



Intel® Rack Scale Design Pod Manager

API Specification

Software Version 1.2

September 2016

Revision 005



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 © 2016 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	11
3.1	API error response	11
3.1.1	Example error JSON object	11
3.2	API error codes	12
3.2.1	General error codes	12
3.2.2	Request error codes	12
3.2.3	Node assembly error codes	12
3.2.4	Node action error codes	12
4	PODM REST API Definition	14
4.1	Odata support	14
4.2	Protocol version	14
4.2.1	Operations	14
4.3	Intel® Rack Scale Design POD manager service root	14
4.3.1	Operations	15
4.4	Chassis collection	17
4.4.1	Operations	17
4.5	Chassis	18
4.5.1	Operations	20
4.6	PowerZone collection	21
4.6.1	Operations	21
4.7	PowerZone	22
4.7.1	Operations	24
4.8	ThermalZone collection	25
4.8.1	Operations	26
4.9	ThermalZone	26
4.9.1	Operations	28
4.10	Storage service collection	30
4.10.1	Operations	30
4.11	Composed node collection	31
4.11.1	Operation	35
4.12	Composed node	37
4.12.1	Operations	40
4.13	PSME resources	44
4.14	Storage Services resources	44
4.15	Simple Storage collection	44
4.15.1	Operations	45
4.16	Simple storage	45
4.16.1	Operations	46



5	Common Property Description	48
5.1	Status.....	48
5.2	Status -> State.....	48
5.3	Status -> Health.....	48
5.4	ComputerSystem.Reset.....	48
5.5	BootSourceOverrideTarget/Supported.....	48
6	Appendix	50
6.1	Creating new Composed Node - explanation.....	50
6.1.1	Creating Composed Node using JSON template.....	50
6.1.2	Specifying requirements for a Composed Node	50
6.1.3	General assumptions for allocation	50
6.1.4	Specifying processor requirements	50
6.1.5	Specifying memory requirements.....	51
6.1.6	Specifying Remote Drive requirements.....	52
6.1.7	Specifying Local Drive requirements.....	54
6.1.8	Specifying Ethernet interface requirements	55
6.1.9	Allocation algorithm	56

Figures

Figure 1	PODM REST API hierarchy	9
Figure 2	ComposedNodeState changes during assembly process	40

Tables

Table 1	Terminology	7
Table 2	Reference documents	7
Table 3	Resources and URI.....	9
Table 4	API error response attributes	11
Table 5	General error codes	12
Table 6	Request error codes.....	12
Table 7	Node assembly error codes	12
Table 8	Node disassembly error codes	12
Table 9	Intel® Rack Scale Design POD manager service Root attributes	15
Table 10	Chassis collection attributes.....	17
Table 11	Chassis attributes.....	18
Table 12	PowerZone collection attributes	21
Table 13	PowerZone attributes	22
Table 14	PowerSupply attributes	23
Table 15	ThermalZone collection attributes	25
Table 16	ThermalZone attributes.....	26
Table 17	Fan attributes	27
Table 18	Temperature attributes	28
Table 19	Storage service collection attributes	30
Table 20	Composed node collection attributes.....	31
Table 21	Composed node Allocation action attributes.....	32
Table 22	Remote master target properties	35
Table 23	Composed node attributes.....	37
Table 24	Simple storage collection attributes.....	44



Table 25	Simple storage attributes.....	45
----------	--------------------------------	----



Re vision History

Revision	Description	Date
0.8	External review comments addressed	February 18, 2016
0.61	Review comments incorporated	January 22, 2016
0.5	Ready for review	January 8, 2016
0.1	First internal draft	December 18, 2015

§



1 Introduction

1.1 Scope

This document contains information about the Intel® Rack Scale Design Pod Manager RESTAPI, which was designed and implemented for the Intel® Rack Scale Design Software v1.2 release for the Bulldog Creek SDV.

1.2 Intended audience

The intended audience for this document is designers and engineers working with the Intel® Rack Scale Design Software 1.2 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
SDV	Software Development Vehicle
URI	Uniform resource identifier
UUID	Universally Unique Identifier
XML	Extensible Markup Language

1.4 References

Table 2 Reference documents

Doc ID	Title	Location
332868	Intel® Rack Scale Design GAMI API Specification	http://intel.com/intelRSD
332869	Intel® Rack Scale Design Pod Manager REST API Specification	http://intel.com/intelRSD
332870	Intel® Rack Scale Design Pod Manager Release Notes	http://intel.com/intelRSD
332871	Intel® Rack Scale Design Pod Manager User Guide	http://intel.com/intelRSD
332873	Intel® Rack Scale Design PSME REST API Specification	http://intel.com/intelRSD
332872	Intel® Rack Scale Design PSME Release Notes	http://intel.com/intelRSD
332874	Intel® Rack Scale Design PSME User Guide	http://intel.com/intelRSD
332877	Intel® Rack Scale Design RMM REST API Specification	http://intel.com/intelRSD
332876	Intel® Rack Scale Design RMM Release Notes	http://intel.com/intelRSD
332875	Intel® Rack Scale Design RMM User Guide	http://intel.com/intelRSD
332878	Intel® Rack Scale Design Storage Services API Specification	http://intel.com/intelRSD



Doc ID	Title	Location
332936	Intel® Rack Scale Design BIOS/BMC Tech Guide	http://intel.com/intelRSD
332937	Intel® Rack Scale Design Architectural Requirements Specification	http://intel.com/intelRSD
334611	Intel® Rack Scale Design Getting Started Guide	http://intel.com/intelRSD
n/a	Scalable Platforms Management API	http://dmtof.org/standards/redfish

