UpperThresholdNonCritical": 42,
            "UpperThresholdCritical": 42,
            "UpperThresholdFatal": 42,
            "LowerThresholdNonCritical": 42,
            "LowerThresholdCritical": 5,
            "LowerThresholdFatal": 42,
            "MinReadingRange": 0,
```

```
            "MaxReadingRange": 200,
            "PhysicalContext": "Intake",
            "RelatedItem": [
                {"@odata.id": "/redfish/v1/Chassis/Drawer1" }
            ]
        }
    ],
    "Fans": [
        {
            "@odata.id": "/redfish/v1/Chassis/Rack1/Thermal#/Fans/0",
            "MemberId": "0",
            "Name": "BaseBoard System Fan",
            "PhysicalContext": "Backplane",
            "Status": {
                "State": "Enabled",
                "Health": "OK"
            },
            "Reading": 2100,
            "ReadingUnits": "RPM",
            "UpperThresholdNonCritical": 42,
            "UpperThresholdCritical": 4200,
            "UpperThresholdFatal": 42,
            "LowerThresholdNonCritical": 42,
            "LowerThresholdCritical": 5,
            "LowerThresholdFatal": 42,
            "MinReadingRange": 0,
            "MaxReadingRange": 5000,
            "Redundancy" : [
                {"@odata.id":
"/redfish/v1/Chassis/Rack1/Thermal#/Redundancy/0"}
            ],
            "RelatedItem" : [
                {"@odata.id": "/redfish/v1/Chassis/Rack1" }
            ]
        }
    ],
    "Redundancy": [
        {
            "@odata.id": "/redfish/v1/Chassis/Rack1/Thermal#/Redundancy/0",
            "MemberId": "0",
            "Name": "BaseBoard System Fans",
            "RedundancyEnabled": false,
            "RedundancySet": [
                { "@odata.id": "/redfish/v1/Chassis/1/Thermal#/Fans/0" }
            ],
            "Mode": "N+m",
            "Status": {
                "State": "Disabled",
                "Health": "OK"
            },
            "MinNumNeeded": 1,
            "MaxNumSupported": 2
        }
    ]
}
```

#### 4.18.1.2 PUT

Operation is not allowed on this resource.

#### 4.18.1.3 PATCH

Operation is not allowed on this resource.

#### 4.18.1.4 POST

Operation is not allowed on this resource.

#### 4.18.1.5 DELETE

Operation is not allowed on this resource.

§

# 5 Common Property Description

## 5.1 Status

| Attribute | Type | Nullable | Description |
|---|---|---|---|
| State | String | Yes | This indicates the known state of the resource, such as if it is enabled. Allowed values: See Section 5.2. |
| Health | String | Yes | This represents the health state of this resource in the absence of its dependent resources. Allowed values: See Section 5.3. |
| HealthRollup | String | Yes | This represents the overall health state from the view of this resource. Allowed values: See Section 5.3. |

## 5.2 Status -> State

- Enabled: This function or resource has been enabled
- Disabled: This function or resource has been disabled
- StandbyOffline: This function or resource is enabled, but awaiting an external action to activate it
- StandbySpare: This function or resource is part of a redundancy set and is awaiting a failover or other external action to activate it.
- InTest: This function or resource is undergoing testing
- Starting: This function or resource is starting
- Absent: This function or resource is not installed
- UnavailableOffline: This function or resource is present but cannot be used
- Deferring: The element will not process any commands but will queue new requests.
- Quiesced: The element is enabled but only processes a restricted set of commands.
- Updating: The element is updating and may be unavailable or degraded.

## 5.3 Status -> Health

- OK: Normal
- Warning: A condition exists that requires attention
- Critical: A critical condition exists that requires immediate attention

## 5.4 ComputerSystem.Reset

- On: Turn the system on
- ForceOff: Turn the system off immediately (nongraceful) shutdown
- GracefulRestart: Perform a graceful system shutdown followed by a restart of the system
- ForceRestart: Perform an immediate (non-graceful) shutdown, followed by a restart of the system
- Nmi: Generate a nonmaskable interrupt to cause an immediate system halt
- ForceOn: Turn the system on immediately
- PushPowerButton: Simulate the pressing of the physical power button on this system
- GracefulShutdown: Perform a graceful system shutdown and power off

## 5.5 BootSourceOverrideTarget/Supported

- None: Boot from the normal boot device
- Pxe: Boot from the preboot execution (PXE) environment

- Floppy: Boot from a floppy disk drive
- Cd: Boot from a CD/DVD disc
- Usb: Boot from a USB device as specified by the system BIOS
- Hdd: Boot from a hard drive
- BiosSetup - Boot to the BIOS Setup Utility
- Utilities: Boot the manufacturer's Utilities programs
- Diags: Boot the manufacturer's Diagnostics program
- UefiShell: Boot to the UEFI Shell
- UefiTarget: Boot to the UEFI Device specified in the UefiTargetBootSourceOverride property
- SDCard: Boot from an SD Card
- UefiHttp: Boot from a UEFI HTTP network location
- RemoteDrive: Boot from a remote drive (e.g. iSCSI)

§

# 6 Appendix

## 6.1 Creating new Composed Node - explanation

### 6.1.1 Creating Composed Node using JSON template

To create a Composed Node using Pod Manager REST API, it is necessary to create a JSON template describing requested resources. It needs to be supplied to Pod Manager by performing an HTTP POST request on Composed Node Collection Action URI located at "/redfish/v1/Nodes/Actions/Allocate" on the Pod Manager service.

The JSON template may contain various details of resources to be used in Composed Node. All JSON template elements are optional, but each requirement should be coherent itself. It is possible to supply Pod Manager with a JSON template containing no specific requirements (e.g. **{}** – a pair of empty curly braces in the HTTP request body) thus allowing Pod Manager to propose a Composed Node containing resources chosen arbitrarily by Pod Manager.

### 6.1.2 Specifying requirements for a Composed Node

The JSON template contains requirements for a single Composed Node. Basic customization covers setting a "Name" and "Description" of such System (both being of type *String*). As "Name" parameter is required by Redfish for all resources, if it's not supported Pod Manager will use the default name. The example below allocates a single Composed Node with requested name and description:

