# Contents

1 **Introduction** ................................................................................................................................. 6  
  1.1 Scope.................................................................................................................................................. 6  
  1.2 Intended audience ............................................................................................................................... 6  
  1.3 Terminology ....................................................................................................................................... 6  
  1.4 References ......................................................................................................................................... 7  
  1.5 Notes and Symbol Convention ......................................................................................................... 7  
2 **Storage Services API** ..................................................................................................................... 8  
  2.1 Storage Services API structure and relations ..................................................................................... 8  
    2.1.1 Storage Services API physical resource hierarchy ....................................................................... 8  
3 **Storage Services REST API Error Codes** ...................................................................................... 10  
  3.1 API error response ............................................................................................................................. 10  
    3.1.1 Example error JSON object ....................................................................................................... 10  
  3.2 API error codes................................................................................................................................. 11  
    3.2.1 General error codes ................................................................................................................. 11  
    3.2.2 Request error codes ................................................................................................................. 11  
4 **Storage Services REST API Definition** ....................................................................................... 12  
  4.1 Odata support ................................................................................................................................... 12  
    4.1.1 Operations ............................................................................................................................... 12  
  4.2 Service root .................................................................................................................................... 12  
    4.2.1 Operations ............................................................................................................................... 13  
  4.3 Storage service collection .................................................................................................................. 14  
    4.3.1 Operations ............................................................................................................................... 14  
  4.4 Storage service ............................................................................................................................... 14  
    4.4.1 Operations ............................................................................................................................... 14  
  4.5 Remote target collection .................................................................................................................... 15  
    4.5.1 Operations ............................................................................................................................... 15  
  4.6 Remote target .................................................................................................................................. 18  
    4.6.1 Operations ............................................................................................................................... 18  
  4.7 Logical drive collection ..................................................................................................................... 20  
    4.7.1 Operations ............................................................................................................................... 20  
  4.8 Logical drive .................................................................................................................................... 21  
    4.8.1 Operations ............................................................................................................................... 22  
  4.9 Physical drive collection .................................................................................................................... 23  
    4.9.1 Operations ............................................................................................................................... 23  
  4.10 Physical drive ............................................................................................................................... 24  
    4.10.1 Operations .............................................................................................................................. 24  
  4.11 Manager ......................................................................................................................................... 25  
  4.12 Network protocol ............................................................................................................................. 25  
  4.13 Ethernet interface collection .......................................................................................................... 25  
  4.14 Ethernet interface ........................................................................................................................... 25  
5 **Common Property Description** .................................................................................................... 26  
  5.1 Status .............................................................................................................................................. 26  
  5.2 Status -> State ................................................................................................................................. 26  
  5.3 Status -> Health .............................................................................................................................. 26
## Revision History

<table>
<thead>
<tr>
<th>Revision</th>
<th>Description</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>001</td>
<td>Initial Release</td>
<td>December 19, 2017</td>
</tr>
</tbody>
</table>
1 Introduction

1.1 Scope
This document contains information about the Intel® Rack Scale Design Storage Services REST API, which is designed and implemented for the Intel® Rack Scale Design Software v2.2 release.

There are no technical changes from the Intel® RSD 2.1.3 Storage Services API Specifications.

1.2 Intended audience
The intended audiences for this document include designers and engineers working with the Intel® Rack Scale Design Software v2.2 release.

1.3 Terminology

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC</td>
<td>Baseboard management controller</td>
</tr>
<tr>
<td>HTTP</td>
<td>Hypertext Transfer Protocol</td>
</tr>
<tr>
<td>JSON</td>
<td>JavaScript object notation</td>
</tr>
<tr>
<td>NIC</td>
<td>Network interface card</td>
</tr>
<tr>
<td>OCCI</td>
<td>Open Cloud Computing Interface</td>
</tr>
<tr>
<td>OData</td>
<td>Open data protocol</td>
</tr>
<tr>
<td>OVF</td>
<td>Open virtualization format</td>
</tr>
<tr>
<td>Pod</td>
<td>A physical collection of multiple racks.</td>
</tr>
<tr>
<td>PODM</td>
<td>Pod Manager</td>
</tr>
<tr>
<td>PSME</td>
<td>Pooled System Management Engine</td>
</tr>
<tr>
<td>REST</td>
<td>Representational State Transfer</td>
</tr>
<tr>
<td>SDV</td>
<td>Software development vehicle</td>
</tr>
<tr>
<td>URI</td>
<td>Uniform resource identifier</td>
</tr>
<tr>
<td>UUID</td>
<td>Universally unique identifier</td>
</tr>
<tr>
<td>VM</td>
<td>Virtual machine</td>
</tr>
<tr>
<td>XML</td>
<td>Extensible Markup Language</td>
</tr>
</tbody>
</table>
1.4 References

