OpenManage Enterprise Power Manager

RESTful API Guide version 2.0

June 2021
Rev. A00

Notes, cautions, and warnings

ⓘ **NOTE:** A NOTE indicates important information that helps you make better use of your product.

△ **CAUTION: A CAUTION indicates either potential damage to hardware or loss of data and tells you how to avoid the problem.**

⚠ **WARNING: A WARNING indicates a potential for property damage, personal injury, or death.**

Contents

1

About this document

This document contains information about the resource models for Dell EMC OpenManage Enterprise Power Manager. Each RESTful API specification includes the URI, method, parameters, request codes, and response codes. For OpenManage Enterprise RESTful API specifications, see *OpenManage Enterprise and OpenManage Enterprise - Modular Edition RESTful API Guide* .

2

Overview

RESTful API support in Dell EMC OpenManage Enterprise Power Manager

Power Manager supports RESTful API. APIs on these products enhance the systems management capabilities. The RESTful interface is provided over HTTPS in JSON format that is based on ODATA v4 usable by clients, scripts, and browser-based GUIs. The APIs enable you to build console management tools that are based on common programming and scripting languages such as Python, Java, and C.

ⓘ **NOTE:** If you are using the REST APIs to create objects while the Power Manager User Interface (UI) is open, ensure that you refresh the UI before viewing the changes made through APIs.

For more information about REST, see publically available resources.

Topics:

Introduction to Power Manager

Dell EMC OpenManage Enterprise Power Manager is a plugin to the Dell EMC OpenManage Enterprise (OME) console and uses fine-grained instrumentation to provide increased visibility to power consumption, system anomalies, and resource utilization on Dell EMC servers. Also, Power Manager supports virtual machine monitoring, and alerts and reports about power and thermal events in servers, chassis, and custom groups consisting of servers and chassis. This reporting enables increased control, faster response times, greater accuracy, and broader decision-making intelligence than is otherwise possible.

When used with PowerEdge servers, modular systems with an iDRAC Enterprise or iDRAC Datacenter license, or supported chassis and OpenManage Enterprise Advanced license, Power Manager leverages information from the OME console to deliver platform-level power reporting. Power Manager then communicates with Integrated Dell Remote Access Controller (iDRAC) or Chassis Management Controller (CMC) on each managed device to provide power-management data and execution of control policy—making it easy for administrators to identify areas in which they can gain efficiencies and cut wasteful costs. Also, Power Manager supports monitoring and managing power load on Power Distribution Units (PDUs) by communicating with the PDU firmware.

RESTful application programming interface

Representational State Transfer or REST is a software architectural style used within the World Wide Web. REST architectures are commonly used for many IT solutions, including the definition of web-based APIs. Systems that adhere to REST practices are often referred to as RESTful interfaces. RESTful interfaces use the HTTP methods—GET, POST, DELETE, and so on—that web browsers use to access web pages.

OData

OData is an open protocol standard for the definition and exchange of information using RESTful APIs. When implementing a common interface across multiple vendors, it is important to standardize the data formats. Standardizing the data formats ensure that the data structures remain interchangeable between different manufacturers.

JSON data

Console API represents data using JSON. JSON is a lightweight data-interchange format that is readable and can also be easily parsed by machines. JSON is based on a subset of the JavaScript Programming Language. JSON uses a text format that is language independent but uses conventions familiar to programmers of the C-family of languages such as C, C++, C#, Java, JavaScript, PERL, and Python. These properties make JSON an ideal data-interchange language.

HTTPS communication

The Hypertext Transfer Protocol or HTTP is an application protocol for distributed, collaborative, hypermedia information systems. HTTP forms the foundation of data communication for the World Wide Web. Secure HTTP (HTTPS) is a secure version of HTTP where it operates within a network connection encrypted by TLS or SSL. By enforcing HTTPS, the security of console management is significantly enhanced.

3

Key integration concepts

This section describes the key integration concepts that are applicable to the use cases and resource model.

Topics:

Base URI

REST API tree structure

The console software provides a web-based hypermedia driven API using a simple folder structure. The client that is provided with the Service Root URI navigates through the entire resource tree through the links in the response payload on the individual URIs. The following schemes are provided for the Service Root URI:

- OEM Scheme
- Redfish Scheme

OEM scheme

```
https://[IP or DNS name]/api
```

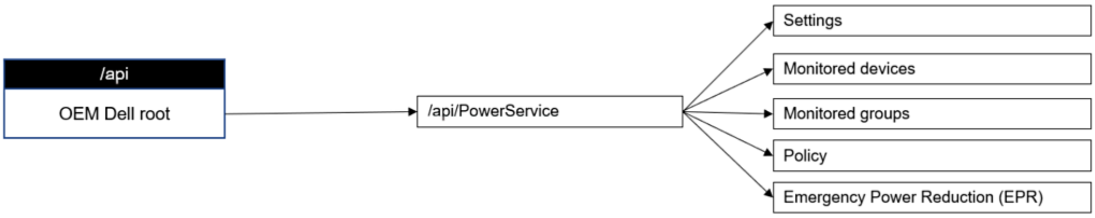

Figure 1. Resource tree for Power Manager—Modular OEM scheme

Resource model

```
https://[IP or DNS name]/api/$metadata
```

Security

To improve security, only HTTPS is supported. You can also update the SSL self-signed certificate with a custom certificate. For example, you can upload a PKCS-12 certificate or sign an application-generated Certificate Signing Request (CSR).

Authentication

Several common schemes are available for enabling authentication of REST requests. Basic Authentication and X-Auth-Token Authentication are some of the common schemes.

Basic authentication

The authorization header in the request has the base-64 encoding of the credentials—username and password. If you do not provide the credentials, a 401—Authorization Failure error is returned. Basic Authentication is supported only when SSL/TLS is used for the transport.

X-Auth-Token authentication

X-Auth-Token Authentication provides a more secure implementation. To establish a session, perform a POST operation on the `SessionService` REST API.

HTTP Method	POST
API	https://[IP Address]/api/SessionService/Sessions

```
Input:
{
    "UserName":"root",
    "Password":"linux",
    "SessionType":"API"
}
```

The X-Auth-Token is available in the header that is returned.

```
connection →Keep-Alive
content-length →268
content-type →application/json; odata.metadata=minimal
date →Tue, 05 Sep 2017 11:55:29 GMT
keep-alive →timeout=5, max=150
location →/api/SessionService/Sessions('e1817fe6-97e5-4ea0-88a9-d865c7302152')
odata-version →4.0
server →Apache
x-auth-token →13bc3f63-9376-44dc-a09f-3a94591a7c5d
x-frame-options →DENY
```

This X-Auth-Token is used in the header for subsequent REST operations and to authenticate the user.

Authorization

Table 1. Authentication and authorization requirements

Actions	Authentication required	Authorization required
Read operation on any instrumentation data	Y	Y
Modify instrumentation data	Y	Y
Invoke actions	Y	Y
View service root	N	N
View metadata document	N	N
View OData service document	N	N
View message registry	Y	N
View Redfish version URI	N	N
View JSONSchemaFile resource URI	N/A	N/A
View JSON schemas URI	N/A	N/A

OME

Table 2. Roles and associated privileges for OME

Privilege	Roles		
	ADMINISTRATOR	DEVICE_MANAGER	VIEWER
APPLIANCE_SETUP	Y		
BASELINE_MANAGEMENT	Y		
DEVICE_CONFIGURATION	Y	Y	
DEVICE_UPDATE	Y	Y	
DISCOVERY_MANAGEMENT	Y		
INVENTORY_MANAGEMENT	Y		
GROUP_MANAGEMENT	Y		
JOB_MANAGEMENT	Y	Y	
MONITORING_SETUP	Y	Y	
OPERATING_SYSTEM_DEPLOYMENT	Y	Y	
POWER_CONTROL	Y	Y	
POWER_MANAGEMENT	Y	Y	
REPORT_MANAGEMENT	Y	Y	
REPORT_RUN	Y	Y	Y
SECURITY_SETUP	Y		
TEMPLATE_MANAGEMENT	Y	Y	
TRAP_MANAGEMENT	Y		
VIEW	Y	Y	Y

Resource addressing

Each managed resource must be uniquely addressable using a distinct URI. The URI syntax must be intuitive and should indicate the relationships with a parent resource.

There are several ways to address specific resources either as instances of specific resource classes or within an associated parent context.

Following is the format of a resource URI:

```
<BASE_URI>/<resource class identifier>/<resource instance identifier>
```

Following is an example of a resource URI, where 1234 is the unique identifier for the device:

```
<BASE_URI>/DeviceService/Devices(1234)
```

Resource operations

Standard HTTP methods are used for creating, retrieving, updating, and deleting resources. The mapping of the HTTP methods to operational semantics is described in the following table:

Table 3. HTTP methods

HTTP method	Description	Example
GET	Used for retrieving the resource representation. This method does not modify the resource across repeated invocations. The query parameters are appended to the URI to appropriately filter the resource instances.	Retrieve logs with dates only after 2016-11-15 16:35:39.820. `<BASE_URI>/ApplicationService/AuditLogs?$filter=CreatedDate gt '2016-11-15 16:35:39.820'`
POST	Used for creating resources or performing actions.	Create a user session. Payload is not displayed. `<BASE_URI>/SessionService/Sessions`
PUT	Used for updating a specific instance or create a specific resource instance with a specific identifier.	Update the user account details. Payload is not displayed. `<BASE_URI>/AccountService/Accounts('1414')`
DELETE	Used for removing a specific resource. If the resource does not exist, a Success response is returned.	Delete a specific user account. `<BASE_URI>/AccountService/Accounts('1414')`

When you perform tasks using these methods, they return an HTTP response code. For more information about the response codes, see Status Code Definitions on the W3.org website. If a method fails, you may also see an Error and Event Message code. You can search for more details about this code using the Dell QRL app on your mobile device.

Security considerations

For performing update and delete operations, you must be authorized to perform the operations on the resource. If you do not have the required permission, an *Unauthorized* error is returned.

Data filtering

The console software provides filtering options on certain URIs that return a collection of entities. Using this feature, the clients can extract a selected set of records using comparison operators on attributes of the model entity behind the collection. The relevant URI sections contain the information about the attributes and the operators that support filtering.

```
<Collection API>?$filter=<attribute> <operator> <value>&<attribute> <operator> <value>
```

Examples:

Filter devices by system ID	`/api/DeviceService/Devices?$filter=SystemId eq 123`
Filter devices by model using equal operator	`/api/DeviceService/Devices?$filter=Model eq 'PowerEdge MX740c'`
Filter devices by model using contains operator	`/api/PowerService/MonitoredDevices?$filter=contains(Model,'PowerEdge')`
Filter devices by device type	`/api/DeviceService/Devices?$filter=Type eq 3000 or Type eq 5000` **NOTE:** For enumeration of Type, see `/api/DeviceService/DeviceType`.
Adding combinations of filters	`/api/PowerService/MonitoredDevices?$filter=contains(Model,'PowerEdge') and Type eq 2000`

For more information about `$filter`, see OData Version 4.0 Part 1: Protocol at docs.oasis-open.org.

Data sorting

The console software provides sorting options on certain URIs that return a collection of entities. The sorting options enable the clients to get sorted results using the Sort operators on certain attributes of the model entity behind the collection. The relevant URI section contain information about the attributes that support sorting.

```
<Collection API>?$orderby=<attribute> <asc/desc>
```

Examples: Get devices sorted in ascending order by Status

```
/api/DeviceService/Devices?$orderby=Status asc
```

NOTE: For enumeration of Status, see `/api/DeviceService/DeviceStatuses`.

NOTE: For more information about `$orderby`, see OData Version 4.0 Part 1: Protocol at docs.oasis-open.org.