```
{
        "Name": "Customized Composed Node name",
        "Description": "Description of a customized Composed Node"
}
```

The JSON template may contain requirements for: Processors, Memory, Remote Drives, Local Drives and Ethernet Interfaces. To specify requirements for those resources, a proper section must appear in the JSON template.

### 6.1.3 General assumptions for allocation

Requirements are treated as a minimal required value, so the resulting Composed Node may have better parameters than requested. Composed Node customization and resource customization sections described below can be used jointly.

Each resource type description has an associated table which contains details about specific requirements. **Key** is the JSON object field. **JSON type** contains data type as defined by json.org. **Allowed values** contains additional restrictions to JSON type or hints (e.g. for enumerations or *boolean* values). **Nullable** indicates if *null* value can be passed for a specified key. **Notes, limitations** provide additional hints about specific requirements.

#### 6.1.3.1 Location requirements

Processor, Memory, Local Drive and Ethernet Interface sections may contain Resource and Chassis objects. Resource must contain Pod Manager URI (presented as "@odata.id") of discovered resource (Processor's URI in Processor section, URI to Memory resource in Memory section and so on). Chassis must contain Pod Manager URI of discovered Chassis in which applicable resources will be looked for.

### 6.1.4 Specifying Processor requirements

The JSON template may contain requirements for multiple Processors. The example below specifies requirements for a single Processor to be used in Composed Node.

```
{
        "Processors": [{
                "Model": "Multi-Core Intel(R) Xeon(R) processor 7xxx Series",
                "TotalCores": 2,
                "AchievableSpeedMHz": 3700,
                "InstructionSet": "x86-64",
                "Oem": {
                        "Brand": "X7",
                        "Capabilities": [
                                "sse"
                        ],
                },
                "Resource": {
                        "@odata.id": "/redfish/v1/Systems/1/Processors/1"
                },
                "Chassis": {
                        "@odata.id": "/redfish/v1/Chassis/1"
                }
        }]
}
```

| Attribute | Type | Allowed values | Nullable | Description |
|---|---|---|---|---|
| Model | String | | Yes | String representing Processor model. |
| TotalCores | Number | | Yes | Positive integer value expected |
| AchievableSpeedMHz | Number | | Yes | Positive integer value expected |
| InstructionSet | String | "x86", "x86-64", "IA-64", "ARM-A32", "ARM-A64", "MIPS32", "MIPS64", "OEM" | Yes | One of allowed, enumerated values |
| Oem | Object | | Yes | |
| Oem → Brand | String | "E3", "E5", "E7", "X3", "X5", "X7", "I3", "I5", "I7", "Unknown" | Yes | One of allowed, enumerated values |
| Oem->Capabilities | Array | CPU capabilities string | Yes | List of processor capabilities (like "sse") |
| Resource | Object | Exact location of a single Processor. | Yes | See Location requirements section |
| Chassis | Object | Exact location of a single chassis. | Yes | See Location requirements section |

The template can also provide the requirement for total processor cores available in composed node:

```
        "TotalSystemCoreCount": 32
```

Allocation assumptions:

- Which Processors will meet supplied requirements?
  - located on the same computer system as other resources
  - with exact match on Model
  - with exact match on Brand
  - with at least TotalCores
  - with at least AchievableSpeedMHz
  - with exact match on InstructionSet
  - with superset of processor capabilities specified Capabilities array

If a computer system contains processors whose cores sum up to number equal or greater than TotalSystemCoreCount, it will meet the requirements.

### 6.1.5 Specifying Memory requirements

The JSON template may contain requirements for multiple Memory Modules. The example below specifies requirements for a single Memory Module to be used in Composed Node.

```
{
        "Memory": [{
                "CapacityMiB": 16000,
                "MemoryDeviceType": "DDR3",
                "SpeedMHz": 1600,
                "Manufacturer": "Intel",
                "DataWidthBits": 64,
                "Resource": {
                        "@odata.id": "/redfish/v1/Systems/1/Memory/1"
                },
                "Chassis": {
                        "@odata.id": "/redfish/v1/Chassis/1"
                }
        }]
}
```

| Attribute | Type | Allowed values | Nullable | Notes, limitations |
|---|---|---|---|---|
| CapacityMiB | Number | | Yes | Positive value expected |
| MemoryDeviceType | String | "DDR", "DDR2", "DDR3", "DDR4", "DDR4_SDRAM", "DDR4E_SDRAM", "LPDDR4_SDRAM", "DDR3_SDRAM", "LPDDR3_SDRAM", "DDR2_SDRAM", "DDR2_SDRAM_FB_DIMM", "DDR2_SDRAM_FB_DIMM_PROBE", "DDR_SGRAM", "DDR_SDRAM", "ROM", "SDRAM", "EDO", "FastPageMode", "PipelinedNibble" | Yes | One of allowed, enumerated values |
| SpeedMHz | Number | | Yes | Positive integer value expected |
| Manufacturer | String | | Yes | String representing Memory Module manufacturer name |
| DataWidthBits | Number | | Yes | Positive integer value expected |
| Resource | Object | Exact location of a single Memory Module. | Yes | See Location requirements section |
| Chassis | Object | Exact location of a single chassis. | Yes | See Location requirements section |

The template can also provide requirements for total memory available in composed node, without dividing it into memory modules:

```
        "TotalSystemMemoryMiB": 32000
```

Allocation assumptions:

- Which Memory Modules will meet supplied requirements?

  - With at least CapacityMiB
  - located on the same computer system as other resources
  - with exact match on MemoryDeviceType
  - with at least SpeedMHz
  - with exact match on Manufacturer
  - with at least DataWidthBits
- If a computer system contains Memory Modules of total size at least TotalSystemMemory, it will meet the requirements.

### 6.1.6 Specifying Remote Drive requirements

The JSON template may contain requirements for multiple Remote Drives, but currently only one set of requirements is supported. The example below specifies requirements for a single Remote Drive to be used in Composed Node.