Table 2. Reference Documents

<table>
<thead>
<tr>
<th>Doc ID</th>
<th>Title</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>336811</td>
<td>Intel® Rack Scale Design (RSD) Conformance and Software Reference Kit</td>
<td><a href="http://www.intel.com/intelRSD">Getting Started Guide v2.2, Revision 001</a></td>
</tr>
<tr>
<td>336814</td>
<td>Intel® Rack Scale Design Pod Manager (PDOM) Release Notes, Software v2.2, Revision 001</td>
<td></td>
</tr>
<tr>
<td>336815</td>
<td>Intel® Rack Scale Design Pod Manager (PDOM) User Guide, Software v2.2, Revision 001</td>
<td></td>
</tr>
<tr>
<td>336816</td>
<td>Intel® Rack Scale Design PSME Release Notes, Software v2.2, Revision 001</td>
<td></td>
</tr>
<tr>
<td>336810</td>
<td>Intel® Rack Scale Design PSME User Guide, Software v2.2, Revision 001</td>
<td></td>
</tr>
<tr>
<td>336855</td>
<td>Intel® Rack Scale Design PSME REST API Specification, Software v2.2, Revision 001</td>
<td></td>
</tr>
<tr>
<td>336857</td>
<td>Intel® Rack Scale Design Pod Manager REST API Specification, Software v2.2, Revision 001</td>
<td></td>
</tr>
<tr>
<td>336858</td>
<td>Intel® Rack Scale Design Rack Management Module (RMM) API Specification, Software v2.2, Revision 001</td>
<td></td>
</tr>
<tr>
<td>336859</td>
<td>Intel® Rack Scale Design Generic Assets Management Interface API Specification, Software v2.2, Revision 001</td>
<td></td>
</tr>
<tr>
<td>336860</td>
<td>Intel® Rack Scale Design Firmware Extension Specification, Software v2.2, Revision 001</td>
<td></td>
</tr>
<tr>
<td>336861</td>
<td>Intel® Rack Scale Design Architecture Specification, Software v2.2, Revision 001</td>
<td></td>
</tr>
<tr>
<td>336862</td>
<td>Intel® RSD v2.2 Solid State Drive (SSD) Technical Advisory</td>
<td></td>
</tr>
<tr>
<td>RFC2119</td>
<td>Key words for use in RFCs to Indicate Requirement Levels, March 1997</td>
<td><a href="https://www.ietf.org/rfc/rfc2119.txt">https://www.ietf.org/rfc/rfc2119.txt</a></td>
</tr>
<tr>
<td>SDP0266</td>
<td>Scalable Platforms Management API Specification v1.1.0</td>
<td><a href="https://www.dmtf.org/sites/default/files/standards/documents/DSP0266_1.1.0.pdf">https://www.dmtf.org/sites/default/files/standards/documents/DSP0266_1.1.0.pdf</a></td>
</tr>
</tbody>
</table>

1.5 Notes and Symbol Convention

- Symbol and note convention are similar to typographical conventions used in CIMI specification.
- Notation used in JSON serialization description:
  - Mandatory in italics indicate data types instead of literal Mandatory.
  - Characters are appended to items to indicate cardinality:
    - "?" (0 or 1)
    - "*" (0 or more)
    - "+" (1 or more)
  - Vertical bars, "|", denote choice. For example, "a|b" means a choice between "a" and "b".
  - Parentheses, "(" and ")", are used to indicate the scope of the operators "?", "+", "*" and "|".
  - Ellipses (i.e., "...") indicate points of extensibility.