§

2 PODM API

2.1 PODM API structure and relations

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

2.1.1 PODM API physical resource hierarchy

Figure 1 PODM REST API hierarchy

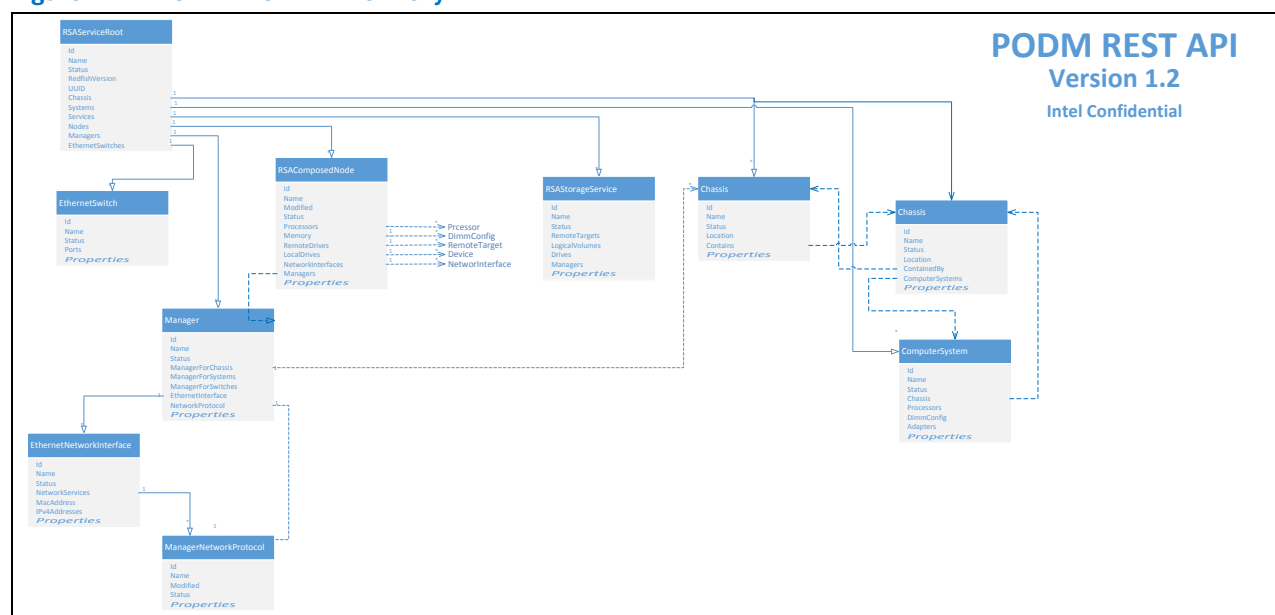


Table 3 Resources and URI

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}
Memory Chunk Collection	/redfish/v1/Systems/{systemID}/MemoryChunk
Memory Chunk	/redfish/v1/Systems/{systemID}/MemoryChunk/{chunkID}
Storage Adapter Collection	/redfish/v1/Systems/{systemID}/Adapters
Storage Adapter	/redfish/v1/Systems/{systemID}/Adapters/{adapterID}
Devices Collection	/redfish/v1/Systems/{systemID}/Adapters/{adapterID}/Devices
Devices	/redfish/v1/Systems/{systemID}/Adapters/{adapterID}/Devices/{deviceID}
Manager Collection	/redfish/v1/Managers
Manager	/redfish/v1/Managers/{managerID}



Resource	URI
Network Protocol	/redfish/v1/Managers/{managerID}/NetworkProtocol
Network Interface Collection	/redfish/v1/Systems/{systemID}/EthernetInterfaces /redfish/v1/Managers/{managerID}/EthernetInterfaces
Network Interface	/redfish/v1/Systems/{systemID}/EthernetInterfaces/{nicID} /redfish/v1/Managers/{managerID}/EthernetInterfaces/{nicID}
Ethernet Switch Collection	/redfish/v1/EthernetSwitches
Ethernet Switch	/redfish/v1/EthernetSwitches/{switchID}
Fabric Switch Port Collection	/redfish/v1/EthernetSwitches/{switchID}/Ports
Fabric Switch Port	/redfish/v1/EthernetSwitches/{switchID}/Ports/{portID}
VLAN Network Interface Collection	/redfish/v1/EthernetSwitches/{switchID}/Ports/{portID}/VLANs
VLAN Network Interface	/redfish/v1/EthernetSwitches/{switchID}/Ports/{portID}/VLANs/{vlanID}
Intel® Rack Scale Design Storage Service Collection	/redfish/v1/Services
Intel® Rack Scale Design Storage Service	/redfish/v1/Services/{serviceID}
Intel® Rack Scale Design Remote Target Collection	/redfish/v1/Services/{serviceID}/Targets
Intel® Rack Scale Design Remote Target	/redfish/v1/Services/{serviceID}/Targets/{targetID}
Intel® Rack Scale Design Logical Drive Collection	/redfish/v1/Services/{serviceID}/LogicalDrives
Intel® Rack Scale Design Logical Drive	/redfish/v1/Services/{serviceID}/LogicalDrives/{driveID}
Intel® Rack Scale Design Physical Drive Collection	/redfish/v1/Services/{serviceID}/Drives
Intel® Rack Scale Design Physical Drive	/redfish/v1/Services/{serviceID}/Drives/{driveID}
Intel® Rack Scale Design Composed Node Collection	/redfish/v1/Nodes
Intel® Rack Scale Design 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}





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® Rack Scale Design 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/$metadata#Message.1.0.0.Message",
        "MessageId": "Base.1.0. MalformedJSON",
        "Message": "The request body submitted was malformed JSON and could not be parsed by the receiving service",
        "Severity": "Error"
      },
      {
        "@odata.type": "/redfish/v1/$metadata#Message.1.0.0.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 HTTP 500 Internal Error code.

3.2.1 General error codes

Table 5 General error codes

Error code	Description	HTTP status code
UnknownException	A generic error message, given when an unexpected condition was encountered and no more specific message is suitable.	500 Internal Error
ServiceUnavailable	The server is currently unable to handle the request due to temporary overloading or maintenance of the server	503 Service unavailable

3.2.2 Request error codes

Table 6 Request error codes

Error code	Description	HTTP status code
InvalidEndpoint	Invalid endpoint in /redfish/v1 namespace	404 Not Found
InvalidHttpMethod	Invalid HTTP request method	405 Method Not Allowed
MalformedUri	Malformed URI	400 Bad Request
BadAcceptHeader	Bad Accept field in request header	406 Not Acceptable
InvalidPayload	Request payload is invalid or missing	400 Bad Request

3.2.3 Node assembly error codes

Table 7 Node assembly error codes

Error code	Description	HTTP status code
NotSufficientResources	Not sufficient resources for creating machine	500 Internal Server Error
UnsupportedCreationRequest	Unable to create machine due to error in machine template or bad creation request	400 BadRequest
ResourcesStateMismatch	Conflict during allocation	409 Conflict

3.2.4 Node action error codes

Table 8 Node disassembly error codes

Error code	Description	HTTP status code
ComposedNodeActionException	Unable to perform action on actual state of the composed node.	409 Conflict



Each resource available on REST API has own list of properties that describes it. Properties can be mandatory; in such cases they must be present in the resulting JSON. When a property is not mandatory, it may not be present in the JSON.





4 PODM REST API Definition

4.1 Odata support

Intel® Rack Scale Design support Odata v4.0 as it is defined in Redfish specification.

All resources within this RESTful API are identified by 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 Intel® Rack Scale Design POD manager service root

Intel® Rack Scale Design POD Manager Service Root resource – entry point. Table 9 lists the attributes.