Data pagination

The console software provides pagination options on certain URIs that return a collection of entities. The pagination options enable the clients to get paginated results. If a URI supports pagination, the relevant URI sections indicate it.

```
<Collection API>?$top=<required number of records>&$skip=<number of records to skip>
```

Examples:

Get 5th to 8th devices from the Device collection	`/api/DeviceService/Devices?$skip=4&$top=4`

Get first four devices from the Device collection	`/api/DeviceService/Devices?$top=4`

NOTE: For more information about `$skip` and `$top`, see OData Version 4.0 Part 1: Protocol at docs.oasis-open.org.

Request headers

The request header represents headers in the client HTTPS request that are used to communicate client-preferences to the service end point. The service provides the supported preference in the response header.

The following table lists a few examples of request headers:

Table 4. Examples of request headers

Request Header	Description	Example
x-auth-token	Authentication token taken from the return header of the `SessionService/Sessions` POST operation to create a session.	x-auth-token: d6399a19-38c8-467b-a1ec-75ffa03efb7c
Accept-Language	Choice of language that the client can request—Optional.	Accept-Language: en

Response codes

For synchronous operations, the server returns HTTP response codes 200 or 204. For operations that take a long time, the server returns a status code of 202 along with an HTTP response header (Location). This response corresponds to the URI of the temporary resource that can be used to monitor the operation.

The following table lists some of the response codes:

Table 5. HTTP response codes

Request	Response code
Success codes	
GET	● 200 – OK with message body ● 204 – OK with no message body ● 206 – OK with partial message body
POST	● 201 – Resource created (operation complete) ● 202 – Resource accepted (operation pending)
PUT	● 202 – Accepted (operation pending) ● 204 – Success (operation complete)
DELETE	● 202 – Accepted (operation pending) ● 204 – Success (operation complete)
Failure codes	
Invalid parameter	400 - Invalid parameter
Authorization	401 - Authorization failure
Permission denied	403 - Permission denied
Not found	404 - Resource not found
Invalid request method	405 - Invalid request method
Internal server error	500 - Internal server error

Table 5. HTTP response codes (continued)

Request	Response code
Service unavailable	503 - Service unavailable

For more information about response codes, see Status Code Definitions on the W3.org website.

Response headers

The following table lists a few examples of response headers:

Table 6. Examples of response headers

Response Header	Description	Example
Connection	Control options for the current connection and list of hop-by-hop request fields.	Connection: Keep-Alive
Content-Type	Specifies the format of the content that the server returns. If there are multiple formats that can be accepted in the client request (using the Accept header), the server chooses the appropriate supported format.	Content-Type: application/json; odata.metadata=minimal
Keep-alive	● *Timeout* header parameter indicates the time that a connection is allowed to remain idle before it is closed. ● *Max* header parameter indicates the maximum number of requests that are permitted before the connection is closed.	Timeout=5; max=150
Content-length	The length of the request body in 8-bit bytes or octets.	Content-Length: 348
date	The date and time that the message originated, in HTTP-date format as defined by RFC 7231 Date/Time Formats.	date: Thu, 02 Apr 2009 11:11:28 GMT
Odata-version	The version of Odata that is used.	Odata: 4.0
Location	Used in redirection or when a new resource is created.	Location: <BASE_URI>/SessionService/Sessions('3204bb9d-409d-4bd9-8a5f-d44005c81a2c')
Server	A name for the server.	Server: Apache
x-frame-options	Clickjacking protection: ● deny – no rendering within a frame ● sameorigin – no rendering if origin mismatch ● allow-from – allow from specified location ● allowall – non-standard, allow from any location	DENY

4

Settings

Topics:

- /api/PowerService/Settings
- /api/PowerService/Actions/PowerService.UpdateSettings

/api/PowerService/Settings

This URI represents the list of power settings.

Supported versions of Power Manager:

- 2.0
- 1.2
- 1.1
- 1.0

GET method for Power Service Settings

To get power and temperature monitoring units, metric gathering interval, date range configuration for **Top Energy Consumers** widgets and **Report** settings.

Description	This method returns the list of settings that are configured.
Privilege	VIEW
HTTP response codes	200

Example

```
Input: None
Output:
{
    "@odata.context": "/api/$metadata#Collection(PowerService.Settings)",
    "@odata.count": 9,
    "value": [
        {
            "@odata.type": "#PowerService.Settings",
            "Id": 1,
            "Name": "TEMPERATURE_DISPLAY_UNIT",
            "DefaultValue": 1,
            "Value": 1

        },
        {
            "@odata.type": "#PowerService.Settings",
            "Id": 2,
            "Name": "POWER_DISPLAY_UNIT",
            "DefaultValue": 1,
            "Value": 1
        },
        {
            "@odata.type": "#PowerService.Settings",
            "Id": 3,
            "Name": "METRIC_GATHERING_INTERVAL",
            "DefaultValue": 15,
            "Value": 15
        },
        {
            "@odata.type": "#PowerService.Settings",
            "Id": 5,
            "Name": "BUILT_IN_REPORT_TIME_INTERVAL",
            "DefaultValue": 90,
            "Value": 90
        },
        {
            "@odata.type": "#PowerService.Settings",
            "Id": 6,
            "Name": "BUILT_IN_REPORT_TIME_GRANULARITY",
            "DefaultValue": 2,
            "Value": 2
        },
        {
            "@odata.type": "#PowerService.Settings",
            "Id": 7,
            "Name": "TOP_ENERGY_CONSUMERS_DURATION","DefaultValue": 8,
            "Value": 8
        },
        {
            "@odata.type": "#PowerService.Settings",
            "Id": 8,
            "Name": "DELETE_METRIC_DATA",
            "DefaultValue": 2,
            "Value": 2
        }
    ]
}
```

Table 7. Attributes

Attribute Name	Description
Id	Power Settings record ID.
Name	Name associated with the power settings name.
Default Value	Default values associated with Power Manager.
Value	Value associated with Power Manager Settings.

/api/PowerService/Actions/PowerService.UpdateSettings

Use this URI to configure Power Manager Settings.

Supported versions of Power Manager:

- 2.0
- 1.2
- 1.1
- 1.0

POST method for Power Service Settings

This method is to set power and temperature monitoring units, metric gathering interval, date range configuration for Top Energy Consumers widgets, Report settings.

Description To save power settings values for corresponding settings.

Privilege APPLIANCE_SETUP

HTTP response codes 200

Example

```
Input:
{
  "Settings":[
  {
       "Id": 1,
    "Value": 1
  },
  {
       "Id": 2,
       "Value": 1
   },
   {
        "Id": 3,
        "Value": 15
    },
    {
         "Id": 5,
         "Value": 90
     },
     {
          "Id": 6,
          "Value": 2
     },
     {
          "Id": 7,
          "Value": 8
     },
     {
          "Id": 8,
          "Value": 1
     }
            ]
}

Output:
{
"Status": true
}
```

Table 8. Power Settings Enumeration

Enumeration Value	Description
1	Temperature Display Unit
2	Power Display Unit
3	Metric Gathering Interval
5	Built-in Report Time Interval
6	Built-in report Time Granularity
7	Top Energy Consumers Duration
8	Delete Metric Data

Table 9. Temperature Display Unit Enumeration

Enumeration Value	Description
1	Celsius
2	Fahrenheit

Table 10. Power Display Unit Enumeration

Enumeration Value	Description
1	Watt
2	BTU/Hr

Table 11. Metric Gathering Interval Enumeration

Enumeration Value	Description
15	15 Minutes
30	30 Minutes
60	60 Minutes

Table 12. Built-in Report Time Interval Enumeration

Enumeration Value	Description
1	1 Day
7	7 Days
15	15 Days
30	30 Days
90	90 Days
180	180 Days
365	365 Days

Table 13. Built In Report Time Granularity Enumeration

Enumeration Value	Description
1	1 Hour
2	1 Day

Table 14. Top Energy Consumers Enumeration

Enumeration Value	Description
4	1 Day
5	1 Week
6	2 Weeks
7	1 Month
8	3 Months
9	6 Months
10	1 Year

Table 15. Delete Metric Data Enumeration

Enumeration Value	Description
1	Delete data
2	Keep data

5

Monitored Devices

Topics:

/api/PowerService/Actions/PowerService.AddDevices

This URI is to add the devices to Power Manager.

Supported versions of Power Manager:

- 2.0
- 1.2
- 1.1
- 1.0

POST method to add Devices

This method adds device(s) to Power Manager.

Description Adds device(s) to Power Manager.

Privilege DEVICE_CONFIGURATION

HTTP response codes 200

Example

```
Input:
{
"Ids": [10128,10127]
}
Output:
{
"Status": "SUCCESS"
}
```

Table 16. Attributes

Attribute Name	Description
Id	Device Id(s) to be monitored by the Power Manager

/api/PowerService/Actions/PowerService.RemoveDevices

This URI removes the device(s) from Power Manager monitored devices.

Supported versions of Power Manager:

- 2.0
- 1.2
- 1.1
- 1.0

POST method to remove Monitored Devices

This method removes device(s) from Power Manager.

Description Removes device(s) from Power Manager.

Privilege DEVICE_CONFIGURATION

HTTP response codes 200

Example

```
Input:
{
"Ids": [10128,10127]
}
Output:
{
"Status": "SUCCESS"
}
```

Table 17. Attributes

Attribute Name	Description
Id	Device Id(s) to be removed from the Power Manager

NOTE: To remove all devices from Power Manager, provide value as **-1**.

/api/PowerService/MonitoredDevices

This URI represents the devices that are monitored by Power Manager for collection and monitoring power and thermal data.

Table 18. Filters

Filter Name	Description
Top	Required number of records
Skip	Number of records to skip. Default value is zero.

Supported versions of Power Manager:

- 2.0
- 1.2
- 1.1
- 1.0

GET method for Monitored Devices

This method returns monitored device details.

Description Returns the monitored device details.

Privilege VIEW

HTTP response codes 200

Example

```
Input: None
Output:
{
    "@odata.context": "/api/
$metadata#Collection(PowerService.MonitoredDevice)",
    "@odata.count": 1,
    "value": [
        {
            "@odata.type": "#PowerService.MonitoredDevice",
            "@odata.id": "/api/PowerService/MonitoredDevices(10079)",
            "Id": 10079,
            "DeviceName": "100.96.27.67",
            "ServiceTag": "DSFYSF2",
            "Type": 1000,
            "Model": "PowerEdge M630",
            "PowerState": 17,
            "ConnectionState": true,
            "HealthStatus": 1000,
            "ManagedState": 3000,
            "AddedOn": "2019-06-25 12:55:34.107889",
            "IsPartOfGroup": false,
            "Location": "DC/Room/Aisle/Rack 1",
            "SlotNumber": 10,
            "GroupAssociation": [
                {
                    "Id": 11359,
                    "Name": "W1"
                }
            ],
            "LicenseStatus": 1,
            "PowerPolicyCapable": true,
            "AllDevices": true,
            "EPRStatus": {
                "@odata.id": "/api/PowerService/MonitoredDevices(10079)/
EPRStatus"
            },
            "Policies@odata.navigationLink": "/api/PowerService/
MonitoredDevices(10079)/Policies"
        }
    ]
}
```

Table 19. Attributes

Attribute Name	Description
Id	Device Id
DeviceName	Device Name
ServiceTag	Service Tag of the device.
Type	Device Type
Model	Device Model
PowerState	Power State of the device.

Table 19. Attributes (continued)

Attribute Name	Description
ConnectionState	Connection State of the device
HealthStatus	Health Status of the device
ManagedState	Managed State of the device
AddedOn	Date when the device is added.
GroupAssociation	Group to which the device belongs to.
GroupAssociation.Id	Group Id of the group to which the device belongs to.
GroupAssociation.Name	Group Name device belongs to.
LicenseStatus	License status of the device.
PowerPolicyCapable	Device is Policy capable or not.
AllDevices	Filter attribute to get all devices or individual devices.
IsPartOfGroup	Device is part of group or not.

License enumerations are defined in following table.