Note: The lack of ellipses does not mean no extensibility point exists; rather it is just not explicitly called out.
2 Storage Services API

2.1 Storage Services API structure and relations

The Intel® Rack Scale Design Storage Services REST API provides the REST-based interface that allows full management of Storage Services including storage asset discovery and configuration.

2.1.1 Storage Services API physical resource hierarchy

Figure 1 shows the Intel® Rack Scale Design Storage Services REST API hierarchy.

Figure 1. Intel® Rack Scale Design Storage Services REST API Hierarchy
### Table 3. Resources and URIs

<table>
<thead>
<tr>
<th>Resource</th>
<th>URI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service Root</td>
<td>/redfish/v1</td>
</tr>
<tr>
<td>Manager Collection</td>
<td>/redfish/v1/Managers</td>
</tr>
<tr>
<td>Storage Service Collection</td>
<td>/redfish/v1/Services</td>
</tr>
<tr>
<td>Storage Service</td>
<td>/redfish/v1/Services/{serviceID}</td>
</tr>
<tr>
<td>Remote Target Collection</td>
<td>/redfish/v1/Services/1/Targets</td>
</tr>
<tr>
<td>Remote Target</td>
<td>/redfish/v1/Services/1/Targets/{targetID}</td>
</tr>
<tr>
<td>Logical Drive Collection</td>
<td>/redfish/v1/Services/1/LogicalDrives</td>
</tr>
<tr>
<td>Logical Drive</td>
<td>/redfish/v1/Services/1/LogicalDrives/{driveID}</td>
</tr>
<tr>
<td>Drive Collection</td>
<td>/redfish/v1/Services/1/Drives</td>
</tr>
<tr>
<td>Drive</td>
<td>/redfish/v1/Services/1/Drives/{driveID}</td>
</tr>
<tr>
<td>Manager</td>
<td>/redfish/v1/Managers/{managerID}</td>
</tr>
<tr>
<td>Network Service</td>
<td>/redfish/v1/Managers/{managerID}/NetworkService</td>
</tr>
</tbody>
</table>
This chapter contains descriptions of all error codes that may be returned by the REST calls implemented in the Storage Services REST API of the Intel® Rack Scale Design v2.2 software release.

3.1 API error response

In the case of an error, Storage Services 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 is the JSON object with the properties listed in Table 4.

### Table 4. API Error Response Attributes

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MessageId</td>
<td>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.</td>
</tr>
<tr>
<td>Message</td>
<td>A human readable error message indicating the semantics associated with the error. This is the complete message, and do not rely on substitution variables.</td>
</tr>
<tr>
<td>MessageArgs</td>
<td>An optional array of strings representing the substitution parameter values for the message. This will be included in the response if a MessageId is specified for a parameterized message.</td>
</tr>
<tr>
<td>Severity</td>
<td>An optional string representing the severity of the error.</td>
</tr>
<tr>
<td>Resolution</td>
<td>An optional string describing recommended action(s) to take to resolve the error.</td>
</tr>
<tr>
<td>RelatedProperties</td>
<td>An optional array of JSON Pointers defining the specific properties within a JSON payload described by the message.</td>
</tr>
</tbody>
</table>