**Table 9 Intel® Rack Scale Design POD manager service Root attributes**

Name	Service root					
Type URI	/redfish/v1/					
Attribute	Type	Redfish Required	Intel® Rack Scale Design Required	Nullable	Description	
Id	String	No		No	Resource identifier	
Name	String	Yes		No	Name of service root	
Description	String	No			Provides a description of this resource and is used for commonality in the schema definitions	
UUID	String	No			Unique identifier for a service instance – must be constant for particular drawer. The format of this string shall be a 32-byte value in the form 8-4-4-4-12	
RedfishVersion	String	No		No	The version of the Redfish service in format Major.Minor.Errata	
EventService	Object	No			This is the schema definition for the Event Service. It represents the properties for the service itself and has links to the actual list of subscriptions.	
Chassis	Object	No			Link to chassis collection (Drawers)	
Systems	Object	No			Link to Computer Systems collection (logical server nodes)	
Services	Object	No			Link to services collection	
Managers	Object	No			Link to Managers collection	
EthernetSwitches	Object	No			Link to Ethernet Switches collection	
Nodes	Object				Link to Composed Nodes collection	
Oem	Object	No			Oem extension object	
					“Intel_RackScale” extensions:	
					<table><tr><th>Attribute</th><th>Type</th><th>Description</th></tr><tr><td>ApiVersion</td><td>String</td><td>Version of Intel® Rack Scale API in format: Major.minor.errata</td></tr></table>	Attribute
Attribute	Type	Description				
ApiVersion	String	Version of Intel® Rack Scale API in format: Major.minor.errata				
Links	Object	Yes		No	Link sections	

4.3.1 Operations

4.3.1.1 GET

Request:

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

Response:

```
{
  "@odata.context": "/redfish/v1/$metadata#ServiceRoot.ServiceRoot",
  "@odata.id": "/redfish/v1/",
  "@odata.type": "#ServiceRoot.1.0.0.ServiceRoot",
  "Id": "RootService",
  "Name": "Root Service",
  "RedfishVersion": "1.0.0",
  "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"
},
"Oem": {
  "Intel:RackScale": {
    "@odata.type": "#Intel.Oem.ServiceRoot",
    "ApiVersion": "1.2.0",
  }
},
"Links": {}
}
```

4.3.1.2 PUT

Operation is not allowed on this resource.

4.3.1.3 PATCH

Operation is not allowed on this resource.

4.3.1.4 POST

Operation is not allowed on this resource.

4.3.1.5 DELETE

Operation is not allowed on this resource.



4.4 Chassis collection

Table 10 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.

4.4.1 Operations

4.4.1.1 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"
    }
  ]
}
```

4.4.1.2 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

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 11 Chassis attributes

Name	Chassis				
Type URI	/redfish/v1/Chassis/{chassisId}				
Attribute	Type	Redfish Required	Intel® Rack Scale Design Required	Nullable	Description
Id	String	No		No	Resource identifier
Name	String	Yes		No	Name of service root
Description	String	No			Provides a description of this resource and is used for commonality in the schema definitions
ChassisType	String	Yes		No	ChassisType shall indicate the physical form factor for the type of chassis. Allowed values: "Pod" - A collection of equipment racks in a large, likely transportable, container "Rack" - An equipment rack, typically a 19-inch wide freestanding unit "Blade" - An enclosed or semi-enclosed, typically vertically-oriented, system chassis which must be plugged into a multi-system chassis to function normally "Enclosure" - A generic term for a chassis that does not fit any other description "RackMount" - A single system chassis designed specifically for mounting in an equipment rack "Expansion" - A chassis which expands the capabilities or capacity of another chassis "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 "Shelf" - An enclosed or semi-enclosed, typically horizontally-oriented, system chassis which must be plugged into a multi-system chassis to function normally "Drawer" - An enclosed or semi-enclosed, typically horizontally-oriented, system chassis which may be slid into a multi-system chassis "Module" - A small, typically removable, chassis or card which contains devices for a particular subsystem or function "Other" - A chassis that does not fit any of these definitions



Name	Chassis						
Type URI	/redfish/v1/Chassis/{chassisId}						
Manufacturer	String	No			This is the manufacturer of this chassis		
Model	String	No			This is the model number for the chassis		
SKU	String	No			This is the SKU for this chassis		
SerialNumber	String	No			The serial number for this chassis		
PartNumber	String	No			The part number for this chassis		
AssetTag	String	No			The user assigned asset tag for this chassis		
IndicatorLED	String	No			The state of the indicator LED, used to identify the chassis. Allowed values: "Lit" "Blinking" "Off" "Unknown"		
Status	Object	No			See chapter 5.1 for status of resource.		
Oem	Object	No			Oem extension object "Intel_RackScale" extensions:		
					Attribute	Type	Description
					Location	Object	Property that show this chassis ID and its parent. Contains following properties: "Id" – string containing location ID of this chassis "ParentId" – string containing location ID of parent chassis
					RMMPresent	Boolean, null	RMM presence in a rack
					RackSupportDisaggregatedPowerCooling	Boolean, null	Indicates if Rack support disaggregated (shared) power and cooling capabilities
					UUID	String, null	Chassis unique ID
					GeoTag	String, null	Provides info about geographically location of this chassis.
Actions	Object	No		No	Chassis actions.		
ThermalZones	Object	No			A reference to ThermalZones collection		
PowerZones	Object	No			A reference to PowerZones collection		
Links	Object	No		No	Link sections		
ComputerSystems	Array	No			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	No			An array of references to the Ethernet switched contained in this chassis.		
ManagedBy	Array	No			An array of references to the managers contained in this chassis		
ManagersInChassis	Array	No			An array of references to the managers located in this chassis		
ContainedBy	Object	No			A reference to the chassis that this chassis is contained by		
Contains	Array	No			An array of references to any other chassis that this chassis has in it		



4.5.1 Operations

4.5.1.1 GET

Request:

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

Response:

```
{
  "@odata.context": "/redfish/v1/$metadata#Chassis/Members/$entity",
  "@odata.id": "/redfish/v1/Chassis/Rack1",
  "@odata.type": "#Chassis.1.0.0.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",
  "PowerState": "On",
  "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"
  },
  "Links": {
    "Contains": [
      {
        "@odata.id": "/redfish/v1/Chassis/Drawer1"
      }
    ]
  }
}
```



```

    ],
    "ContainedBy": null,
    "ComputerSystems": [],
    "Switches": [],
    "ManagedBy": [
      {
        "@odata.id": "/redfish/v1/Managers/RMM"
      }
    ],
    "ManagersInChassis": [
      {
        "@odata.id": "/redfish/v1/Managers/RMM"
      }
    ],
    "Oem": {}
  }
}

```

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 PowerZone collection

PowerZone collection resource.

Table 12 PowerZone collection attributes

Name	PowerZone Collection		
Type URI	/redfish/v1/Chassis/{chassisID}/PowerZones		
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.

4.6.1 Operations

4.6.1.1 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"
    }
  ]
}
```

4.6.1.2 PUT

Operation is not allowed on this resource.