Table 20. License Status Enumeration

Enumeration Value	Description
1	Active
2	Expired
3	Not Applicable

Table 21. Filters

Field	Operator
Id	eq
DeviceName	contains
ServiceTag	contains
Model	contains
Type	eq
PowerState	eq
ManagedState	eq
ConnectionState	eq
HealthStatus	eq
AllDevices	eq
LicenceStatus	eq
IsPartOfGroup	eq

List of all the sort options that this method supports:

- Id
- DeviceName
- ServiceTag
- Model
- Type
- PowerState

- ManagedState
- ConnectionState
- HealthStatus
- AddedOn
- IsPartOfGroup
- Location
- SlotNumber

Table 22. Enumeration

Type	State	Enumeration Value
Health	Critical	4000
	Warning	3000
	Normal	1000
	Unknown	2000
Power	Online	17
	Offline	18
	Unknown	1
Connection	Connected	true
	Disconnected	false
Device Type	Compute	1000
	Chassis	2000
Managed	Error	1000
	Managed	3000
	Monitored	4000
	Managed with Alerts	6000
	Proxied	7000

/api/JobService/Jobs

This URI refreshes the list of devices for Power Manager capabilities.

Supported versions of Power Manager:

- 2.0
- 1.2
- 1.1
- 1.0

POST method to refresh device list compatible with Power Manager

Use this method to refresh the device list for Power Manager capabilities.

Description View and add the devices that are supported and compatible with Power Manager by refreshing the list for Power Manager capabilities.

Privilege GROUP_MANAGEMENT

HTTP response codes

201

Example

```
Input:
{
    "JobName": "WorkingSet Management Task",
    "JobDescription": "WorkingSet Management Task - User Triggered",
    "Schedule": "startnow",
    "State": "Enabled",
    "JobType": {
        "Id": 306,
        "Name": "WorkingSet_Management_Task"
    },
    "Params": [
        {
            "Key": "23",
            "Value": ""
        }
    ],
    "Targets": [
        {
            "Id": 500,
            "Data": "",
            "TargetType": {
                "Id": 6000,
                "Name": "GROUP"
            }
        }
    ]
}

Output:
{
    "Id": 10219,
    "JobName": "WorkingSet Management Task",
    "JobDescription": "WorkingSet Management Task - User Triggered",
    "NextRun": null,
    "LastRun": null,
    "StartTime": null,
    "EndTime": null,
    "Schedule": "startnow",
    "State": "Enabled",
    "CreatedBy": "admin",
    "UpdatedBy": null,
    "LastRunStatus": {
        "Id": 2200,
        "Name": "NotRun"
    },
    "JobType": {
        "Id": 306,
        "Name": "WorkingSet_Management_Task",
        "Internal": true
    },
    "JobStatus": {
        "Id": 2080,
        "Name": "New"
    },
    "Targets": [
        {
            "JobId": 10219,
            "Id": 500,
            "Data": null,
            "TargetType": {
                "Id": 6000,
                "Name": "GROUP"
            }
        }
    ],
    "Params": [
        {
            "JobId": 10219,
            "Key": "23",
            "Value": ""
        }
    ],
    "Visible": true,
    "Editable": true,
    "Builtin": false,
    "UserGenerated": true,
    "IdUserOwner": 10068
}
```

/api/PowerService/DeviceStatistics

URI to represent total power manager capable and power manager monitored devices.

Supported versions of Power Manager:

- 2.0

GET method to list device statistics

This method represents the Power Manager capable, and Power Manager monitored devices.

Description This method returns the Power Manager capable, and Power Manager monitored devices.

Privilege VIEW

HTTP response codes 200

Example

```
Input: None
Output:
{
    "@odata.context": "/api/$metadata#PowerService.DeviceStatistic",
    "@odata.type": "#PowerService.DeviceStatistic",
    "@odata.id": "/api/PowerService/DeviceStatistics",
    "CapableDevices": 20,
    "MonitoredDevices": 20
}
```

Table 23. Attributes

Attributes	Description
CapableDevices	Total number of power manager capable devices.
MonitoredDevices	Total number of power manager monitored devices

6

Monitored Groups

Topics:

/api/PowerService/Actions/PowerService.AddGroups

This URI adds the group(s) to Power Manager.

Supported versions of Power Manager:

- 2.0
- 1.2
- 1.1
- 1.0

POST method to add Groups

This method adds group(s) to Power Manager.

Description Adds group(s) to Power Manager.

Privilege GROUP_MANAGEMENT

HTTP response codes 200

Example

```
Input:
{
"Ids": [10131,10132]
}
Output:
{
"Status": "SUCCESS"
}
```

Table 24. Attributes

Attribute Name	Description
Id	Group Id(s) to be added

/api/PowerService/Actions/PowerService.RemoveGroups

This URI removes the group(s) from Power Manager.

Supported versions of Power Manager:

- 2.0
- 1.2
- 1.1
- 1.0

POST method to remove Monitored Groups

This method removes group(s) from Power Manager.

Description Removes Group(s) from Power Manager

Privilege GROUP_MANAGEMENT

HTTP response codes 200

Example

```
Input:
{
"Ids": [10131,10132]
}
Output:
{
"Status": "SUCCESS"
}
```

Table 25. Attributes

Attribute Name	Description
Ids	Device Id(s) to be removed.

NOTE: To remove all groups from the Power Manager, provide the value as **-1**.

/api/PowerService/CapableGroups

This URI represents the Power Manager monitor capable groups.

Table 26. Filters

Filter Name	Description
Top	Required number of records
Skip	Number of records to skip. Default value is zero.

Supported versions of Power Manager:

- 2.0
- 1.2
- 1.1
- 1.0

GET method for Representing Capable Groups

This method represents the capable groups.

Description Returns a list of capable groups.

Privilege VIEW

HTTP response codes 200

Example

```
Input: None
Output:
{
"@odata.context":"/api/$metadata#Collection(PowerService.CapableGroups)",
"@odata.count": 1,
"value": [
    {
    "@odata.type": "#PowerService.CapableGroups",
    "@odata.id": /api/PowerService/CapableGroups(10690)",
    "Id": 10690,
    "ParentId": 10689,
    "Name": "Aisle - A",
    "Description": "Aisle - A",
    "DeviceCount": 7,
    "PmPluginDeviceCount": 7,
    "Devices@odata.navigationLink":"/api/PowerService/CapableGroups
(10690)/Devices"
    }
  ]
}
```

Table 27. Attributes

Attribute name	Description
Id	Group ID
ParentId	Group Parent ID
Name	Group Name
Description	Description of the group
DeviceCount	Total device in the group
PmPluginDeviceCount	Total Power Manager capable devices in group

/api/PowerService/CapableGroups(<Group ID>)/Devices

This URI represents the device from a capable group

Table 28. Filters

Filter Name	Description
Top	Required number of records
Skip	Number of records to skip. Default value is zero.

Supported versions of Power Manager:

- 2.0
- 1.2
- 1.1

- 1.0

GET method for Representing All Devices from Capable Group

This method represents all the devices from capable group.

Description Returns all the devices belonging to a capable group.

Privilege VIEW

HTTP response codes 200

Example

```
Input: None
Output:
{
"@odata.context": "/api/$metadata#Collection(PowerService.Device)",
   "@odata.count": 1,
     "value": [
        {
            "@odata.type": "#PowerService.Device",
            "Id": 10069,
            "DeviceName": "SKCLUSN1.tejd.bdcsv.lab",
            "ServiceTag": "4xBzewP",
            "Type": 1000,
            "PowerState": 17,
            "ManagedState": 3000,
            "ConnectionState": true,
            "HealthStatus": 4000,
            "Model": "PowerEdge R940",
            "InWorkingSet": true,
            "IsPolicyCapable": true
            "IsAddedIndividually": true
        }
]
}
```

Table 29. Attributes

Attribute name	Description
Id	Device ID
DeviceName	Device Name
ServiceTag	Device Service tag
Type	Device Type
PowerState	Power State of device
ManagedState	Managed State of the device
ConnectionState	Connection State of the device
HealthStatus	Health Status of the device
Model	Device Model
InWorkingSet	Available in Power Manager monitored list or not.
IsPolicyCapable	Capable to apply policy or not.
IsAddedIndividually	Added as individual device for monitoring or not.

Table 30. Filters

Field	Operator
Id	eq

Table 30. Filters (continued)

Field	Operator
DeviceName	contains
ServiceTag	contains
Model	contains
Type	eq
PowerState	eq
ManagedState	eq
ConnectionState	eq
HealthStatus	eq
IsAddedIndividually	eq

List of all the sort options that this method supports:

- Id
- DeviceName
- ServiceTag
- Model
- Type
- PowerState
- ManagedState
- ConnectionState
- HealthStatus
- IsAddedIndividually

/api/PowerService/Groups(<Group ID>)/Devices

This URI represents all the Power Manager capable devices.

Supported versions of Power Manager:

- 2.0
- 1.2
- 1.1
- 1.0

GET method for Representing All Power Manager capable devices from a Group

This method represents all the Power Manager supported devices (All Licensed, Power Monitor Capable and Supported Model)

Description

Returns all the Power Manager devices belonging to a monitored group

Privilege

VIEW

HTTP response codes

200

Example

```
Input: None
Output:
{
    "@odata.context": "/api/$metadata#Collection(PowerService.Device)",
    "@odata.count": 1,
    "value": [
            {
            "@odata.type": "#PowerService.Device",
            "Id": 10069,
            "DeviceName": "SKCLUSN1.tejd.bdcsv.lab",
            "ServiceTag": "4xBzewP",
            "Type": 1000,
            "PowerState": 17,
            "ManagedState": 3000,
            "ConnectionState": true,
            "HealthStatus": 4000,
            "Model": "PowerEdge R940",
            "InWorkingSet": true,
            "IsPolicyCapable": true,
            "IsAddedIndividually": true
            }
        ]
}
```

Table 31. Attributes

Attribute Name	Description
Id	Device Id.
DeviceName	Device Name.
ServiceTag	Device Service Tag.
Type	Device Type.
PowerState	Power State of device.
ManagedState	Managed State of the device.
ConnectionState	Connection state of the device.
HealthStatus	Health status of the device.
Model	Device Model.
InWorkingSet	Available in monitored list.
IsPolicyCapable	Capable to apply Power cap policy or not.
IsAddedIndividually	Device is added individually or not.

Table 32. Filter

Field	Operator
Id	eq
DeviceName	Contains
ServiceTag	Contains
Model	Contains
Type	eq
PowerState	eq
ManagedState	eq
ConnectionState	eq
HealthStatus	eq
IsAddedIndividually	eq

List of all the sort options that this method supports:

- Id
- DeviceName
- ServiceTag
- Model
- Type
- PowerState
- ManagedState
- ConnectionState
- HealthStatus
- IsAddedIndividually

/api/PowerService/MonitoredGroups

This URI represents groups which are monitored by the Power Manager.

Table 33. Filters

Filter Name	Description
Top	Required number of records
Skip	Number of records to skip. Default value is zero.

Supported versions of Power Manager:

- 2.0
- 1.2
- 1.1
- 1.0

GET method for Representing Monitored Groups

This method represents all groups which are monitored.

Description Returns all the Power Manager monitored groups.

Privilege VIEW

HTTP response codes 200

Example

```
Input: None
Output:
{
"@odata.context": "/api/
$metadata#Collection(PowerService.MonitoredGroup)",
    "@odata.count": 1,
    "value": [
        {
        "@odata.type": "#PowerService.MonitoredGroup",
      "@odata.id": "/api/PowerService/MonitoredGroups(10690)",
      "Id": 10690,
      "ParentId": 10689,
      "Name": "Aisle - A",
      "Description": "",
      "AddedOn": "2019-06-10 19:05:42.413941",
      "DevicesInWorkingSet": 7,
      "TemperatureTriggeredPolicyExist: false"
      "Type": 1,   //Info: 1 = Static , 2 = Physical
      "Location": "DC/Room/Aisle",
      "Devices@odata.navigationLink": "/api/PowerService/
MonitoredGroups(10690)/Devices",
      "EPRStatus": {
                "@odata.id": "/api/PowerService/MonitoredGroups(10690)/
EPRStatus"
            },
            "Policies@odata.navigationLink": "/api/PowerService/
MonitoredGroups(10690)/Policies"
       }
  ]
}
```

Table 34. Attributes

Attribute name	Description
Id	Group Id
ParentId	Parent group Id
Type	Type of the group
Name	Name of the group
Description	Description of the Group
AddedOn	Date of the group is added.
DevicesInWorkingSet	Number of devices available in monitored list
TemperatureTriggeredPolicyExist	Temperature policy is created or not.