**Example error JSON object**

```json
{
  "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 HTTP 500 Internal Error code.

3.2.1 General error codes

Table 5. General Error Codes

<table>
<thead>
<tr>
<th>Error code</th>
<th>Description</th>
<th>HTTP status code</th>
</tr>
</thead>
<tbody>
<tr>
<td>UnknownException</td>
<td>Exception that causes response generation to fail.</td>
<td>500 Internal Error</td>
</tr>
</tbody>
</table>

3.2.2 Request error codes

Table 6. Request Error Codes

<table>
<thead>
<tr>
<th>Error code</th>
<th>Description</th>
<th>HTTP status code</th>
</tr>
</thead>
<tbody>
<tr>
<td>InvalidEndpoint</td>
<td>Invalid endpoint.</td>
<td>404 Not Found</td>
</tr>
<tr>
<td>MalformedUri</td>
<td>URI is malformed.</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>InvalidPayload</td>
<td>Request payload is invalid or missing.</td>
<td>400 Bad Request</td>
</tr>
</tbody>
</table>
4 Storage Services 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 RESTfull API are identified by unique identifier property named "@odata.id". Resource Identifiers are represented in JSON payloads as uri paths relative to the Redfish Schema portion of the uri. Uri paths 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.

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 is "/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 minor versions.
- **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 will conform to the same version of the protocol supported by the root service.

4.1.1 Operations

4.1.1.1 GET

Request:

GET /redfish
Content-Type: application/json

Response:

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

4.2 Service root

Service Root resource – entry point.

Detailed info about this resource property can be obtained from metadata file: ServiceRoot.xml
4.2.1 Operations

4.2.1.1 GET

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

Response:
{
    "@odata.context": "/redfish/v1/$metadata#ServiceRoot",
    "@odata.id": "/redfish/v1",
    "@odata.type": "#ServiceRoot.v1_0_0.ServiceRoot",
    "Id": "RootService",
    "Name": "Root Service",
    "Description": "description-as-string",
    "RedfishVersion": "1.0.0",
    "UUID": "92384634-2938-2342-8820-489239905423",
    "Systems": {},
    "Chassis": {},
    "Managers": {
        "@odata.id": "/redfish/v1/Managers"
    },
    "EventService": {
        "@odata.id": "/redfish/v1/EventService"
    },
    "Services": {
        "@odata.id": "/redfish/v1/Services"
    },
    "EthernetSwitches": {},
    "PCleSwitches": {},
    "Devices": {},
    "Oem": {
        "Intel_RackScale": {
            "@odata.type": "#Intel.Oem.ServiceRoot",
            "ApiVersion": "2.0.0"
        }
    },
    "Links": {}
}

4.2.1.2 PUT
Operation is not allowed on this resource.

4.2.1.3 PATCH
Operation is not allowed on this resource.

4.2.1.4 POST
Operation is not allowed on this resource.

4.2.1.5 DELETE
Operation is not allowed on this resource.
4.3 **Storage service collection**

Storage service collection resource – provides a collection of available storage services.

Detailed info about this resource property can be obtained from metadata file:
StorageServiceCollection.xml

### 4.3.1 Operations

#### 4.3.1.1 GET

**Request:**

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

**Response:**

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

#### 4.3.2 PUT

Operation is not allowed on this resource.

#### 4.3.3 PATCH

Operation is not allowed on this resource.

#### 4.3.4 POST

Operation is not allowed on this resource.

#### 4.3.5 DELETE

Operation is not allowed on this resource.

### 4.4 **Storage service**

Storage service resource – provides detailed information about a storage service provided by PSME.

Detailed info about this resource property can be obtained from metadata file: StorageService.xml

### 4.4.1 Operations

#### 4.4.1.1 GET

**Request:**

```plaintext
GET /redfish/v1/Services/1
Content-Type: application/json
```
Response:

```
{
   "@odata.context": "/redfish/v1/$metadata#Services/Members/1/$entity",
   "@odata.id": "/redfish/v1/Services/RSS1",
   "@odata.type": "#StorageService.v1_0_0.StorageService",
   "Id": "RSS1",
   "Name": "Storage Service",
   "Description": "Storage Service",
   "Status": {
      "State": "Enabled",
      "Health": "OK"
   },
   "RemoteTargets": {
      "@odata.id": "/redfish/v1/Services/RSS1/Targets"
   },
   "LogicalDrives": {
      "@odata.id": "/redfish/v1/Services/RSS1/LogicalDrives"
   },
   "Drives": {
      "@odata.id": "/redfish/v1/Services/RSS1/Drives"
   },
   "Oem": {},
   "Links": {
      "ManagedBy": [
         {
            "@odata.id": "/redfish/v1/Managers/RSS"
         }
      ],
      "Oem": {}
   }
}
```

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  **Remote target collection**

Intel® Rack Scale Design remote target collection resource – provides a collection of available storage remote targets.

Detailed info about this resource property can be obtained from metadata file: `RemoteTargetCollection.xml`

4.5.1  **Operations**
4.5.1.1 GET

Request:

GET /redfish/v1/Services/1/Targets
Content-Type: application/json

Response:

```
{
   "@odata.context": "/redfish/v1/$metadata#RemoteTargets",
   "@odata.id": "/redfish/v1/Services/1/Targets",
   "@odata.type": "#RemoteTargetCollection.RemoteTargetCollection",
   "Name": "Remote Targets Collection",
   "Members@odata.count": 1,
   "Members": [
      {
         "@odata.id": "/redfish/v1/Services/1/Targets/1"
      }
   ]
}
```

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

POST operation is used to create new remote target resource. The following parameters should be used in this call:

**Table 7. Remote Target POST Attributes**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>String</td>
<td>No</td>
<td>Name of target.</td>
</tr>
<tr>
<td>Type</td>
<td>String</td>
<td>No</td>
<td>Type of target</td>
</tr>
<tr>
<td>Addresses</td>
<td>Array</td>
<td>Yes</td>
<td>Array of objects with address of target. Details for currently supported &quot;iSCSI&quot; object are in Table 8 below.</td>
</tr>
<tr>
<td>Initiator</td>
<td>Array</td>
<td>Yes</td>
<td>Array of objects with initiator details. Details in table below.</td>
</tr>
</tbody>
</table>

**Table 8. Details of Addresses "iSCSI" Object**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TargetLUN</td>
<td>Array</td>
<td>Yes</td>
<td>Array of objects mapping Logical drives to LUN numbers:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>TargetLUN</strong>&lt;br&gt;<strong>Attribute</strong>&lt;br&gt;LUN Number&lt;br&gt;<strong>Type</strong>&lt;br&gt;Number&lt;br&gt;<strong>Required</strong>&lt;br&gt;Yes&lt;br&gt;<strong>Description</strong>&lt;br&gt;LUN (Logical Unit Number)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Logical Drive</strong>&lt;br&gt;Link Object&lt;br&gt;<strong>Type</strong>&lt;br&gt;Link Object&lt;br&gt;<strong>Required</strong>&lt;br&gt;Yes&lt;br&gt;<strong>Description</strong>&lt;br&gt;Object containing odata.id reference to Logical Drive assigned to LUN number</td>
</tr>
<tr>
<td>TargetIQN</td>
<td>String</td>
<td>Yes</td>
<td>iSCSI target IQN</td>
</tr>
<tr>
<td>Attribute</td>
<td>Type</td>
<td>Required</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>--------</td>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>CHAP</td>
<td>Object</td>
<td>No</td>
<td>Challenge Handshake Authentication Protocol (CHAP) authentication parameters of iSCSI target.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Username</td>
<td>String</td>
<td>Yes</td>
<td>CHAP one way user name.</td>
</tr>
<tr>
<td>Secret</td>
<td>String</td>
<td>Yes</td>
<td>CHAP one way secret.</td>
</tr>
<tr>
<td>MutualUsername</td>
<td>String</td>
<td>No</td>
<td>CHAP mutual user name.</td>
</tr>
<tr>
<td>MutualSecret</td>
<td>String</td>
<td>No</td>
<td>CHAP mutual secret.</td>
</tr>
</tbody>
</table>

Table 9. Details of Initiator "iSCSI" Object

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InitiatorIQN</td>
<td>String</td>
<td>Yes</td>
<td>String containing InitiatorIQN. If ALL initiators are allowed to connect to target, InitiatorIQN should be an empty string (&quot;&quot;), or an Initiator array should be empty.</td>
</tr>
</tbody>
</table>

Request:

POST /redfish/v1/Services/1/Targets
Content-Type: application/json
{
   "Name": "Remote Target",
   "Type": "Network Storage",
   "Addresses": [
       {
           "iSCSI": {
               "TargetLUN": [
                   {
                       "LUN": 1,
                       "LogicalDrive": {
                           "@odata.id": "/redfish/v1/Services/1/LogicalDrives/1"
                       }
                   }
               ],
               "TargetIQN": "iqn.2015-01.com.example:ceph-ubuntu14",
               "CHAP": {
                   "Type": "Mutual",
                   "Username": "valid_user",
                   "Secret": "my_password",
                   "MutualUsername": "user2",
                   "MutualSecret": "user_password"
               }
           }
       
   
   ]
}
4.5.1.5 DELETE
Operation is not allowed on this resource.

4.6 Remote target
Remote target resource – provides detailed information about the storage remote target.

4.6.1 Operations
4.6.1.1 GET

Note: Because of confidential nature of CHAP secret fields, it won’t be shown in GET request, null will be shown instead.

Request:
GET /redfish/v1/Services/1/Targets/1
Content-Type: application/json

Response:

```json
{
    "@odata.context": "/redfish/v1/$metadata#RemoteTargets/Links/Members/$entity",
    "@odata.id": "/redfish/v1/Services/1/Targets/1",
    "@odata.type": ":RemoteTarget.v1_1_0.RemoteTarget",
    "Id": "1",
    "Name": "Remote Target",
    "Description": "Remote Target",
    "Status": {
        "State": "Enabled",
        "Health": "OK"
    },
    "Type": "Network Storage",
    "Addresses": [
        {"iSCSI": [
            "TargetLUN": [
                {"LUN": 1,
                "LogicalDrive": "/redfish/v1/Services/1/LogicalDrives/1"
            }
        ]
    }
}
"TargetIQN": "iqn.2015-01.com.example:ceph-ubuntu14",
"TargetPortalIP": "10.102.44.54",
"TargetPortalPort": 3260,
"CHAP": {
  "Type": "Mutual",
  "Username": "valid_user",
  "Secret": null,
  "MutualUsername": "user2",
  "MutualSecret": null
}

"Initiator": [
  {
    "iSCSI": {
      "InitiatorIQN": "iqn.2015-01.com.example:fedora21"
    }
  }
],
"Oem": {},
"Links": {}

4.6.1.2 PUT

Operation is not allowed on this resource.

4.6.1.3 PATCH

Request:

PATCH /redfish/v1/Services/1/Targets/2
Content-Type: application/json
{
  "Addresses": [
    {
      "iSCSI": {
        "CHAP": {
          "Type": "Mutual",
          "Username": "valid_user",
          "Secret": "mypass",
          "MutualUsername": "user2",
          "MutualSecret": "userpass"
        }
      }
    },
    "Initiator": [
      {
        "iSCSI": {
          "InitiatorIQN": "iqn.2015-01.com.example:fedora21"
        }
      }
    ]
  ]
}
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

Request:

DELETE /redfish/v1/Services/1/Targets/2

Response:

HTTP/1.1 204 No Content

4.7 Logical drive collection

Logical drive collection resource – provides a collection of available storage logical drives (logical discs, partitions, volume groups, volumes, etc.).

Detailed info about this resource property can be obtained from metadata file: LogicalDriveCollection.xml.

4.7.1 Operations

4.7.1.1 GET

Request:

GET /redfish/v1/Services/1/LogicalDrives
Content-Type: application/json

Response:

{
    "@odata.context": "/redfish/v1/$metadata#LogicalDrives",
    "@odata.id": "/redfish/v1/Services/1/LogicalDrives",
    "@odata.type": ">#LogicalDriveCollection.LogicalDriveCollection",
    "Name": "Logical Drives Collection",
    "Members@odata.count": 1,
    "Members": [
        {
            "@odata.id": "/redfish/v1/Services/1/LogicalDrives/1"
        }
    ]
}