4.6.1.3 PATCH

Operation is not allowed on this resource.

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

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

Table 13 PowerZone attributes

Name	PowerZone						
Type URI	/redfish/v1/Chassis/{chassisId}/PowerZones/{zoneID}						
Attribute	Type	Redfish Required	Intel® Rack Scale Design Required	Nullable	Description		
Id	String	No		No	Resource identifier		
Name	String	Yes		No	Name of service root		
Description	String	No			Provides a description of this resource and is used for commonality in the schema definitions		
Status	Object	No			See chapter 5.1 for status of resource.		
RackLocation	Object	No			Zone physical location.		
					Attribute	Type	Description
					RackUnits	String	Rack.Units: indicates the rack unit type. "RU": rack unit, each RU 44.5mm



							"OU": openU, each OU 48mm
					XLocation	Number	The horizontal location within uLocation, from left to right(1.. MAXIMUM) 0 indicate not available.
					ULocation	Number	The index of the top-most U of the component, from top to bottom(1..MAXIMUM) 0 indicate not available
					UHeight	Number	The height of managed zone, e.g. 8 for 8U, 16 for 16U
MaxPSUsSupported	Number	No			The maximum number of Power Supply Unit supported by PowerZone.		
Presence	String	No			Indicates the aggregated Power Supply Unit presence information Aggregated Power Supply Unit presence format: Length of string indicate total slot of Power Supply Units in PowerZone. For each byte the string: "1" means present "0" means not present		
NumberOfPSUsPresent	Number	No			Indicates the number of existing Power Supply Unit in PowerZone.		
PowerConsumedWatts	Number	No			The total power consumption of PowerZone, sum of trays' power consumption		
PowerOutputWatts	Number	No			The total power production of PowerZone, sum of PSUs' output		
PowerCapacityWatts	Number	No			The maximum power capacity supported by PowerZone.		
PowerSupplies	Array	No		No	Details of the power supplies associated with this system or device. Detailed description in Table 14		
Actions	Object	No		No	Actions that can be executed on zone		
Links	Object	No		No	Links to resources related to this resource.		

Table 14 PowerSupply attributes

Attribute	Type	Redfish Required	Intel® Rack Scale Design Required	Nullable	Description			
Name	String	No			Power Supply name			
PowerCapacityWatts	Number	No			The maximum capacity of this Power Supply			
LastPowerOutputWatts	Number	No			The average power output of this Power Supply			
Manufacturer	String	No			The manufacturer of this Power Supply			
ModelNumber	String	No			The model number for this Power Supply			
FirmwareRevision	String	No			The firmware version for this Power Supply			
SerialNumber	String	No			The serial number for this Power Supply			
PartNumber	String	No			The part number for this Power Supply			
Status	Object	No			Status of resource, see chapter 5.1			
RackLocation	Object	No				Attribute	Type	Description



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

4.7.1 Operations

4.7.1.1 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.1.0.0.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": {}
}

```

4.7.1.2 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 ThermalZone collection

ThermalZone collection resource.

Table 15 ThermalZone collection attributes

Name	ThermalZone Collection		
Type URI	/redfish/v1/Chassis/{chassisID}/ThermalZones		
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.



4.8.1 Operations

4.8.1.1 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"
    }
  ]
}
```

4.8.1.2 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

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

Table 16 ThermalZone attributes

Name	ThermalZone				
Type URI	/redfish/v1/Chassis/{chassisId}/ThermalZones/{zoneID}				
Attribute	Type	Redfish Required	Intel® Rack Scale Design Required	Nullable	Description
Id	String	No		No	Resource identifier
Name	String	Yes		No	Name of service root
Description	String	No			Provides a description of this resource and is used for commonality in the schema definitions
Status	Object	No			See chapter 5.1 for status of resource.



RackLocation	Object	No			Zone physical location.		
					Attribute	Type	Description
					RackUnits	String	Rack.Units: indicates the rack unit type. "RU": rack unit, each RU 44.5mm "OU": openU, each OU 48mm
					XLocation	Number	The horizontal location within uLocation, from left to right(1.. MAXIMUM) 0 indicate not available.
					ULocation	Number	The index of the top-most U of the component, from top to bottom(1..MAXIMUM) 0 indicate not available
UHeight	Number	The height of managed zone, e.g. 8 for 8U, 16 for 16U					
Presence	String	No			Indicates the aggregated Power Supply Unit presence information Aggregated Power Supply Unit presence format: Length of string indicate total slot of Power Supply Units in PowerZone. For each byte the string: "1" means present "0" means not present		
DesiredSpeedPWM	Number	No			The desired FAN speed in current ThermalZone present in PWM unit		
DesiredSpeedRPM	Number	No			The desired FAN speed in current ThermalZone present in RPM unit		
MaxFansSupported	Number	No			Number of maximum fans that can be installed in given Thermal Zone		
NumberOfFansPresent	Number	No			The existing number of Fan in current ThermalZone.		
VolumetricAirflow	Number	No			Rack Level PTAS Telemetry – Volumetric airflow in current ThermalZone.		
Temperatures	Array	No		No	Array of temperature sensors. Detailed description in Table 18		
Fans	Array	No		No	Details of the fans associated with this thermal zone. Detailed description in Table 14		
Actions	Object	No		No	Actions that can be executed on thermal zone		
Links	Object	No		No	Links to resources related to this resource.		

Table 17 Fan attributes

Attribute	Type	Redfish Required	Intel® Rack Scale Design Required	Nullable	Description		
Name	String	No			Power Control Function name		
ReadingRPM	Number	No			Fan rpm reading		
Status	Object	No			Status of resource, see chapter 5.1		
RackLocation	Object	No			Fan physical location.		
					Attribute	Type	Description



Attribute	Type	Redfish Required	Intel® Rack Scale Design Required	Nullable	Description
					<div>RackUnitsStringRack.Units: indicates the rack unit type. "RU": rack unit, each RU 44.5mm "OU": openU, each OU 48mm</div>
					<div>XLocationNumberThe horizontal location within uLocation, from left to right(1.. MAXIMUM) 0 indicate not available.</div>
					<div>ULocationNumberThe index of the top-most U of the component, from top to bottom(1..MAXIMUM) 0 indicate not available</div>
					<div>UHeightNumberThe height of managed fun, e.g. 8 for 8U, 16 for 16U</div>

Table 18 Temperature attributes

Attribute	Type	Redfish Required	Intel® Rack Scale Design Required	Nullable	Description
Name	String	No			Power Supply name
ReadingCelsius	Number	No			Current value of the temperature sensor's reading
PhysicalContext	String	No			Describes the area or device to which this temperature measurement applies: "Intake" - The intake point of the chassis "Exhaust" - The exhaust point of the chassis "Backplane" - A backplane within the chassis "PowerSupply" - A power supply "SystemBoard" - The system board (PCB) "ComputeBay" - Within a compute bay "PowerSupplyBay" - Within a power supply bay
Status	Object	No			See chapter 5.1 for status of resource.