Table 35. Filters

Field	Operator
Id	eq
ParentId	eq
Type	eq
Name	contains
Description	contains

List of all the sort options that this method supports:

- Id
- ParentId
- Type
- Name
- Description
- Location
- DevicesInWorkingSet
- AddedOn

/api/PowerService/MonitoredGroups(<Group ID>)/Devices

This URI represents all the devices of a monitored group

Table 36. Filters

Filter Name	Description
Top	Required number of records
Skip	Number of records to skip. Default value is zero.

Supported versions of Power Manager:

- 2.0
- 1.2
- 1.1
- 1.0

GET method for Representing All Devices from a MonitoredGroup

This method represents all the device of a monitored group.

Description Returns all the devices belongs to a Monitored group.

Privilege VIEW

HTTP response codes 200

Example

```
Input: None
Output:
{
    "@odata.context": "/api/$metadata#Collection(PowerService.Device)",
    "@odata.count": 1,
    "value": [
        {
            "@odata.type": "#PowerService.Device",
            "Id": 10069,
            "DeviceName": "SKCLUSN1.tejd.bdcsv.lab",
            "ServiceTag": "4xBzewP",
            "Type": 1000,
            "PowerState": 17,
            "ManagedState": 3000,
            "ConnectionState": true,
            "HealthStatus": 4000,
            "Model": "PowerEdge R940",
            "InWorkingSet": true,
            "IsPolicyCapable": true
            "IsAddedIndividually": false
        }
    ]
}
```

Table 37. Attributes

Attribute name	Description
Id	Device Id
DeviceName	Device Name
ServiceTag	Device Service Tag
Type	Device Type
PowerState	Power State of device.
ManagedState	Managed State of the device
ConnectionState	Connection state of the device
HealthStatus	Health status of the device
Model	Device Model
InWorkingSet	Available in monitored list
IsPolicyCapable	Capable to apply Power cap policy or not.

Table 37. Attributes (continued)

Attribute name	Description
IsAddedIndividually	Added as individual device for monitoring or not.

7

Power Policy

Topics:

api/PowerService/PowerBounds

This URI represents the device Lower bound and Upper bound.

Supported versions of Power Manager:

- 2.0
- 1.2
- 1.1
- 1.0

POST method for retrieving Power Bounds

This method returns device upper and lower bounds.

Description Returns device upper and lower bounds value.

Privilege VIEW

HTTP response codes 201

Example

```
Input:
{
    "EntityId": [10078]
}
Output:
{
    "LowerBound": 709,
    "UpperBound": 1057
}
```

Table 38. Attributes

Attribute	Description
EntityId	ID of device.
LowerBound	Device Lower power bound.

Table 38. Attributes (continued)

Attribute	Description
UpperBound	Device Upper power bound.

NOTE: Only single entity ID is supported and pass only device ids.

/api/PowerService/Policies

This URI represents the power policies that are created

Table 39. Filters

Filter Name	Description
Top	Required number of records
Skip	Number of records to skip. Default value is zero.

Supported versions of Power Manager:

- 2.0
- 1.2
- 1.1
- 1.0

GET method for retrieving Policies

This method returns all the power policies.

Description Returns all the power policies.

Privilege VIEW

HTTP response codes 200

Example

```
Input: None
Output:
{
    "@odata.context": "/api/
$metadata#Collection(PowerService.PowerPolicy)",
    "@odata.count": 2,
    "value": [
        {
            "@odata.type": "#PowerService.PowerPolicy",
            "@odata.id": "/api/PowerService/Policies(32)",
            "PolicyId": 32,
            "Type": 1,
            "Name": "Chassis Policy",
            "Description": "Power policy created on chassis",
            "Enabled": true,
            "AssignedTo": "D89RG52",
            "CreatedTime": "2019-06-25 13:49:59.342645",
            "ExecutionState": 3,
            "IsAssociatedToGroup": false
            "PolicyDetails": {
                "@odata.id": "/api/PowerService/Policies(32)/
PolicyDetails"
            }
        },
        {
            "@odata.type": "#PowerService.PowerPolicy",
            "@odata.id": "/api/PowerService/Policies(30)",
            "PolicyId": 30,
            "Type": 1,
            "Name": "Group Policy",
            "Description": "Updated policy on group",
            "Enabled": true,
            "AssignedTo": "DeviceGroup",
            "CreatedTime": "2019-06-19 12:07:23.935816",
            "ExecutionState": 3,
            "IsAssociatedToGroup": false
            "PolicyDetails": {
                "@odata.id": "/api/PowerService/Policies(30)/
PolicyDetails"
            }
        }
    ]
}
```

Table 40. Attributes

Attribute Name	Description
PolicyId	ID of policy
Name	Name associated with the policy.
Type	Type of policy.
Description	Description associated with the policy
Enabled	Indicates if the policy is enabled or disabled
AssignedTo	Entity on which the policy is assigned to.
CreatedTime	Time at which the policy is created.
ExecutionState	State associated with the policy.
PolicyDetails	Link to the detailed view of the policy.
IsAssociatedToGroup	The policy is associated to group or not.

Table 41. Policy Execution State Enumeration

Enumeration Value	Description
1	NOSTATE (Policy execution state is not yet determined).
2	EXECUTING (Policy execution is in progress or pending).
3	SUCCESS (Policy execution is successful).
5	FAILED (Policy execution failed).

Table 42. Filters

Field	Operator
Name	contains
Description	contains
Enabled	eq
AssignedTo	contains
Type	eq

Table 43. Type

Enumeration Value	Description
1	Static policy

Table 43. Type (continued)

Enumeration Value	Description
2	Temperature-triggered policy

List of all the sort options that this method supports:

- PolicyId
- Name
- Description
- Enabled
- AssignedTo
- CreatedTime
- Type
- IsAssociatedToGroup

/api/PowerService/Policies(<PolicyId>)

This URI represents a specific policy.

Supported versions of Power Manager:

- 2.0
- 1.2
- 1.1
- 1.0

GET method for Policies(<PolicyId>)

This method returns a specific policy.

Description	Returns details of a specific policy.
Privilege	VIEW
HTTP response codes	200

Example

```
Input: None

Output:
{
    "@odata.context": "/api/
$metadata#Collection(PowerService.PowerPolicy)",
    "@odata.count": 1,
    "value": [
        {
            "@odata.type": "#PowerService.PowerPolicy",
            "@odata.id": "/api/PowerService/Policies(1)",
            "PolicyId": 4,

        "Type": 1,

             "Name": "Policy for Power",
            "Description": "Policy for Power",
            "Enabled": true,
            "AssignedTo": "4xBzewP",
            "CreatedTime": "2019-06-12 16:54:14.411943",
            "ExecutionState": 3,
            "IsAssociatedToGroup": false
            "PolicyDetails": {
                "@odata.id": "/api/PowerService/Policies(4)/
PolicyDetails"
            }
```

```
         }
     ]
}
```

Table 44. Attributes

Attribute Name	Description
PolicyId	ID of policy
Name	Name associated with the policy.
Type	Type of policy.
Description	Description associated with the policy
Enabled	Indicates if the policy is enabled or disabled.
AssignedTo	Entity on which the policy is assigned to.
CreatedTime	Time at which the policy is created.
ExecutionState	State associated with the policy.
PolicyDetails	Link to the detailed view of the policy.
IsAssociatedToGroup	The policy is associated to group or not.

Table 45. Policy Execution State Enumeration

Enumeration Value	Description
1	NOSTATE (Policy execution state is not yet determined).
2	EXECUTING (Policy execution is in progress or pending).
3	SUCCESS (Policy execution is successful).
5	FAILED (Policy execution failed).

/api/PowerService/Policies(<PolicyId>)/PolicyDetails

This method returns details of a power policy.

Supported versions of Power Manager:

- 2.0
- 1.2
- 1.1
- 1.0

GET method for Policies(<PolicyId>)/PolicyDetails of a specific policy

Description Returns details of a power policy.

Privilege VIEW

HTTP response codes 200

Example

```
Input: None

Output:
```

```
{
    "@odata.context": "/api/$metadata#PowerService.PowerPolicyModel",
    "@odata.type": "#PowerService.PowerPolicyModel",
    "@odata.id": "/api/PowerService/Policies(30)/PolicyDetails",
    "PolicyId": 30,
    "Name": "Group Policy",
    "Type": 1,
    "Description": "Power policy created on group",
    "Enabled": true,
    "Schedule": {
        "StartTime": "",
        "EndTime": "",
        "StartDate": "",
        "EndDate": "",
        "DaysOfTheWeek": []
    },
    "Devices": [
        {
            "Id": 10070,
            "DeviceName": "100.96.25.126",
            "Type": 2000,
            "ServiceTag": "D89RG52",
            "PowerCapValue": 16670,
            "Ip": null
        }
    ],
    "Group": {
        "Id": 10132,
        "PowerCapValue": 16670,
        "TemperatureThreshold": 45,
        "Name": "DeviceGroup",
        "Description": ""
    }
}
```

Table 46. Attributes

Attribute name	Description
PolicyId	ID of power policy.
Name	Name associated with the policy.
Type	Type of policy.
Description	Description associated with the policy.
Enabled	Indicates if the policy is enabled or disabled.
Schedule	Indicates when the policy will be active.
StartTime	Time at which the policy interval starts.
EndTime	Time at which the policy interval ends.
StartDate	Date from which the policy is active.
EndDate	Date until which the policy is active.
DaysOfTheWeek	Days of week in which the policy is active.
Devices	Target devices of the policy.
Devices.Id	ID of the target device.
Devices.DeviceName	Name of the device.
Devices.Type	Type of the device.
Devices.ServiceTag	Service tag of the device.

Table 46. Attributes (continued)

Attribute name	Description
Devices.PowerCapValue	Power cap value for the device.
Devices.Ip	IP address of the device.
Group	Target group of the policy (if applicable).
Group.Id	ID of the group.
Group.PowerCapValue	Power cap value for the group.
Group.TemperatureThreshold	Temperature threshold value for the group.
Group.Name	Name of the group.
Group.Description	Description of the group.

NOTE: The output Group object will have data only when the policy is created for group.

/api/PowerService/MonitoredDevices(<DeviceId>)/Policies

This method returns policies that are created on the specific device.

Supported versions of Power Manager:

- 2.0
- 1.2
- 1.1
- 1.0

GET method for policies created on specified devices

This URI returns the power policies that are created on the specified device.

Description	Returns all the power policies.
Privilege	VIEW
HTTP response codes	200

Example

```
Input: None

Output:
{
    "@odata.context": "/api/$metadata#Collection(PowerService.Policy)",
    "@odata.count": 2,
    "value": [
        {
            "@odata.type": "#PowerService.Policy",
            "Id": 32,
            "Type": 1,
            "PolicyName": "Chassis Policy",
            "PolicyDescription": "Power policy created on chassis",
            "PolicyState": true
        },
        {
            "@odata.type": "#PowerService.Policy",
            "Id": 30,
            "Type": 1,
            "PolicyName": "Group Policy",
            "PolicyDescription": "Updated policy on group",
            "PolicyState": true
        }
    ]
}
```

Table 47. Attributes

Attribute name	Description
Id	ID of the policy
Type	Type of the policy.
PolicyName	Name associated with the policy.
PolicyDescription	Description associated with the policy
PolicyState	Indicates if the policy is enabled or disabled

/api/PowerService/MonitoredGroups(<GroupId>)/Policies

This URI returns the power policies that are created on the specified group.