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

Table 10. Logical Drive POST Attributes

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>String</td>
<td>Yes</td>
<td>REST name of drive</td>
</tr>
<tr>
<td>Type</td>
<td>String</td>
<td>Yes</td>
<td>Type of drive – currently only &quot;LVM&quot; is supported</td>
</tr>
<tr>
<td>Mode</td>
<td>String</td>
<td>Yes</td>
<td>Drive mode – for Type=&quot;LVM&quot; only supported mode is &quot;LV&quot;</td>
</tr>
<tr>
<td>Protected</td>
<td>boolean</td>
<td>Yes</td>
<td>If new drive should be protected</td>
</tr>
<tr>
<td>CapacityGiB</td>
<td>Number</td>
<td>Yes</td>
<td>New drive capacity in GiB</td>
</tr>
<tr>
<td>Image</td>
<td>String</td>
<td>No</td>
<td>Any name that identifies the content of image which was copied to this</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Logical Volume (LV)</td>
</tr>
<tr>
<td>Bootable</td>
<td>Boolean</td>
<td>Yes</td>
<td>If the Logical Volume (LV) contains a bootable operating system</td>
</tr>
<tr>
<td>Snapshot</td>
<td>Boolean</td>
<td>Yes</td>
<td>Type of drive replication – Yes – using Copy On Write, No – using disc clone</td>
</tr>
<tr>
<td>links-&gt;</td>
<td>Link object</td>
<td>Yes</td>
<td>Must contain single link to LogicalDrive which is Logical Volume Group</td>
</tr>
<tr>
<td>LogicalDrives</td>
<td></td>
<td></td>
<td>(&quot;Mode&quot; = &quot;LVG&quot;)</td>
</tr>
<tr>
<td>links-&gt; MasterDrive</td>
<td>Link object</td>
<td>Yes</td>
<td>Contains the link to the single LogicalDrive which is Logical Volume</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(&quot;Mode&quot; = &quot;LV&quot;) which should be used as source for snapshot/clone</td>
</tr>
</tbody>
</table>

Request:

POST /redfish/v1/Services/1/LogicalDrives
Content-Type: application/json
{
  "Name": "Logical Drive",
  "Type": "LVM",
  "Mode": "LV",
  "Protected": false,
  "CapacityGiB": 8096,
  "Image": "Ubuntu 12.04.4LTS / Linux 3.11 / 2014.1",
  "Bootable": true,
  "Snapshot": true,
  "links": {
    "LogicalDrives": [
      {
        "@odata.id": "/redfish/v1/Services/1/LogicalDrives/4"
      }
    ],
    "MasterDrive": {
      "@odata.id": "/redfish/v1/Services/1/LogicalDrives/12"
    }
  }
}

Response:

HTTP/1.1 201 Created
Location: http://<IP>:<PORT>/redfish/v1/Services/1/LogicalDrives/2

4.7.1.5 DELETE

Operation is not allowed on this resource.

4.8 Logical drive

Logical drive resource – provides detailed information about a single logical drive available in storage service.

Detailed info about this resource property can be obtained from metadata file: LogicalDrive.xml
4.8.1 Operations

4.8.1.1 GET

Request:

GET /redfish/v1/Services/1/LogicalDrives/1
Content-Type: application/json

Response:

```
{
  "@odata.context": "/redfish/v1/$metadata#LogicalDrives/Links/Members/$entity",
  "@odata.id": "/redfish/v1/Services/1/LogicalDrives/1",
  "@odata.type": "#LogicalDrive.LogicalDrive",
  "Id": "1",
  "Name": "Logical Drive",
  "Description": "Logical Drive",
  "Status": {
    "State": "Enabled",
    "Health": "OK"
  },
  "Type": "LVM",
  "Mode": "RAID0",
  "Protected": false,
  "CapacityGiB": 8096,
  "Image": "Ubuntu 12.04.4LTS / Linux 3.11 / 2014.1",
  "Bootable": true,
  "Snapshot": false,
  "Oem": {},
  "Links": {
    "LogicalDrives": [
    ],
    "PhysicalDrives": [
      {
        "@odata.id": "/redfish/v1/Services/1/Drives/2"
      }
    ],
    "MasterDrive": {
      "@odata.id": "/redfish/v1/Services/1/LogicalDrives/12"
    },
    "UsedBy": [
      {
        "@odata.id": "/redfish/v1/Services/1/LogicalDrives/14"
      }
    ],
    "Targets": [
      {
        "@odata.id": "/redfish/v1/Services/1/Targets/2"
      }
    ],
    "Oem": {}
  }
}
```

4.8.1.2 PUT

Operation is not allowed on this resource.
4.8.1.3  PATCH

Request:
PATCH /redfish/v1/Services/1/LogicalDrives/1
Content-Type: application/json
{
   "Bootable": true,
}

Response:
HTTP/1.1 204 No Content

4.8.1.4  POST

Operation is not allowed on this resource.