4.9.1 Operations

4.9.1.1 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.1.0.0.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": {} }
```

4.9.1.2 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 Storage service collection

Intel® Rack Scale Design storage service collection resource – provides collection of available storage services. Table 19 lists the attributes.

Table 19 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.

4.10.1 Operations

4.10.1.1 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"
  }
]
}

```

4.10.1.2 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 Composed node collection

Intel® Rack Scale Design Composed Node collection resource – provides collection of all logical nodes. Table 20 lists the attributes.

Table 20 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	<p>Actions available:</p> <p>Allocate – this action is first mandatory step to create composed node. In response for this action proper resources will be find and allocated for node composition. Node resource will be created and URL (link) of this node will be returned.</p> <p>To allocate a Composed Node using PodM REST API it is necessary to create a JSON template describing requested resources.</p> <p>The JSON template may contain various details concerning resources to be used in Composed Node. All JSON template elements are optional, but each should be coherent itself. It is possible to supply PodM with a JSON template containing no specific requirements (e.g. {} – a pair of empty curly braces in HTTP request body) thus allowing PodM to propose a Composed Node containing resources chosen arbitrarily by PodM.</p> <p>Format of JSON template (action payload) is described in Table 21</p>



			For more information about node allocation and assembly, refer to PODM_Allocation_Guide document.
--	--	--	---------------------------------------------------------------------------------------------------

Table 21 Composed node Allocation action attributes

Attribute	Type	Mandatory	Description																											
Name	String	No	Name of composed node. Note: Because ComposedNode is a Redfish resource - its 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 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: <table><tr><th>Attribute</th><th>Type</th><th>Description</th></tr><tr><td>Model</td><td>String</td><td>Processor model that should be used for composed node (exact model name)</td></tr><tr><td>TotalCores</td><td>Number</td><td>Minimum number of processor cores</td></tr><tr><td>AchievableSpeedMHz</td><td>Number</td><td>Minimum achievable processor operating frequency.</td></tr><tr><td>InstructionSet</td><td>String</td><td>Processor supported instruction set. "x86" – x86 32-bit "x86-64" – x86 64-bit "IA-64" – Intel IA-64 "ARM*-A32" – ARM 32-bit "ARM-A64" – ARM 64-bit "MIPS32" – MIPS 32-bit "MIPS64" – MIPS 64-bit "OEM" – OEM-defined</td></tr><tr><td>Resource</td><td>Object</td><td>Reference to particular processor that should be used in composed node</td></tr><tr><td>Chassis</td><td>Object</td><td>Link to chassis object within this processor should be contained.</td></tr><tr><td>Brand</td><td>String</td><td>Brand of CPU that should be used to allocate node. Allowable values: Xeon family: E3, E5, E7 SoC/Atom family: X3 (Avoton), X5 (Broadwell-DE), X7 Core family: I3, I5, I7 "Unknown" – processor doesn't fit to any above categories</td></tr><tr><td>Capabilities</td><td>Array</td><td>Array of strings describing processor capabilities (like reported in /proc/cpuinfo flags), such as:</td></tr></table>	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. "x86" – x86 32-bit "x86-64" – x86 64-bit "IA-64" – Intel IA-64 "ARM*-A32" – ARM 32-bit "ARM-A64" – ARM 64-bit "MIPS32" – MIPS 32-bit "MIPS64" – MIPS 64-bit "OEM" – OEM-defined	Resource	Object	Reference to 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: Xeon family: E3, E5, E7 SoC/Atom family: X3 (Avoton), X5 (Broadwell-DE), X7 Core family: I3, I5, I7 "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:
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. "x86" – x86 32-bit "x86-64" – x86 64-bit "IA-64" – Intel IA-64 "ARM*-A32" – ARM 32-bit "ARM-A64" – ARM 64-bit "MIPS32" – MIPS 32-bit "MIPS64" – MIPS 64-bit "OEM" – OEM-defined																												
Resource	Object	Reference to 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: Xeon family: E3, E5, E7 SoC/Atom family: X3 (Avoton), X5 (Broadwell-DE), X7 Core family: I3, I5, I7 "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:																												



					"sse" - Streaming SIMD Extensions "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
			DimmDeviceType	String	Type details of DIMM: "DDR" "DDR2" "DDR3" "DDR4" "DDR4 SDRAM" "DDR4E SDRAM" "LPDDR4SDRAM" "DDR3 SDRAM" "LPDDR3 SDRAM" "DDR2 SDRAM" "DDR2 SDRAM-FB-DIMM" "DDR2SDRAM-FB-DIMM PROBE" "DDR SGRAM" "DDR SDRAM" "ROM" "SDRAM" "EDO" "FastPageMode" "PipelinedNibble"
			SpeedMHz	Number	Minimum supported memory speed.
			Manufacturer	String	Requested memory manufacturer.
			DataWidthBits	Number	Requested memory data width in bits.
			Resource	Object	Reference to particular memory module that should be used in composed node
			Chassis	Object	Link to chassis object within this memory dimm should be contained.
RemoteDrives	Array	No	Array of requirements for remote drives that should be created/connected to composed node		
			Attribute	Type	Description
			CapacityGiB	Number	Minimum drive capacity requested for composed node
			iSCSIAddress	String	Defines TargetIQN of RemoteTarget to be set for new Remote Target (should be unique in PodM). Note: Master property is required for creating new target.



			Master	Object	Defines master logical volume that should be taken to create new remote target. It contains properties described in Table 22
LocalDrives	Array	No	Array of requirements for local drives for composed node.		
			Attribute	Type	Description
			CapacityGiB	Number	Minimum drive capacity requested for composed node
			Type	String	Drive type "HDD" "SSD" "NVMe"
			MinRPM	Number	Minimum rotation speed of requested drive.
			SerialNumber	String	Serial number of requested drive
			Interface	String	Interface of requested drive: "SAS" "SATA" "PCIe*"
			Resource	Object	Reference to particular local drive that should be used in composed node
			Chassis	Object	Link to chassis object within this drive should be contained.
EthernetInterfaces	Array	No	Array of requirements for Ethernet interfaces of composed node		
			Attribute	Type	Description
			SpeedMbps	Number	Minimum speed of Ethernet interface requested for composed node
			VLANs	Array	Array of VLANs that should be configured on connected switch port for composed node for given Ethernet interface in following format: VLANid – number indicating VLAN Id Tagged – Boolean value describing if given VLAN is tagged.
			PrimaryVLAN	Number	Primary VLAN ID that should be set for given Ethernet Interface
			Resource	Object	Reference to particular Ethernet interface that should be used in composed node
			Chassis	Object	Link to chassis object within this network interface should be contained.