Supported versions of Power Manager:

- 2.0
- 1.2
- 1.1
- 1.0

GET method for policies created on specified groups

Description Returns all the power policies.

Privilege VIEW

HTTP response codes 200

Example

```
Input: None

Output:
{
    "@odata.context": "/api/$metadata#Collection(PowerService.Policy)",
    "@odata.count": 1,
    "value": [
        {
            "@odata.type": "#PowerService.Policy",
            "Id": 30,
            "Type": 1,
            "PolicyName": "Group Policy",
            "PolicyDescription": "Updated policy on group",
            "PolicyState": true
        }
    ]
}
```

Table 48. Attributes

Attribute name	Description
Id	ID of the policy.
Type	Type of the policy. **NOTE:** If you have not provided the Type attribute value, by default the value is considered as one.
PolicyName	Name associated with the policy.
PolicyDescription	Description associated with the policy.
PolicyState	Indicates if the policy is enabled or disabled.

Table 49. Policy type

Enumeration Value	Policy Type
1	Static
2	Temperature-triggered

/api/PowerService/Actions/PowerService.CreatePolicy

This method creates a policy.

Supported versions of Power Manager:

- 2.0
- 1.2
- 1.1
- 1.0

POST method for creating Policy

This method creates a power policy.

Description	Creates a power policy
Privilege	DEVICE_CONFIGURATION
HTTP response codes	200

Example

```
Input to create static power policy:
{
    "Name": "Chassis Policy",
    "Description": "Power policy created on chassis",
    "Type" : 1,
    "Enabled": true,
    "Schedule": {
        "StartTime": "6:0",
        "EndTime": "15:30",
        "StartDate": "2019-06-25",
        "EndDate": "2019-07-30",
        "DaysOfTheWeek": ["MON", "TUE", "WED", "FRI", "SUN"]
    },
    "Targets": [
        {
            "Id": 10070,
            "PowerCapValue": 16661
        }
    ]
}

Input to create Temperature Triggered policy:
       {
        "Name": "Temp 02",
        "Description": "Description 01",
        "Enabled": true,
        "Type": 2,
        "Targets": [
            {
                "Id": 10189,
                "TemperatureThreshold": 21
            }
        ]
    }

Output:
{
    "Id": 32
}
```

Table 50. Attributes

Attribute name	Description
Name	Name associated with the policy.
Description	Description associated with the policy
Enabled	Indicates if the policy is enabled or disabled.
Schedule	Indicates when the policy is enabled.
StartTime	Time at which the policy interval starts.
EndTime	Time at which the policy interval ends.
StartDate	Date from which the policy is active.
EndDate	Date until which the policy is active.
DaysOfTheWeek	Days of week in which the policy is active.
Targets	Target device or group of the policy.
Id	ID of the target device or group
PowerCapValue	Power cap value for the device or group.

Table 51. Output Attributes

Attribute name	Description
Id	Policy Id

NOTE: To create a policy for a group, first provide the group information such as group ID and group power cap value. Then, provide the respective device IDs and device power cap values.

NOTE: If you are creating a policy on a group, the Targets field must be an array containing the group Id and group power cap as first value, followed by device id and power cap value of each member device.

NOTE: StartTime and EndTime must be left empty for policy which is active entire day.

NOTE: StartDate and EndDate must be left empty for policy which is active forever.

NOTE: DaysOfTheWeek must be set as empty array for policy which is active on every day.

/api/PowerService/Actions/PowerService. EditPolicy

This URI edits a policy.

Supported versions of Power Manager:

- 2.0
- 1.2
- 1.1
- 1.0

POST method for Editing Policy

This method edits a power policy.

Description	Edits a power policy
Privilege	DEVICE_CONFIGURATION
HTTP response codes	200

Example

```
Input to edit static power policy:
{
    "Id": 30,
    "Name": "Group Policy",
    "Description": "Updated policy on group",
    "Enabled": true,
    "Schedule": {
        "StartTime": "",
        "EndTime": "",
        "StartDate": "2019-06-1",
        "EndDate": "2019-07-28",
        "DaysOfTheWeek": ["MON", "TUE", "WED", "FRI", "THU", "SAT", 
"SUN"]
    },
    "Targets": [
        {
            "Id": 10132,
            "PowerCapValue": 16663
        },
        {
            "Id": 10070,
            "PowerCapValue": 16663
        }
    ]
}

Input to edit temperature-triggered policy:
    {
        "Id": 3,
        "Name": "Edited Temp Policy",
        "Description": "Edited Description",
        "Enabled": false,
        "Targets": [
        {
            "Id": 10189,
            "TemperatureThreshold": 30
        }
        ]
    }


Output:
{
    "Id": 30
}
```

Table 52. Attributes

Attribute name	Description
Id	ID of the power policy
Name	Name associated with the policy.
Description	Description associated with the policy
Enabled	Indicates if the policy is enabled or disabled.
Schedule	Indicates when the policy is enabled
StartTime	Time at which the policy interval starts.
EndTime	Time at which the policy interval ends.
StartDate	Date from which the policy is active.
EndDate	Date until which the policy is active.
DaysOfTheWeek	Days of week in which the policy is active.
Targets	Target device or group of the policy.
Id	ID of the target device or group
PowerCapValue	Power cap value for the device or group.

Table 53. Output Attributes

Attribute name	Description
Id	ID of the power policy

NOTE: All targets and corresponding power cap must be passed in the payload.

NOTE: If you are editing power policy on a group, the Targets field must be an array containing the group Id and group power cap as first value, followed by device id and power cap value of each member device.

NOTE: StartTime and EndTime must be left empty for policy which is active entire day.

NOTE: DaysOfTheWeek must be set as empty array for policy which is active on every day.

/api/PowerService/Actions/PowerService.EnablePolicies

This method enables a policy.

Supported versions of Power Manager:

- 2.0
- 1.2
- 1.1
- 1.0

POST method for Enabling Policy

This method enables power policies..

Description	Enables power policies.
Privilege	DEVICE_CONFIGURATION
HTTP response codes	200

Example

```
Input:
{
    "Ids": [30, 32]
}

Output:
{
    "Ids": [
        30,
        32
    ]
}
```

Table 54. Attributes

Attribute name	Description
Ids	List of IDs of power policies

NOTE: To enable all power policies, provide value for Ids as **-1**.

/api/PowerService/Actions/PowerService.DisablePolicies

This method disables a policy.

Supported versions of Power Manager:

- 2.0
- 1.2
- 1.1
- 1.0

POST method for Disabling Policy

This method disables power policies.

Description	Disables power policies.
Privilege	DEVICE_CONFIGURATION
HTTP response codes	200

Example

```
Input:
{
    "Ids": [30, 32]
}

Output:
{
    "Ids": [
        30,
        32
    ]
}
```

Table 55. Attributes

Attribute name	Description
Ids	List of IDs of power policies

NOTE: To disable all power policies, provide value for Ids as **-1**.

/api/PowerService/Actions/PowerService.DeletePolicies

This method deletes a policy.

Supported versions of Power Manager:

- 2.0
- 1.2
- 1.1
- 1.0

POST method for Deleting Policy

This method deletes power policies.

Description Deletes power policies.

Privilege DEVICE_CONFIGURATION

HTTP response codes 200

Example

```
Input:
{
    "Ids": [30,32]
}

Output:
{
    "Ids": [
        30,
        32
    ]
}
```

Table 56. Attributes

Attribute name	Description
Ids	List of IDs of power policies

ⓘ **NOTE:** To delete all power policies, provide value for Ids as **-1**.

8

Emergency Power Reduction

Topics:

/api/PowerService/MonitoredDevices(<DeviceId>)/EPRStatus

This URI represents the EPR status of the Power Manager monitored devices

Supported versions of Power Manager:

- 2.0
- 1.2
- 1.1
- 1.0

GET method for EPR Status of specified Devices

This method returns the EPR status of the devices that are monitored by the Power Manager.

Description	Returns the EPR status.
Privilege	VIEW
HTTP response codes	200

Example

```
Input: None

Output:
{
    "@odata.context": "/api/$metadata#PowerService.EPRStatus",
    "@odata.type": "#PowerService.EPRStatus",
    "@odata.id": "/api/PowerService/MonitoredDevices(10069)/EPRStatus",
    "Enabled": true,
    "Type": 1
    "EPRSource": 1
}
```

Table 57. Attributes

Attribute name	Description
Enabled	Status of the EPR.
Type	Type of the EPR.
EPRSource	Source of the EPR.

Table 58. EPR Type Enumeration

Enumeration Value	Description
1	Power Throttle Down (0 power cap value).
2	Power Off.

Table 59. EPR Source Enumeration

Enumeration Value	Description
1	Manual
2	Temperature-triggered

/api/PowerService/MonitoredGroups(<GroupId>)/EPRStatus

This URI represents the EPR status of the Power Manager monitored groups

Supported versions of Power Manager:

- 2.0
- 1.2
- 1.1
- 1.0

GET method for EPR Status of specified Groups

This method returns the EPR status of the groups that are monitored by the Power Manager.

Description Returns the EPR status.

Privilege VIEW

HTTP response codes 200

Example

```
Input: None

Output:
{
    "@odata.context": "/api/$metadata#PowerService.EPRStatus",
    "@odata.type": "#PowerService.EPRStatus",
    "@odata.id": "/api/PowerService/MonitoredGroups(10691)/EPRStatus",
    "Enabled": true,
    "Type": 1
    "EPRSource": 1
}
```

Table 60. Attributes

Attribute name	Description
Enabled	Status of the EPR
Type	Type of the EPR.
EPRSource	Source of the EPR.

Table 61. EPR Type Enumeration

Enumeration Value	Description
1	Power Throttle Down (0 power cap value).
2	Power Off.

Table 62. EPR Source Enumeration

Enumeration Value	Description
1	Manual
2	Temperature-triggered

/api/PowerService/EPR

This URI represents the list of Emergency Power Reduction applied on the targets.

Supported versions of Power Manager:

- 2.0
- 1.2
- 1.1
- 1.0

GET method for retrieving the targets where EPR is applied

This method returns the collection of EPR applied on the targets monitored by the Power Manager

Description Returns the collection of EPR.

Privilege VIEW

HTTP response codes 200

Example

```
Input: None

Output:
{
    "@odata.context": "/api/
$metadata#Collection(PowerService.EprPolicy)",
    "@odata.count": 1,
    "value": [
        {
            "@odata.type": "#PowerService.EprPolicy",
            "@odata.id": "/api/PowerService/EPR(6)",
            "PolicyId": 6,
            "Type": 1,
            "Name": "EPR policy for 10069 at 6/12/19 5:39 PM",
            "Description": "EPR policy for target 10069 applied at
6/12/19 5:39 PM.",
            "Enabled": true,
            "AssignedTo": "4xBzewP",
            "CreatedTime": "2019-06-12 17:39:11.661279",
            "IsEprPowerDown": false,
            "IsAssociatedToGroup": false,
            "ExecutionState": 3
        }
    ]
}
```

Table 63. Attributes

Attribute	Description
PolicyId	Id of the EPR created.
Name	Name of the EPR.
Description	Description of the EPR.
Enabled	Status of the EPR.
AssignedTo	Target on which EPR is applied.
CreatedTime	Time at which EPR is created.
IsEprPowerDown	True, if EPR type is Power Down.
IsAssociatedToGroup	True, if EPR is associated to a group.
ExecutionState	State of the EPR execution.

Table 64. EPR Execution State Enumeration

Enumeration Value	Description
1	NOSTATE (EPR execution state is not yet determined).
2	EXECUTING (EPR execution is in progress or pending).
3	SUCCESS (EPR execution is successful).
5	FAILED (EPR execution failed).

Table 65. Filters

Field	Operator
Type	eq
Name	contains
Description	contains
Enabled	eq
AssignedTo	contains

List of all the sort options that this method supports:

- PolicyId
- Type
- Name
- Description
- Enabled
- AssignedTo
- CreatedTime
- IsEprPowerDown
- IsAssosciatedToGroup

Table 66. EprPowerDown

Value	Description
False	Throttle

Table 66. EprPowerDown (continued)

Value	Description
True	Shutdown

/api/PowerService/Actions/PowerService.EnableEPR

This URI enables you to enable EPR on the target.