```
{
        "RemoteDrives": [{
                "CapacityGiB": 80,
                "iSCSIAddress": "iqn.oem.com:fedora21",
                "Master": {
                        "Type": "Snapshot",
                        "Resource": {
                                "@odata.id":
"/redfish/v1/Services/1/LogicalDrives/1"
                        }
                }
        }]
}
```

| Attribute | Type | Allowed values | Nullable | Description |
|---|---|---|---|---|
| CapacityGiB | Number | | Yes | Positive value expected, required if Master Drive supplied. Should be at least the size of Logical Drive used as Master Drive. |
| iSCSIAddress | String | | No | Required. Defines TargetIQN of RemoteTarget. When no Master Drive is supplied – it defines IQN of an existing target. Otherwise defines IQN to be set for new Remote Target (should be unique in Pod Manager). |
| Master | Object | | Yes | |
| Master →Type | String | “Snapshot”, “Clone” | No | One of allowed, enumerated values. Required if Master Drive is supplied. |
| Master → Address | Object | | No | Pod Manager URI of discovered Logical Volume. Required if Master Drive is supplied. |

#### 6.1.6.1 Using existing Remote Drive

To use an existing Drive it is necessary to:

- Set the iSCSIAddress to TargetIQN of existing target,
- Do not provide Master, or set it to null

```
{
```

```
        "RemoteDrives": [{
                "iSCSIAddress": "iqn.oem.com:fedora21"
        }]
}
```

### 6.1.6.2 Using a Master Drive for fresh Remote Drive creation

To use a fresh Drive created from Master Drive it is necessary to:

- Set CapacityGiB to define capacity of the new Remote Drive that is at least of Master Drive's size,
- Set Address to IQN that is unique in Pod Manager
- Set Master → Type to "Snapshot" or "Clone"
- Set Master → Resource to valid Pod Manager URI of Logical Drive to be used as source Drive

```
{
        "RemoteDrives": [{
                "CapacityGiB": 80,
                "iSCSIAddress": "iqn.oem.com:fedora21",
                "Master": {
                        "Type": "Snapshot",
                        "Resource": {
                                "@odata.id":
"/redfish/v1/Services/1/LogicalDrives/1"
                        }
                }
        }]
}
```

## 6.1.7 Specifying Local Drive requirements

The JSON template may contain requirements for multiple Local Drives (represented by Drive resource). The example below specifies requirements for a single Local Drive to be used in Composed Node.

```
{
        "LocalDrives": [{
                "CapacityGiB": 100,
                "Type": "HDD",
                "MinRPM": 5400,
                "SerialNumber": "12345678",
                "Interface": "SATA",
                "Resource": {
                        "@odata.id": "redfish/v1/Chassis/Blade1/Drives/Disk1"
                },
                "Chassis": {
                        "@odata.id": "/redfish/v1/Chassis/Blade1"
                },
                "FabricSwitch": false           }]
}
```

| Attribute | Type | Allowed values | Nullable | Description |
|---|---|---|---|---|
| CapacityGiB | Number | | Yes | Positive value expected |
| Type | String | "HDD", | Yes | One of allowed, enumerated values |

| | | | | |
|---|---|---|---|---|
| | | "SSD" | | |
| MinRPM | Number | | Yes | Positive integer value expected |
| SerialNumber | String | | Yes | |
| Interface | String | "SAS", "SATA", "NVMe" | Yes | One of allowed, enumerated values |
| Resource | Object | Exact location of a single Device. | Yes | See Location requirements section |
| Chassis | Object | Exact location of a single Chassis. | Yes | See Location requirements section |
| FabricSwitch | Boolean | True, false | Yes | Determines if drive should be connected using fabric switch (PNC) or directly attached to computer system. |

Allocation assumptions:

- Which Local Drives will meet supplied requirements?
  - located on the same computer system as other resources
  - with at least CapacityGiB
  - with exact match on Type
  - with at least MinRPM
  - with exact SerialNumber
  - with exact Interface

#### 6.1.7.1 Pooled NVMe Cotroller (PNC) drives

If PNC is available in Pod Manager, and there is no system fulfilling Local Drive requirements, PNC drives will be attached to Composed Node from the pool of available PNC drives.

Please note that PNC drives that were detached from Composed Node resource (via Detach action or via DELETE operation on node) and have the property "EraseOnDetach" set to *false* (or will be set to *null*), will not be erased. Their property "DriveErased" will not be changed to *true* and because of that they will not be available in the pool of PNC drives ready for node composition. Those drives need to be selected via the Resource property, or erased by action SecureErase on drive resource.

Drives with the property "DriveErased" set to *true* or *null* are available for composition without the need to specify their URI in the Resource property.

Example request that will allocate the node with PNC drive:

```
{
        "LocalDrives": [{
                "CapacityGiB": 100,
                "Type": "SSD",
                "Interface": "NVMe",
                "Chassis": {
                        "@odata.id": "/redfish/v1/Chassis/PCIeSwitchChassis"
                }
        }]
}
```

After node allocation and assembly (in "Assembled" and "Failed" ComposedNodeState), the user is able to attach and remove PNC devices (drives) using AttachEndpoint and DetachEndpoint actions:

Example attaching drive to existing node:

```
POST /redfish/v1/Nodes/1/Actions/ComposedNode.AttachEndpoint
Content-Type: application/json
{
```

```
    “Resource”: {
       "@odata.id":"/redfish/v1/Chassis/PCIeSwitchChassis/Drives/Disk.Bay.2"
    }
}
```

Eample removing drive from existing node:

```
POST /redfish/v1/Nodes/1/Actions/ComposedNode.DetachEndpoint
Content-Type: application/json
{
    “Resource”: {
        "@odata.id":"/redfish/v1/Chassis/PCIeSwitchChassis/Drives/Disk.Bay.3"
    }
}
```

### 6.1.8 Specifying Ethernet interface requirements

The JSON template may contain requirements for multiple Ethernet Interfaces. The example below specifies requirements for a single Ethernet Interface to be used in Composed Node.

```
{
       "EthernetInterfaces": [{
               "SpeedMbps": 1000,
               "PrimaryVLAN": 100,
               "VLANs": [{
                      "VLANId": 100,
                      "Tagged": false
               }],
               "Resource": {
                      "@odata.id":
"/redfish/v1/Systems/1/EthernetInterfaces/1"
               },
               "Chassis": {
                      "@odata.id": "/redfish/v1/Chassis/1"
               }
       }]
}
```

| Attribute | Type | Allowed values | Nullable | Description |
|---|---|---|---|---|
| SpeedMbps | Number | | Yes | Positive integer value expected |
| PrimaryVLAN | Number | | Yes | Positive integer value expected |
| VLANs | Array[Object] | | Yes | Null value will be interpreted as absence of this key.<br>Empty array [ ] will clear all existing vlans, excluding Reserved VLANs. |
| VLANs → VLANId | Number | | No | Positive integer value expected |
| VLANs → Tagged | Boolean | true, false | No | Boolean value |
| Resource | Object | Exact location of a single Ethernet Interface. | Yes | See Location requirements section |
| Chassis | Object | Exact location of a single Chassis. | Yes | See Location requirements section |

Allocation assumptions:

- Which Ethernet Interfaces will meet supplied requirements?
  - located on the same Computer System as other resources

- with at least SpeedMbps
- ones that are connected with SwitchPorts (when VLANs section is provided)

#### 6.1.8.1 Reserved VLANs

There is a possibility to restrict usage of some VLANs by changing the configuration file located in /etc/pod-manager/allocation.json.

Example file looks like:

```
{
  "ReservedVlanIds": [1, 170, 4088, 4091, 4094]
}
```

Where 1, 170, 4088, 4091, 4094 are VLANs that are reserved. Reserved VLANs have the following implications:

- Allocation JSON cannot contain such VLANs and such requests result in an error
- Reserved VLANs are not deleted during allocation
- Reserved VLANs are not deleted during disassembly

### 6.1.9 Allocation algorithm

Node composition starts with an HTTP POST request of JSON template on "/redfish/v1/Nodes/Actions/Allocate" Composed Node Collection Action URI on Pod Manager service. If the JSON template is well-formed and contains a supported set of requirements, the allocation process starts. Four major scenarios are currently supported:

- Allocating resources for Composed Node to be booted from Local Drive.
- Allocating resources for Composed Node to be booted from existing Remote Drive.
- Allocating resources for Composed Node to be booted from Remote Drive that need to be created.
- Allocating resources for Composed Node with VLAN requirements specified. This scenario is used with one of the three scenarios above.

The allocation process is preceded by a general verification of JSON template that checks if requested node can be realized by available resources and consists of:

- Selecting and allocating a Computer System that contains resources matching template requirements for Processors, Memory, Local Drives and Ethernet Interfaces.
- Selecting or creating a Remote Drive to be used with a previously selected Computer System and allocating it.

#### 6.1.9.1 Detailed process of selecting and allocating a computer system for a composed node

- Find all Computer Systems that are not yet allocated (not used by any other allocated Composed Node) with Status Enabled and Health OK.
- Filter Computer Systems by specified Resource and Chassis (if supplied in template)
- Filter Computer Systems by Processors: return all Computer Systems that contain at least the requested quantity of Processors that meet requirements (if supplied in template):
  - Exactly matching requested model,
  - Exactly matching requested brand,
  - With at least the requested number of cores,
  - With at least the requested frequency,
  - Exactly matching the requested instruction set.

- Filter Computer Systems by Memory: return all Computer Systems with at least the total requested size of memory located on Memory Modules where each one meets the requirements below (if supplied in template):
  - Memory of the exactly requested DIMM device type
  - With at least the requested speed MHz
  - With the exact requested manufacturer
  - With at least the requested data width bits
- Filter Computer Systems by Local Drives: return all Computer Systems that contain for each requested Drive one distinct Device meeting requirements (if supplied in template):
  - With at least the requested capacity specified
  - Exactly matching the requested Drive type
  - With at least the requested min RPM
  - With the exact requested serial number
  - With the exact Interface
- Filter Computer Systems by Ethernet Interfaces: return all Computer Systems that contain for each requested Ethernet Interface one distinct Ethernet Interface meeting requirements (if supplied in template):
  - With at least the requested speed.
  - If the VLANs section is provided, then Computer Systems with Ethernet Interfaces which are not connected with EthernetSwitchPorts are filtered out (as described below)
- A first Computer System from the resulting filtered collection is then allocated to be used in Composed Node.

#### 6.1.9.2 Connection between computer system's EthernetInterface and EthernetSwitchPort

In order to enable particular VLAN usage on Composed Node, there is a need to map the Ethernet Switch Port and Computer System's Ethernet interface. This mapping is done using a MAC address as an identifier. Fields used for this mapping are:

- NeighborMAC on EthernetSwitchPort resource
- MacAddress on EthernetInterface resource

If those two properties contain the same value, Computer System's Ethernet Interface and Ethernet Switch Port are treated as connected. Only Computer Systems with Ethernet Interfaces which are connected to Ethernet Switch Ports could be used in allocation with the specified VLANs requirement.

#### 6.1.9.3 Detailed process of selecting remote drives

- Determine what type of Remote Drive is requested
- When requesting an existing Remote Drive:
  - Find all Targets that are not yet allocated (not used by any allocated Composed Node)
  - Find first Target that exactly matches requested IQN and allocate it to be used in Composed Node
- When requesting a new Remote Drive:
  - Check if Target does not exist with requested IQN to be set for newly created target
  - Check if Logical Drive requested as Master Drive exists on Storage Service handled by Pod Manager, and select this Storage Service to handle new Target creation
  - Find all Logical Volume Groups meeting requirements:
    - Located on selected Storage Service
    - Having free space of at least requested capacity for a new Remote Drive
  - A first Logical Volume Group from the resulting filtered collection is selected as a placement for a new Logical Volume, which will be exposed as a new Target (Remote Drive)

- A new Logical Volume is created on a selected Logical Volume Group (as a snapshot or as a clone)
- A new Target is created on top of a newly created Logical Volume.
- Newly created Target is allocated to be used in Composed Node.

#### 6.1.9.4 Post-allocation scenarios

A Composed Node is created as a new REST resource at /redfish/v1/Nodes/{NodeId} when a proper Computer System is found and is successfully allocated. State of Composed Node is set to “Allocated”. An “Allocated” Composed Node is a Pod Manager proposition that can be either accepted or rejected.

- If accepted, the user has to send an HTTP POST request on ComposedNode.Assemble action of the proposed Composed Node to assemble it:
  - If no Remote Drive was requested, a Composed Node’s state is set to “Assembled”.
  - When a Remote Drive is requested, Composed Node remains “Assembling” until Target creation finishes. When Target is successfully assembled to be used with the Composed Node, node’s state is set to “Assembled”
  - Assembly process does not end with sending power on request, so it’s necessary to perform a ComposedNode.Reset action to power on a Composed Node after assembly.
- If rejected, the user can continue sending HTTP POST requests of JSON template on /redfish/v1/Nodes/Actions/Allocate to create more proposals to pick from. When finding the right pick, it is recommended to send an HTTP DELETE on all rejected proposals of Composed Nodes to free the resources allocated by them.

§