4.8.1.5  DELETE

Request:
DELETE /redfish/v1/Services/1/LogicalDrives/5

Response:
HTTP/1.1 204 No Content

4.9  Physical drive collection

Physical drive collection resource – provides a collection of all storage drives available in a storage service.

Detailed info about this resource property can be obtained from metadata file: PhysicalDriveCollection.xml

4.9.1  Operations

4.9.1.1  GET

Request:
GET /redfish/v1/Services/1/Drives
Content-Type: application/json

Response:
[
   
   "@odata.context": "/redfish/v1/$metadata#Drives",
   "@odata.id": "/redfish/v1/Services/1/Drives",
   "@odata.type": ">#PhysicalDriveCollection.PhysicalDriveCollection",
   "Name": "Physical Drives Collection",
   "Members@odata.count": 1,
   "Members": [
      
      
      "@odata.id": "/redfish/v1/Services/1/Drives/1"
   ]
]

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 Physical drive

Physical drive resource – provides detailed information about a single drive identified by `{driveID}`.

Detailed info about this resource property can be obtained from metadata file: `PhysicalDrive.xml`

#### 4.10.1 Operations

##### 4.10.1.1 **GET**
Request:

```
GET /redfish/v1/Services/1/Drives/{driveID}
Content-Type: application/json
```

Response:

```json
{
    "@odata.context": "/redfish/v1/$metadata#Drive/Links/Members/$entity",
    "@odata.id": "/redfish/v1/Services/1/Drives/1",
    "@odata.type": "<<PhysicalDrive.v1_0_0.PhysicalDrive",
    "Id": "1",
    "Name": "Simple drive"
    "Description": "Physical drive"
    "Interface": < { "PCIe", "SAS", "SATA" } >
    "CapacityGiB": 500,
    "Type": < { "HDD", "SSD" } >,
    "RPM": 0,
    "Manufacturer": "Intel",
    "Model": "S3710",
    "SerialNumber": "XYZ123456789",
    "Status": {
        "State": < { "Enabled", "Disabled", "Offline", "InTest", "Starting", "Absent" } >,
        "Health": < { "OK", "Warning", "Critical" } >,
        "HealthRollup": < { "OK", "Warning", "Critical" } >
    },
    "Oem": {},
    "Links": {
        "UsedBy": [ 
            { @odata.id": "/redfish/v1/Services/1/LogicalDrives/1" 
            }]
    }]
}
```

##### 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 Manager
Refer to the Intel® Rack Scale Design PSME REST API Specification, Doc# 560243

4.12 Network protocol
Refer to the Intel® Rack Scale Design PSME API Specification, Doc# 560244

4.13 Ethernet interface collection
Refer to the Intel® Rack Scale Design PSME REST API Specification, Doc# 560243.

4.14 Ethernet interface
Refer to the Intel® Rack Scale Design PSME API Specification, Doc# 560244.
Common Property Description

5

5.1 Status

Table 11. Redfish Status Properties

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Nullable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>State</td>
<td>String</td>
<td>Yes</td>
<td>This indicates the known state of the resource, such as if it is enabled. Allowed values – refer to Status -&gt; Health</td>
</tr>
<tr>
<td>Health</td>
<td>String</td>
<td>Yes</td>
<td>This represents the health state of this resource in the absence of its dependent resources. Allowed values – refer to Status -&gt; Health</td>
</tr>
<tr>
<td>HealthRollup</td>
<td>String</td>
<td>Yes</td>
<td>This represents the overall health state from the view of this resource. Allowed values – refer to Status -&gt; Health</td>
</tr>
</tbody>
</table>

5.2 Status -> State

- **Enabled**: This function or resource has been enabled
- **Disabled**: This function or resource has been disabled
- **Offline**: This function or resource is enabled, but currently unavailable
- **InTest**: This function or resource is under doing testing
- **Starting**: This function or resource is starting
- **Absent**: This function or resource is not installed

5.3 Status -> Health

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