Supported versions of Power Manager:

- 2.0
- 1.2
- 1.1
- 1.0

POST method to Enable EPR

This method enables EPR on the specified target.

Description	Enables EPR on the specified target
Privilege	DEVICE_CONFIGURATION
HTTP response codes	200
Example	

```
Input:
{
  "Target": 10149,
  "Action": "POWER_THROTTLE"
}

Output:
{
  "Id": 7
}
```

Table 67. Attributes

Attribute Name	Description
Target	Device Id or Group Id on which EPR is to be applied.
Action	EPR action (POWER_THROTTLE or POWER_OFF)
Id	Id of the created EPR.

NOTE: EPR can be applied to single device or single group at a time.

/api/PowerService/Actions/PowerService.DisableEPR

This URI enables you to disable the EPR from the targets.

Supported versions of Power Manager:

- 2.0
- 1.2
- 1.1

- 1.0

POST method to Disable EPR

This method disables EPR on the specified target.

Description Disbales EPR on the specified target

Privilege DEVICE_CONFIGURATION

HTTP response codes 200

Example

```
Input:
{
  "Ids": [6,7]
}

Output:
{
  "Ids": [6,7]

}
```

Table 68. Attributes

Attribute Name	Description
Id	EPR Ids to be disabled.

ⓘ **NOTE:** To disable all the EPRs, provide value for Ids as **-1**.

9

Unmonitored devices

Topics:

api/PowerService/UserDefinedEntities/Actions/UserDefinedEntities.Create

Use this URI to add unmonitored devices that cannot be discovered in OpenManageEnterprise.

Supported versions of Power Manager:

- 2.0
- 1.2

POST method to add unmonitored device

Description Use this method to add unmonitored devices that is not supported in OpenManange Enterprise.

Privilege DEVICE_CONFIGURATION

HTTP response codes 200

Example

```
Input:
{

    "Name": "iDRAC-6SN09X2",
    "Description": "iDRAC-6SN09X2",
    "HostName": "W103L1HH92",
    "Identifier": "C3YCW22",
    "Size": 2,
    "EstimatedMaxPower": 10000,
    "Model": "Power Edge"
}

Output:
{
    "Status": "SUCCESS"
}
```

Table 69. Attributes

Attribute	Description
Name	Name of the unmonitored device.
Description	Description of the unmonitored device.
HostName	Host name of the unmonitored device.

Table 69. Attributes (continued)

Attribute	Description
Identifier	Identifier of the unmonitored device.
Size	Space occupied by the unmonitored device in a rack.
EstimatedMaxPower	Maximum power defined for the unmonitored device.
Model	Model of the unmonitored device.

api/PowerService/UserDefinedEntities/Actions/UserDefinedEntities.Update

Use this URI to edit the unmonitored device details.

Supported versions of Power Manager:

- 2.0
- 1.2

POST method to edit unmonitored device

Description Use this method to edit unmonitored device details.

Privilege DEVICE_CONFIGURATION

HTTP response codes 200

Example

```
Input:
{
    "Id":1,
    "Name": "Device-A",
    "Description": "WCK100098",
    "HostName": "WCK100098",
    "Identifier": "C3YCW22",
    "Size": 2,
    "EstimatedMaxPower": 10000,
    "Model": "Power Edge"
}

Output:
{
    "Status": "SUCCESS"
}
```

Table 70. Attributes

Attribute	Description
Id	Id of the unmonitored device.
Name	Name of the unmonitored device.
Description	Description of the unmonitored device.
HostName	Host name of the unmonitored device.
Identifier	Identifier of the unmonitored device.
Size	Space occupied by the unmonitored device in a rack.
EstimatedMaxPower	Maximum power defined for the unmonitored device.

Table 70. Attributes (continued)

Attribute	Description
Model	Model of the unmonitored device.

api/PowerService/UserDefinedEntities/Actions/UserDefinedEntities.Delete

Use this URI to delete the unmonitored device details.

Supported versions of Power Manager:

- 2.0
- 1.2

POST method to delete unmonitored device

Description Use this method to delete unmonitored devices that is not supported in OpenManange Enterprise.

Privilege DEVICE_CONFIGURATION

HTTP response codes 200

Example

```
Input:
{
"Ids": [6,7]
}

Output:
{
    "Status": "SUCCESS"
}
```

Table 71. Attributes

Attribute	Description
Ids	Unmonitored device id to be deleted.

NOTE: To delete all the devices, provide the value for Ids as -1.

api/PowerService/UserDefinedEntities/Devices

Use this URI to get list of all unmonitored devices.

Supported versions of Power Manager:

- 2.0
- 1.2

GET method to list unmonitored devices

Description Use this method to view all the unmonitored devices.

Privilege VIEW

HTTP response codes 200

Example

```
Input: None

Output:
{
    "@odata.count": 1,
    "value": [
        {
            "Id":1,
            "Name": "Device A",
            "Description": "Device A",
            "HostName": "W109GHK890",
            "Identifier": "CXDFGHJ",
            "Size": 2,
            "EstimatedMaxPower": 10000,
            "Model": "Power Edge",
            "Location": "DC/Room/Aisle/Rack1",
            "RackSlot": 10,

        }
    ]
}
```

Table 72. Filters

Field	Operator
Id	eq
Name	contains
Description	contains
Hostname	contains
Identifier	contains
Size	eq
EstimatedMaxPower	eq
Model	contains

List of all the sort options that this method supports:

- Id
- Name
- Description
- HostName
- Identifier
- Size
- EstimatedMaxPower
- Model
- RackSlot
- Location

api/PowerService/UserDefinedEntities/Devices(<DeviceId>)

Use this URI to get individual user defined entity details.

Supported versions of Power Manager:

- 2.0
- 1.2

GET method to view individual user defined device details

Description Use this method to view all the unmonitored devices.

Privilege VIEW

HTTP response codes 200

Example

```
Input: None

Output:
{

           "Id":10009,
            "Name": "Device A",
            "Description": "Device A",
            "HostName": "W109GHK890",
            "Identifier": "CXDFGHJ",
            "Size": 2,
            "EstimatedMaxPower": 10000,
            "Model": "Power Edge",
            "Location": "DC/Room/Aisle/Rack1",
            "RackSlot": 10,


}
```

10

Physical Groups

Topics:

api/GroupService/Actions/GroupService.CreateGroup

Use this URI to create physical groups.

Supported versions of Power Manager:

- 2.0
- 1.2

POST method to create physical group

Description	Creates a physical group.
Privilege	GROUP_MANAGEMENT
HTTP response codes	200

Example

```
Input:
{
  "GroupModel":
  {
  "Name": "Rack 001",
  "Description": "Rack 001",
  "MembershipTypeId": 12,
  "ParentId": 1029,
  "GroupTags":[
        {
            "Name":"RACK"
        }
    ],
    "GroupAttributeTypeValues": [
        {
            "AttributeTypeName": "POWER_CAPACITY",
            "IdDataType": 2,
            "Value": "1000"
        },
        {
            "AttributeTypeName": "SPACE_CAPACITY",
            "IdDataType": 2,
            "Value": "42"
        }
    ]
  }
}

Output:
{
    10113
}
```

Table 73. Attributes

Attribute	Description
Name	Name of the physical group.
Description	Description of the physical group.
MembershipTypeId	Define the group type as static. To create a physical group, pass this value as 12.
ParentId	Id of the parent group. To create a group immediately under physical hierarchy, provide the group id as `1029`. To create a group under an existing physical group, pass the respective group id.
AttributeTypeName	Type of capacity.
IdDataType	Represents attribute data type.
Value	Value of the attribute.
GroupTags	Group type under the parent group.
GroupAttributeTypeValues	Details of the group you are creating.

List of all the supported group tags options that this method supports:

- DATA_CENTER
- ROOM
- AISLE
- RACK

Table 74. Supported attributes

Enumeration Value	Description
POWER_CAPACITY	This attribute is mandatory only for the group type rack and optional for other types.
SPACE_CAPACITY	This field is only applicable to group type RACK.

api/GroupService/Actions/GroupService.UpdateGroup

Use this URI to update physical groups.

Supported versions of Power Manager:

- 2.0
- 1.2

POST method to update physical group

Description	Updates a physical group.
Privilege	GROUP_MANAGEMENT
HTTP response codes	200

Example

```
Input:
{
"GroupModel": {
    "Name": "Rack 001",
    "Description": "Rack 001",
    "MembershipTypeId": 12,
    "ParentId": 1029,
    "Id": 10429,
    "GroupAttributeTypeValues": [
        {
            "AttributeTypeName": "POWER_CAPACITY",
            "IdDataType": 2,
            "Value": "500"
        },
        {
            "AttributeTypeName": "SPACE_CAPACITY",
            "IdDataType": 2,
            "Value": "42"
        }
    ]
  }
}

Output:
{
    10429
}
```

api/GroupService/Actions/GroupService.DeleteGroup

Use this URI to delete physical groups.

Supported versions of Power Manager:

- 2.0
- 1.2

POST method to delete physical group

Description	Deletes a physical group.

Privilege	GROUP_MANAGEMENT
HTTP response codes	204

Example

```
Input:
{
    "GroupIds": [10166]
}

Output: None
```

/api/PowerService/PhysicalGroups/GroupDetails

Use this URI to get a list of all the physical groups.

Supported versions of Power Manager:

- 2.0
- 1.2

GET method to list all physical groups

Description	Displays a list of all the physical groups created.
Privilege	VIEW
HTTP response codes	200

Example

```
Input: None

Output:
{
    "@odata.context": "/api/
$metadata#Collection(PhysicalGroup.GroupDetails)",
    "@odata.count": 1,
    "value":[
            {
            "@odata.type": "#PhysicalGroup.GroupDetails",
            "@odata.id": "/api/PowerService/PhysicalGroups/
GroupDetails(11182)",
            "Id": 11182,
            "ParentId": 1029,
            "Name": "DC",
            "Description": "DC",
            "AddedOn": "2020-06-15 12:03:59.579174",
            "DevicesInWorkingSet": 4,
            "Tag": "DATA_CENTER",
            "Location": null,
            "SpaceCapacity": 261,
            "PowerCapacity": 10000,
            "CalculatedPowerCapacity": 14100,
            "SparePowerCapacity": 9273,
            "SpareSpaceCapacity": 229,
            "TemperatureTriggeredPolicyExist": false,
            "AllLeafgroups@odata.navigationLink": "/api/PowerService/
PhysicalGroups/GroupDetails(10356)/AllLeafgroups",
            "Devices@odata.navigationLink": "/api/PowerService/
PhysicalGroups/GroupDetails(10356)/Devices",
            "AssociatedDevices@odata.navigationLink": "/api/PowerService/
PhysicalGroups/GroupDetails(10356)/AssociatedDevices"
            }
               ]
}
```

Table 75. Attributes

Attribute name	Description
id	Group Id
parentId	Group Parent ID
Name	Group name
description	Group description
addedOn	Date when the group is added.
devicesInWorkingSet	Number of devices available in monitored list
type	Type of physical group
tag	Tag of physical group
location	Physical group location
spaceCapacity	Total space capacity
occupiedSpace	Total occupied space
powerCapacity	Total power capacity of group
calculatedPowerCapacity	Sum of all the child group power consumption
sparePowerCapacity	Unused power capacity that is also known as the power headroom.
spareSpaceCapacity	Unused space capacity that is also known as space headroom.
temperatureTriggeredPolicyExist	Temperature policy is created or not.