**Table 22 Remote master target properties**

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

4.11.1 Operation

4.11.1.1 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"
    }
  }
}
```

4.11.1.2 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

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: Below JSON is just an example. For details see Appendix 6, section 6.1.

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,
        "DimmDeviceType": "DDR3",
        "SpeedMHz": 1600,
        "Manufacturer": "Intel",
        "DataWidthBits": 64,
        "Resource": {
            "@odata.id":
"/redfish/v1/Systems/System1/DimmConfig/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/Systems/System1/Adapters/Adapter1/Devices/Device"
        }
    ]],
    "EthernetInterfaces": [{
        "SpeedMbps": 1000,
        "PrimaryVLAN": 100,
        "VLANs": [{
            "VLANId": 100,
```



```

        "Tagged": false
      },
      "Resource": {
        "@odata.id":
"/redfish/v1/Systems/System1/EthernetInterfaces/LAN1"
      }
    },
    "Oem": {
      }
  }
}

```

Response:

```

HTTP/1.1 201 Created
Location: http://<IP>:<Port>/redfish/v1/Nodes/2

```

4.11.1.5 DELETE

Operation is not allowed on this resource.

4.12 Composed node

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

Table 23 Composed node attributes

Name	Composed node		
Type URI	/redfish/v1/Nodes/{nodeID}		
Attribute	Type	Mandatory	Description
Id	String	Yes	Provides a ID of this resource
Name	String	Yes	Name of composed node
Description	String	No	User provided node description



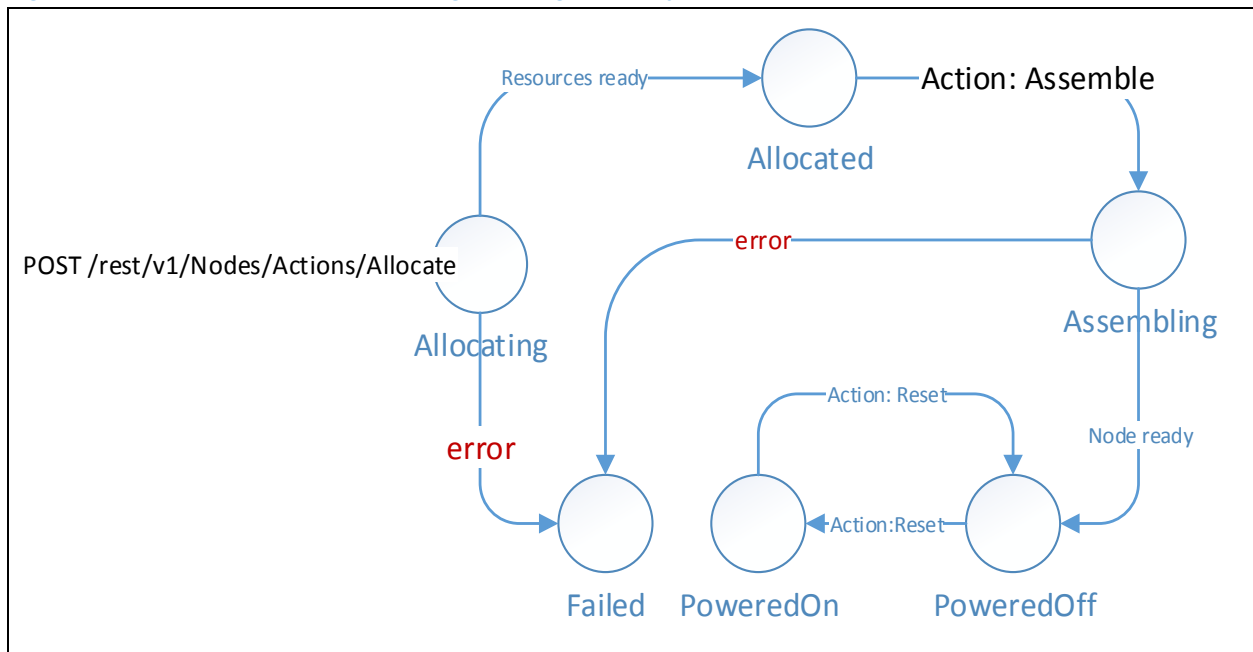
Name	Composed node																		
Type URI	/redfish/v1/Nodes/{nodeID}																		
Attribute	Type	Mandatory	Description																
SystemType	String (enum)	No	Type of computer system: "Logical" – logical instance of computer system																
AssetTag	String	No	The user assigned asset tag for this chassis																
Manufacturer	String	No	This is the manufacturer of computer system used as a base for node.																
Model	String	No	Model of computer system used as a base for this node.																
SKU	String	No	SKU of computer system used as a base for this node.																
SerialNumber	String	No	Serial number of computer system used as a base for this node.																
PartNumber	String	No	Part number of computer system used as a base for this node.																
UUID	String	No	UUID of computer system used as a base for this node.																
HostName	String	No	Hostname of composed node																
PowerState	String (enum)	No	This is the current power state of the node "On" – The system is powered on "Off" – The system is powered off, although some components may continue to have AUX power such as management controller "PoweringOn" – A temporary state between Off and On. This temporary state can be very short. "PoweringOff" – A temporary state between On and Off. The power off action can take time while the OS is in the shutdown process																
BiosVersion	String	No	The version of the system BIOS or primary system firmware																
Processors	Object	Yes	<table> <tr> <th>Name</th><th>Type</th><th>Mandatory</th><th>Description</th></tr> <tr> <td>Count</td><td>Number</td><td>No</td><td>Number of CPUs</td></tr> <tr> <td>Model</td><td>String, Null</td><td>No</td><td>Basic information about processor model</td></tr> <tr> <td>Status</td><td>Object</td><td>No</td><td>See chapter 5.1 for status of resource.</td></tr> </table>	Name	Type	Mandatory	Description	Count	Number	No	Number of CPUs	Model	String, Null	No	Basic information about processor model	Status	Object	No	See chapter 5.1 for status of resource.
Name	Type	Mandatory	Description																
Count	Number	No	Number of CPUs																
Model	String, Null	No	Basic information about processor model																
Status	Object	No	See chapter 5.1 for status of resource.																
Memory	Object	Yes	<table> <tr> <th>Name</th><th>Type</th><th>Mandatory</th><th>Description</th></tr> <tr> <td>TotalSystemMemoryGiB</td><td>Number</td><td>No</td><td>Amount of installed memory in GiB</td></tr> <tr> <td>Status</td><td>Object</td><td>No</td><td>See chapter 5.1 for status of resource.</td></tr> </table>	Name	Type	Mandatory	Description	TotalSystemMemoryGiB	Number	No	Amount of installed memory in GiB	Status	Object	No	See chapter 5.1 for status of resource.				
Name	Type	Mandatory	Description																
TotalSystemMemoryGiB	Number	No	Amount of installed memory in GiB																
Status	Object	No	See chapter 5.1 for status of resource.																
Status	Object, null	No	See chapter 5.1 for status of resource.																
ComposedNodeState	String (enum)	Yes	<p>Current state of assembly process for this node.</p> <ul style="list-style-type: none"> Allocating: Allocating resources for node is in progress. Next state can be <i>Allocated</i> or <i>Failed</i> Allocated: Node resources has been allocated, but assembly not started yet. After <code>ComposedNode.Assemble</code> action state will progress to <i>Assembling</i>. Assembling: Assembly process initiated, but not finished yet. When assembly is done it will change into <i>PoweredOn</i>. PoweredOn: Node successfully assembled and powered on PoweredOff: Node successfully assembled and powered off Failed: Allocation or assembly process failed, or in runtime one of composing components was removed or transitioned in error state. 																
Boot	Object	No	<table> <tr> <th>Name</th><th>Type</th><th>Required</th><th>Description</th></tr> <tr> <td>BootSourceOverrideEnabled</td><td>String, Null</td><td>No</td><td>State of the Boot Source Override feature. Proper values:</td></tr> </table>	Name	Type	Required	Description	BootSourceOverrideEnabled	String, Null	No	State of the Boot Source Override feature. Proper values:								
Name	Type	Required	Description																
BootSourceOverrideEnabled	String, Null	No	State of the Boot Source Override feature. Proper values:																