Table 76. Filters

Field	Operator
Id	eq
ParentId	eq
Name	contains
Description	contains
Tag	contains
SpaceCapacity	eq
PowerCapacity	eq
CalculatedPowerCapacity	eq
SparePowerCapacity	eq
SpareSpaceCapacity	eq

List of all the sort options that this method supports:

- Id
- ParentId
- Tag
- Location
- Name
- DevicesInWorkingSet
- Description

- SpaceCapacity
- PowerCapacity
- SparePowerCapacity

  NOTE: To view underutilized racks for power, use the **SparePowerCapacity** sort option.
- SpareSpaceCapacity

  NOTE: To view underutilized racks for space, use the **SpareSpaceCapacity** sort option.
- CalculatedPowerCapacity
- AddedOn

/api/PowerService/PhysicalGroups/GroupDetails(<GroupID>)

Use this URI to get individual physical group details.

Supported versions of Power Manager:

- 2.0
- 1.2

GET method to get individual physical group details

Description	View individual physical group details.
Privilege	VIEW
HTTP response codes	200

Example

```
Input: None

Output:
{
    "@odata.context": "/api/$metadata#PhysicalGroup.GroupDetails",
    "@odata.type": "#PhysicalGroup.GroupDetails",
    "@odata.id": "/api/PowerService/PhysicalGroups/GroupDetails(11182)",
    "Id": 11182,
    "ParentId": 1029,
    "Name": "DC",
    "Description": "DC",
    "AddedOn": "2020-06-15 12:03:59.579174",
    "DevicesInWorkingSet": 4,
    "Tag": "DATA_CENTER",
    "Location": null,
    "SpaceCapacity": 261,
    "PowerCapacity": 10000,
    "CalculatedPowerCapacity": 14100,
    "SparePowerCapacity": 9269,
    "SpareSpaceCapacity": 229,
    "TemperatureTriggeredPolicyExist": false,
    "AllLeafgroups@odata.navigationLink": "/api/PowerService/
PhysicalGroups/GroupDetails(11182)/AllLeafgroups",
    "Devices@odata.navigationLink": "/api/PowerService/PhysicalGroups/
GroupDetails(11182)/Devices"
}
```

Table 77. Attributes

Attribute name	Description
Id	Group Id

Table 77. Attributes (continued)

Attribute name	Description
parentId	Group Parent ID
Name	Group name
Description	Group description
addedOn	Date when the group is added.
devicesInWorkingSet	Number of devices available in monitored list
Type	Type of physical group
Tag	Tag of physical group
Location	Physical group location
spaceCapacity	Total space capacity
occupiedSpace	Total occupied space
powerCapacity	Total power capacity of group
calculatedPowerCapacity	Sum of all the child group power consumption
sparePowerCapacity	Unused power capacity
spareSpaceCapacity	Unused space capacity
temperatureTriggeredPolicyExist	Temperature policy is created or not.

api/PowerService/PhysicalGroups/GroupDetails<GroupID>/AllLeafgroups

Use this URI to view all the groups under this specific group.

Supported versions of Power Manager:

- 2.0
- 1.2

GET method to list all the groups recursively for the specific group

Description Displays a list of all the physical groups under the specified parent group.

Privilege VIEW

HTTP response codes 200

Example

```
Input: None

Output:
{
    "@odata.context": "/api/
$metadata#Collection(PhysicalGroup.GroupDetails)",
    "@odata.count": 1,
    "value": [
        {
            "@odata.type": "#PhysicalGroup.GroupDetails",
            "@odata.id": "/api/PowerService/PhysicalGroups/
GroupDetails(11183)",
            "Id": 11183,
            "ParentId": 11183,
            "Name": "Rack1",
            "Description": "Rack1",
            "AddedOn": "2020-06-15 12:04:17.941671",
            "DevicesInWorkingSet": 3,
            "Tag": "RACK",
            "Location": "DC",
            "SpaceCapacity": 21,
            "PowerCapacity": 1000,
            "CalculatedPowerCapacity": 1000,
            "SparePowerCapacity": 448,
            "SpareSpaceCapacity": 5,
            "AllLeafgroups@odata.navigationLink": "/api/PowerService/
PhysicalGroups/GroupDetails(11183)/AllLeafgroups",
            "Devices@odata.navigationLink": "/api/PowerService/
PhysicalGroups/GroupDetails(11183)/Devices"
        }
    ]
}
```

Table 78. Filters

Field	Operator
Id	eq
Name	contains
Description	contains
Tag	eq
SpaceCapacity	eq
PowerCapacity	eq
CalculatedPowerCapacity	eq
SparePowerCapacity	eq
SpareSpaceCapacity	eq

List of all the sort options that this method supports:

- Id
- ParentId
- Location
- Name
- DevicesInWorkingSet
- Description
- SpaceCapacity
- PowerCapacity
- SparePowerCapacity
- SpareSpaceCapacity
- CalculatedPowerCapacity
- AddedOn

api/PowerService/Actions/PowerService.UpdateDeviceDetails

Use this URI to add device details information like device size and estimated max power.

Supported versions of Power Manager:

- 2.0

- 1.2

POST method to update device details

Description	Add device details information like device size and estimated max power.
Privilege	DEVICE_CONFIGURATION
HTTP response codes	200

Example

```
Input:
{
        "Devices": [
            {
                "Id": 10069,
                "Size": 2,
                "EstimatedMaxPower": 1000
            },
            {
                "Id": 10068,
                "Size": 2,
                "EstimatedMaxPower": 10000
            }
        ]
    }

Output:
{
    "Status": "SUCCESS"
}
```

/api/PowerService/DeviceDetails

Use this URI to view the device information such as device size and estimated maximum power.

Supported versions of Power Manager:

- 2.0
- 1.2

POST method to fetch device details

Description	Get device details information like device size and estimated max power.
Privilege	VIEW
HTTP response codes	200

Example

```
Input:
{
    "DeviceId" :10075
}


Output:
{
   "Size": 2,
   "EstimatedMaxPower": 1000
}
```

api/PowerService/PhysicalGroups/CapableDevices

Use this URI to get all the OpenManageEnterprise devices such as servers (except blade servers), chassis, storage(only rack placeable) and network switches (only rack placeable), which can be added into a rack.

Supported versions of Power Manager:

- 2.0
- 1.2

GET method to list all rack placeable devices

Description This method returns rack capable devices.

Privilege VIEW

HTTP response codes 200

Example

```
Input: None

Output:
{
    "@odata.context": "/api/
$metadata#Collection(PhysicalGroup.CapableDevices)",
    "@odata.count": 1,
    "value": [
        {
            "@odata.type": "#PhysicalGroup.CapableDevices",
            "@odata.id": "/api/PowerService/PhysicalGroups/
CapableDevices(10204)",
            "Id": 10204,
            "Name": "100.96.25.137",
            "ServiceTag": "HWM3VG2",
            "Type": 1000,
            "Model": "PowerEdge T630",
            "Size": 5,
            "PowerState": 17,
            "ManagedState": 3000,
            "ConnectionState": true,
            "HealthStatus": 4000,
            "EstimatedMaxPower": 0
            "PduOutletCount": 12
        }

    ]
}
```

Table 79. Attributes

Attribute name	Description
id	Device id
Name	Device name
ServiceTag	Service tag of device
Type	Device type
Size	Space occupied by the device in a rack

Table 79. Attributes (continued)

Attribute name	Description
Model	Device model
PowerState	Power State of the device
ManagedState	Managed State of the device
ConnectionState	Connection State of the device
HealthStatus	Health Status of the device
EstimatedMaxPower	Maximum power defined for the device
PduOutletCount	Power State outlet count of the device.

Table 80. Filters

Field	Operator
Id	eq
Name	contains
ServiceTag	contains
Type	eq
Model	contains
PowerState	eq
ManagedState	eq
ConnectionState	eq
HealthStatus	eq
Size	eq
EstimatedMaxPower	eq

List of all the sort options that this method supports:

- Id
- Name
- ServiceTag
- Type
- Model
- PowerState
- ManagedState
- ConnectionState
- HealthStatus
- Size
- EstimatedMaxPower
- PduOutletCount

api/PowerService/PhysicalGroups/CapableDevices(<DeviceId>)

Use this URI to get the selected device details.

Supported versions of Power Manager:

- 2.0
- 1.2

GET method for selected device details

Description This method returns the selected rack capable devices.

Privilege VIEW

HTTP response codes 200

Example

```
Input: None

Output:
{
    "@odata.context": "/api/$metadata#PhysicalGroup.CapableDevices",
    "@odata.type": "#PhysicalGroup.CapableDevices",
    "@odata.id": "/api/PowerService/PhysicalGroups/
CapableDevices(10218)",
    "Id": 10218,
    "Name": "100.96.24.159",
    "ServiceTag": "1W43F02",
    "Type": 1000,
    "Model": "PowerEdge R720",
    "Size": 2,
    "PowerState": 17,
    "ManagedState": 4000,
    "ConnectionState": true,
    "HealthStatus": 4000,
    "EstimatedMaxPower": 0
    "PduOutletCount": 12
}
```

Table 81. Attribute list

Attribute name	Description
id	Device id
Name	Device name
ServiceTag	Service tag of device
Type	Device type
Size	Space occupied by the device in a rack
Model	Device model
PowerState	Power State of the device
ManagedState	Managed State of the device
ConnectionState	Connection State of the device
HealthStatus	Health Status of the device
EstimatedMaxPower	Maximum power defined for the device
PduOutletCount	Power State out let count of the device.

api/PowerService/PhysicalGroups/Actions/PhysicalGroup.AddMemberDevices

Use this URI to add device as a member to a physical group at the rack level.

Supported versions of Power Manager:

- 2.0
- 1.2

POST method add device to physical groups

Description This method adds a device as a member to a physical group at the rack level.

Privilege GROUP_MANAGEMENT

HTTP response codes 200

Example

```
Input:
{
    "GroupId": 10001,
    "MemberDevices" : [
                {
                    "Id": 10001,
                    "SlotNumber": 1

                },
                {
                    "Id": 10002,
                    "SlotNumber": 2
                }
        ]
}

Output:
{
    "Status": "SUCCESS"
}
```

NOTE: To add the device automatically in an empty slot, provide the `SlotNumber` value as `zero`.

NOTE: Ensure that Emergency Power Reduction (EPR) is not enabled on the group while adding a device to the group.

NOTE: If you are adding a Power Distribution Unit (PDU) device, then the `SlotNumber` value is not required.

api/PowerService/PhysicalGroups/Actions/PhysicalGroup.RemoveMemberDevices

Use this URI to remove device from a physical group at the rack level.

Supported versions of Power Manager:

- 2.0
- 1.2

POST method remove device from physical groups

Description This method removes a device as a member from the physical group at the rack level.

Privilege GROUP_MANAGEMENT

HTTP response codes 200

Example

```
Input:
{
    "GroupId": 10001,
    "MemberDeviceIds": [10001,10002]
}

Output:
{
    "Status": "SUCCESS"
}
```

api/PowerService/PhysicalGroups/GroupDetails(<GroupId>)/Devices

Use this URI to get all the devices such as supported and unsupported or either of them under the physical group in a rack.

Supported versions of Power Manager:

- 2.0
- 1.2

GET method to list all the devices from the physical group

Description

This method gets all the devices such as supported and unsupported or either of them under the physical group in a rack.