Name	Composed node					
Type URI	/redfish/v1/Nodes/{nodeID}					
Attribute	Type	Mandatory	Description			
						"Disabled" "Once" "Continuous"
			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: BootSourceOverrideTarget/Supported
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
Actions	Object	Yes	Actions available for this node: <ul style="list-style-type: none">Reset action with following values:<ul style="list-style-type: none">On - Turn the system onForceOff - Turn the system off immediately (non-graceful) shutdownGracefulRestart - Perform a graceful system shutdown followed by a restart of the systemForceRestart - Perform an immediate (non-graceful) shutdown, followed by a restart of the systemNmi - Generate a Diagnostic Interrupt (usually an NMI on x86 systems) to cease normal operations, perform diagnostic actions and typically halt the system.ForceOn - Turn the system on immediatelyPushPowerButton - Simulate the pressing of the physical power button on this systemGracefulShutdown – initiate a soft-shutdown of OS via ACPI			



Name	Composed node		
Type URI	/redfish/v1/Nodes/{nodeID}		
Attribute	Type	Mandatory	Description
			<ul style="list-style-type: none">Assemble: Doesn't consume any parameters. Second step of creating a composed node (after Allocate Action on Nodes collection). That action will assembly logical node – initiate <i>ComposedNodeState</i> change from <i>Allocated</i> state into <i>Assembling</i> state. After finished assemblation, composed node will stay in Powered Off state. To change its state, one needs to execute Reset action with "On" parameter (make sure that Boot is set properly).

Figure 2 ComposedNodeState changes during assembly process

4.12.1 Operations

4.12.1.1 GET

Request:

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

Response:

```
{
  "@odata.context": "/redfish/v1/$metadata#Nodes/Members/$entity",
  "@odata.id": "/redfish/v1/Nodes/Node1",
  "@odata.type": "#ComposedNode.1.0.0.ComposedNode",
  "Id": "Node1",
  "Name": "Composed Node",
  "Description": "Node #1",
  "SystemType": "Logical",
  "AssetTag": "free form asset tag",
  "Manufacturer": "Manufacturer Name - the same as Computer System",
  "Model": "Model Name - the same as Computer System",
  "SKU": "SKU - the same as Computer System",
}
```



```

"SerialNumber": "2M220100SL - the same as Computer System",
"PartNumber": "Computer1 - the same as Computer System",
"UUID": "00000000-0000-0000-0000-000000000000 - the same as Computer
System",
"HostName": null,
"PowerState": "On",
"BiosVersion": "P79 v1.00 (09/20/2013) - the same as Computer System",
"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"
  ]
},
"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/DimmConfig/Dimm1"
    }
  ],
  "EthernetInterfaces": [
    {

```



```
        "@odata.id":
"/redfish/v1/Systems/System1/EthernetInterfaces/LAN1"
    },
    "LocalDrives": [
        {
            "@odata.id":
"/redfish/v1/Systems/System1/StorageControllers/Controller1/Drives/Drive1"
        }
    ],
    "RemoteDrives": [
        {
            "@odata.id": "/redfish/v1/Services/RSS1/Targets/target1"
        }
    ],
    "ManagedBy": [
        {
            "@odata.id": "/redfish/v1/Managers/PODM"
        }
    ],
    "Oem": {}
},
"Actions": {
    "#ComposedNode.Reset": {
        "target": "/redfish/v1/Systems/1/Actions/ComposedNode.Reset",
        "ResetType@Redfish.AllowableValues": [
            "On",
            "ForceOff",
            "GracefulRestart",
            "ForceRestart",
            "Nmi",
            "ForceOn",
            "PushPowerButton",
            "GracefulShutdown"
        ]
    },
    "#ComposedNode.Assemble": {
        "target": "/redfish/v1/Systems/1/Actions/ComposedNode.Assemble"
    }
}
}
```

4.12.1.2 PUT

Operation is not allowed on this resource.

4.12.1.3 PATCH

Currently only "Boot" property is patchable. Following table describe "Boot" properties that can be patched:

Attribute	Type	Required	Description
BootSourceOverrideEnabled	String	No	Describes the state of the Boot Source Override feature. Allowed values: "Disabled" - The system will boot as normal "Once" - On its next boot cycle, the system will boot (one time) to the Boot Source Override Target "Continuous" - The system will boot to the target specified in the BootSourceOverrideTarget until this property is set to Disabled



<code>BootSourceOverrideTarget</code>	String	No	The current boot source to be used at next boot instead of the normal boot device, if <code>BootSourceOverrideEnabled</code> is true. Supported values: "None" - Boot from the normal boot device "Pxe" - Boot from the Pre-Boot EXecution (PXE) environment "Hdd" - Boot from a hard drive
---------------------------------------	--------	----	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

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

Response:

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

4.12.1.4 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
```

4.12.1.5 DELETE

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

- Graceful shutdown 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.
- Computer System is deallocated.
- Remote Target is deallocated (when used in composition).

Request:

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



Response:

HTTP/1.1 204 No Content

4.13 PSME resources

PodM aggregates and exposes PSME resources such as:

- ComputerSystem
- Processor
- DimmConfig
- MemoryChunk
- StorageAdapter
- Device (storage)
- NetworkInterface
- VLAN (PODM only supports GET operations on that resource. Creating/removing VLANs is done automatically during assembly process.)
- Manager
- EthernetSwitch
- SwitchPort
- EventService –not implemented in current release
- EventSubscription – not implemented in current release
- Event – not implemented in current release

For above resources see PSME API Specification.

4.14 Storage Services resources

PodM aggregates and exposes Storage Services resources such as:

- StorageService
- RemoteTarget
- LogicalDrive
- PhysicalDrive

For the above resources see Storage Services specification.

4.15 Simple Storage collection

Table 24 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 25 Simple storage attributes

Name	Simple storage				
Type URI	/redfish/v1/Systems/System1/SimpleStorage/{chassisId}				
Attribute	Type	Redfish Required	Intel® Rack Scale Design Required	Nullable	Description
Id	String	No		No	Resource identifier
Name	String	Yes		No	Name of service root
Description		No			Provides a description of this resource and is used for commonality in the schema definitions
UefiDevicePath	String	No			The UEFI device path used to access this storage controller.



Devices	Object	No		No	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 chapter 5.1 for status of resource.
					CapacityBytes	Number, null	No	The size of the storage device
					Oem	Object	No	Oem defined object
Status	Object	No			See chapter 5.1 for status of resource.			
Oem	Object	No			Oem extension object			

4.16.1 Operations

4.16.1.1 GET

Request:

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

Response:

```
{
  "@odata.context": "/redfish/v1/$metadata#SimpleStorage.SimpleStorage",
  "@odata.id": "/redfish/v1/Systems/System1/SimpleStorage/Storage1",
  "@odata.type": "#SimpleStorage.1.1.0.SimpleStorage",
  "Id": "Storage1",
  "Name": "Simple Storage Controller",
  "Description": "System SATA",
  "UEFIDevicePath": "UEFI Device Path",
  "Status": {
    "State": "Enabled",
    "Health": "OK",
    "HealthRollup": "OK"
  },
  "Devices": [
    {
      "Name": "Drive 1",
      "Manufacturer": "ACME",
      "Model": "Drive Model string",
      "CapacityBytes": 322122547200,
      "Status": {
        "State": "Enabled",
```



```
        "Health": "OK"
    },
    {
        "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.





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 chapter 5.1.
Health	String	Yes	This represents the health state of this resource in the absence of its dependent resources. Allowed values: See chapter 5.3.
HealthRollup	String	Yes	This represents the overall health state from the view of this resource. Allowed values: See chapter 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
- 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
- StandbySpare - This function or resource is part of a redundancy set and is awaiting a failover or other external action to activate it

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

5.5 BootSourceOverrideTarget/Supported

- None: Boot from the normal boot device
- Pxe: Boot from the preboot execution (PXE) environment
- Floppy: Boot from the floppy disk drive
- Cd: Boot from the 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



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 a HTTP POST request on Composed Node Collection Action URI located at `"/redfish/v1/Nodes/Actions/Allocate"` on 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 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 a default name. The example below will allocate 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 a *null* value can be passed for a specified key. **Notes, limitations** provide additional hints about a specific requirement.

6.1.3.1 Location requirements

Processor, Memory, Local Drive and Ethernet Interface sections may contain Resource and Chassis objects. Resource must contain a 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 a 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"
    },
    "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
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

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

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,
    "DimmDeviceType": "DDR3",
    "SpeedMHz": 1600,
  ]
}

```



```
    "Manufacturer": "Intel",
    "DataWidthBits": 64,
    "Resource": {
      "@odata.id": "/redfish/v1/Systems/1/DimmConfig/1"
    },
    "Chassis": {
      "@odata.id": "/redfish/v1/Chassis/1"
    }
  }
}
```

Attribute	Type	Allowed values	Nullable	Notes, limitations
CapacityMiB	Number		Yes	Positive value expected
DimmDeviceType	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

Allocation assumptions:

- Which Memory Modules (represented by DimmConfig resource) will meet supplied requirements?
 - located on the same computer system as other resources
 - with exact match on DimmDeviceType
 - 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 CapacityMiB, 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 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 supplied
Master → Address	Object		No	Pod Manager URI of discovered Logical Volume. Required if Master Drive supplied.

6.1.6.1 Using existing Remote Drive

To use an existing Drive it is necessary to:

- set 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 Device resource under System Adapters). 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/Systems/1/Adapters/1/Devices/1"
        },
        "Chassis": {
            "@odata.id": "/redfish/v1/Chassis/1"
        }
    }]
}
```

Attribute	Type	Allowed values	Nullable	Description
CapacityGiB	Number		Yes	Positive value expected
Type	String	"HDD", "SSD", "NVMe"	Yes	One of allowed, enumerated values
MinRPM	Number		Yes	Positive integer value expected
SerialNumber	String		Yes	
Interface	String	"PCIe", "SAS", "SATA"	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

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

6.1.8 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. 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.

The example file looks like:

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

}

Where 1, 170, 4088, 4091, 4094 are VLANs which 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 a Local Drive
- Allocating resources for Composed Node to be booted from an existing Remote Drive
- Allocating resources for Composed Node to be booted from a Remote Drive that need to be created
- Allocating resources for Composed Node with VLAN requirements specified. This scenario is used with one of the other three

The allocation process is preceded by a general verification of JSON template that checks if the 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 Remote Drive to be used with 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 requested number of cores
 - With at least requested frequency
 - Exactly matching requested instruction set
- Filter Computer Systems by Memory: return all Computer Systems with at least total requested size of memory located on Memory Modules that each of them meet requirements (if supplied in template):
 - Memory of exactly requested dimm device type
 - With at least requested speed MHz
 - With exact requested manufacturer
 - With at least 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 requested capacity specified
 - Exactly matching requested Drive type
 - With at least requested min RPM
 - With exact requested serial number
 - With 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 requested speed
 - If 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 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 Ethernet Switch Port and Computer System's Ethernet interface. This mapping is done using the MAC address as an identifier. Fields used for this mapping:

- 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 specified VLANs requirement.

6.1.9.3 Detailed process of selecting Remote Drives

- Determine what type of Remote Drive is requested
- When requesting 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 allocates 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 resulting filtered collection is selected as a placement for new Logical Volume, which will be exposed as a new Target (Remote Drive)
 - A new Logical Volume is created on selected Logical Volume Group (as a snapshot or as a clone)
 - A new Target is created on top of 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 new REST resource at `/redfish/v1/Nodes/{Nodeid}` when a proper Computer System was found and was 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, user has to send a 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 "PoweredOff".
 - When 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 "PoweredOff"
 - Assembly process doesn't end with sending power on request, so it's necessary to perform `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 HTTP DELETE on all rejected proposals of Composed Nodes to free the resources allocated by them.