Privilege

VIEW

HTTP response codes

200

Example

```
Input:
{
    "@odata.context": "/api/$metadata#Collection(PhysicalGroup.Device)",
    "@odata.count": 4,
    "value": [
        {
            "@odata.type": "#PhysicalGroup.Device",
            "Id": 10132,
            "Name": "100.96.24.38",
            "ServiceTag": "68S4WV2",
            "HostName": null,
            "Type": 1000,
            "PowerState": 17,
            "ManagedState": 3000,
            "ConnectionState": true,
            "HealthStatus": 4000,
            "Model": "PowerEdge R440",
            "DeviceBladeInfo": [],
            "IsUserDefined": false,
            "Size": 1,
            "SlotNumber": 21,
            "IsMonitored": false,
            "EstimatedMaxPower": null
            "AveragePower": 86
        }
    ]
}

Output: None
```

Table 82. Attributes

Attribute name	Description
Id	Device id
Name	Device name
ServiceTag	Service tag of device
HostName	Host name of the device
Type	Device type
PowerState	Power State of the device
ManagedState	Managed State of the device
ConnectionState	Connection State of the device
HealthStatus	Health Status of the device
Model	Device Model
DeviceBladeInfo	List all devices under chassis
IsUserDefined	Value is true for unmonitored device. Value is false for monitored device.
Size	Space occupied by the device in a rack
SlotNumber	Slot number of device on rack
IsMonitored	If the device is monitored by Power Manager
Location	Device location
EstimatedMaxPower	Maximum power defined for the device
AveragePower	Recent average power consumed by the device

Table 83. Filters

Field	Operator
Id	eq
Name	contains
ServiceTag	contains
Type	eq
Model	contains
PowerState	eq
ManagedState	eq
ConnectionState	eq
HealthStatus	eq
Size	eq
SlotNumber	eq
EstimatedMaxPower	eq

List of all the sort options that this method supports:

- Id
- Name
- ServiceTag
- Type

- Size
- Model
- PowerState
- ManagedState
- ConnectionState
- HealthStatus
- SlotNumber
- EstimatedMaxPower

api/PowerService/PhysicalGroups/Actions/PhysicalGroup.UpdateSlots

Use this URI to update physical group member rack slots.

Supported versions of Power Manager:

- 2.0
- 1.2

POST method to update device slots

Description	Use this method to update physical group member rack slots.
Privilege	GROUP_MANAGEMENT
HTTP response codes	200

Example

```
Input:
{
        "GroupId": 10001,
        "MemberDevices" : [

                {
                    "Id": 10001,
                    "SlotNumber": 1
                },
                {
                    "Id": 10002,
                    "SlotNumber": 2
                }

        ]
}


Output:
{
    "Status": "SUCCESS"
}
```

api/PowerService/PhysicalGroups/Actions/PhysicalGroup.MoveDevice

Use this URI to move device from a physical group.

Supported versions of Power Manager:

- 2.0
- 1.2

POST method to move device from one rack to another

Description Use this method to move the device from physical group.

Privilege GROUP_MANAGEMENT

HTTP response codes 200

Example

```
Input:
{
    "EntityId": 10000,
    "FromGroup" : 10001,
    "ToGroup": 10002
}

Output:
{
    "Status": "SUCCESS"
}
```

ⓘ **NOTE:** Ensure that Emergency Power Reduction (EPR) is not enabled on the target group while moving a device to this group.

11

Power Distribution Units

Topics:

/api/PowerService/PhysicalGroups/GroupDetails(<Group ID>)/AssociatedDevices

URI represents all the Power Distribution Unit (PDU) devices that are associated to a specific physical group.

Supported versions of Power Manager:

- 2.0

GET method to represent all the associated PDU devices of a specific physical group

This method represents all the PDU devices from physical group.

Description Returns all the PDU devices that are associated to specific group.

Privilege VIEW

HTTP response codes 200

Example

```
Input: None
Output:
{
    "@odata.context": "/api/
$metadata#Collection(PhysicalGroup.AssociatedDevice)",
    "@odata.count": 1,
    "value": [
        {
            "@odata.type": "#PhysicalGroup.AssociatedDevice",
            "Id": 10213,
            "Name": "100.96.23.19",
            "Identifier": "ZY20082297",
            "SerialNumber": "ZY20082297",
            "Type": 100,
            "PowerState": 17,
            "Model": "MUU3EGM0-12P027-2C20A9A00-S",
            "Manufacturer": "Vertiv",
            "FirmwareVersion": "5.6.1",
            "HardwareRevision": "gmmb",
            "PowerRating": 0,
            "OutletCount": 12,
            "MeteredOutletCount": 12
        }
    ]
}
```

Table 84. Attributes

Attribute	Description
Id	Device id
Name	Device name
Identifier	Device identifier
SerialNumber	Serial number of device
Type	Device type
PowerState	Power State of the device
Model	Model number of device
Manufacturer	Manufacturer name of device
FirmwareVersion	Firmware version of the device
HardwareRevision	Hardware revision of device
PowerRating	Power rating of device
OutletCount	Outlet count of device
MeteredOutletCount	Metered outlet count

Table 85. Filters

Field	Operator
Id	eq
Name	contains
Identifier	contains
Model	contains
ServiceTag	contains
Manufacturer	contains
FirmwareVersion	contains
HardwareRevision	contains
PowerState	eq
PowerRating	eq
OutletCount	eq

List of all the sort options that this method supports:

- Id
- Name
- Identifier
- Model
- PowerState
- OutletCount
- PowerRating
- Manufacturer
- FirwareVersion
- HardwareRevision

/api/PowerService/PowerDevice/Device(arg1)/Details

URI represents power device with id.

Supported versions of Power Manager:

- 2.0

Get method to retrieve power device details by id

This method represents power device details.

Description	This method return power device details.
Privilege	VIEW
HTTP response codes	200

Example

```
Input: None
Output:
{
    "@odata.context": "/api/
$metadata#PowerDevice.DeviceInventoryDetails",
    "@odata.type": "#PowerDevice.DeviceInventoryDetails",
    "@odata.id": "/api/PowerService/PowerDevice/Device(10126)/Details",
    "Id": 10126,
    "Manufacturer": "Vertiv",
    "FirmwareVersion": "5.6.1",
    "HardwareRevision": "gmmb",
    "SerialNumber": "ZY20082297",
    "PowerRating": 0,
    "OutletCount": 12,
    "MeteredOutletCount": 12,
    "Outlets": [
        {
            "Number": 1,
            "Name": "Outlet 1",
            "PowerRating": 0
        },
        {
            "Number": 10,
            "Name": "Outlet 10",
            "PowerRating": 0
        }
    ]
}
```

Table 86. Attributes

Attribute	Description
Id	Device id
Manufacturer	Device manufacturer
FirmwareVersion	Firmware version of device
HardwareRevision	Hardware version of device
SerialNumber	Serial number of device
PowerRating	Power rating of device
OutletCount	Outlet count of device
MeteredOutletCount	Metered outlet count for device

Table 86. Attributes (continued)

Attribute	Description
Outlets	Device outlets details

12

Virtual machines

Topics:

- /api/PowerService/VirtualEntities/VirtualMachines
- /api/PowerService/VirtualEntities/VirtualMachines(<VirtualMachine ID>)
- PowerService/VirtualEntities/VirtualMachines(<VirtualMachine ID>)/Details

/api/PowerService/VirtualEntities/VirtualMachines

URI to represent all the virtual machines.

Supported versions of Power Manager:

- 2.0

GET method to represent all the virtual machines

This method displays all the virtual machines that are monitored in Power Manager.

Description	Returns all the virtual machines.
Privilege	VIEW
HTTP response codes	200

Example

```
Input: None
Output:
{
    "@odata.context": "/api/
$metadata#Collection(VirtualEntities.VirtualMachine)",
    "@odata.count": 2,
    "value": [
        {
            "@odata.type": "#VirtualEntities.VirtualMachine",
            "@odata.id": "/api/PowerService/VirtualEntities/
VirtualMachines(1)",
            "Id": 1,
            "Uuid": "1798203e-dd65-4b5c-9772-07f578b91730",
            "Name": "Lubuntu 18.04 01",
            "DeviceId": 10207,
            "DeviceName": "BR1H903",
            "ServiceTag": "BR1H903",
            "Details": {
                "@odata.id": "/api/PowerService/VirtualEntities/
VirtualMachines(1)/Details"
            }
        },
        {
            "@odata.type": "#VirtualEntities.VirtualMachine",
            "@odata.id": "/api/PowerService/VirtualEntities/
VirtualMachines(7)",
            "Id": 7,
            "Uuid": "cf43af93-97ae-4f6d-9c4a-21b9d374a4ad",
            "Name": "Slitaz 01",
            "DeviceId": 10207,
            "DeviceName": "BR1H903",
            "ServiceTag": "BR1H903",
            "Details": {
                "@odata.id": "/api/PowerService/VirtualEntities/
VirtualMachines(7)/Details"
            }
    ]
}
```

Table 87. Attributes

Attribute	Description
Id	Id of the device
Name	Device name
Uuid	Unique identifier of device
DeviceId	Device id or device name of the host server.
DeviceName	Device name
ServiceTag	Service tag of device

Table 88. Filters

Field	Operator
Id	eq
Name	contains
DeviceId	eq
DeviceName	contains
ServiceTag	contains
Uuid	eq

List of all the sort options that this method supports:

- Id
- Name
- DeviceId
- DeviceName
- ServiceTag

/api/PowerService/VirtualEntities/VirtualMachines(<VirtualMachine ID>)

URI to represent virtual machines with id

Supported versions of Power Manager:

- 2.0

GET method to represent virtual machines with id

This method provides a specific virtual machine.

Description	Returns specific virtual machine
Privilege	VIEW
HTTP response codes	200

Example

```
Input: None
Output:
{
            "@odata.type": "#VirtualEntities.VirtualMachine",
            "@odata.id": "/api/PowerService/VirtualEntities/
VirtualMachines(4)",
            "Id": 4,
            "Uuid": "4d303ae2-7b2c-488b-9c09-c6d23f4e7fa3",
            "Name": "pmp1-edited",
            "DeviceId": 10207,
            "DeviceName": "BR1H903",
            "ServiceTag": "BR1H903",
            "Details": {
                "@odata.id": "/api/PowerService/VirtualEntities/
VirtualMachines(4)/Details"
            }
}
```

Table 89. Attributes

Attribute	Description
Id	Device id
Name	Device name
Uuid	Unique identifier of device
DeviceId	Device id or device name of the host server.
DeviceName	Device name
ServiceTag	Service tag of device

PowerService/VirtualEntities/VirtualMachines(<VirtualMachine ID>)/Details

URI to represent virtual machine details

Supported versions of Power Manager:

- 2.0

GET method to retrieve virtual machine details by virtual machine id

This method represents specific virtual machine details.

Description Returns specific virtual machine details.

Privilege VIEW

HTTP response codes 200

Example

```
Input: None
Output:
{
    "@odata.context": "/api/$metadata#VirtualMachines.Details",
    "@odata.type": "#VirtualMachines.Details",
    "@odata.id": "/api/PowerService/VirtualMachines/MonitoredVMs(100)/
Details",
    "Id": 100,
    "Name": "VM_36",
    "DevicePowerConsumption": 315,
    "DeviceCpuUtilization": 0,
    "VcpuCount": 1,
    "CurrentVcpuUsage": 0
}
```

Table 90. Attributes

Attribute	Description
Id	Device id
Name	Device name
DevicePowerConsumption	Device power consumption
DeviceCpuUtilization	Device CPU utilization
VcpuCount	Vcpu count
CurrentVcpuUsage	Current vcpu usage

13

Report Service

Topics:

- /api/ReportService/ReportDefs

/api/ReportService/ReportDefs

This method creates a custom report.

Supported versions of Power Manager:

- 2.0
- 1.2
- 1.1
- 1.0

Post method for creating a custom Power Manager Report

Description Creates a custom report.

Privilege REPORT_MANAGEMENT

HTTP response codes 201

Example

```
Input:
{
"Name":"Test Report", "Description":"", "ColumnNames":[
{
"Width":20,
"Sequence":0, "Name":"Device Name"
},
{
"Width":20,
"Sequence":1, "Name":"Device Service Tag"
},
{
"Width":20,
"Sequence":2, "Name":"Device Model"
},
{
"Width":20,
"Sequence":3,
"Name":"Software Component ID"
},
{
"Width":20,
"Sequence":4, "Name":"Software Description"
}
],
"FilterGroupId":64, "QueryDefRequest":{
"ContextId":3, "ResultFields":[
{
"FieldId":61
},
{
"FieldId":64
},
{
"FieldId":63
},
{
"FieldId":99
},
{
"FieldId":102
}
],
"SortFields":[],

"ReportSettings": [{
            "SettingId": 1,
            "OperatorId": 1,
            "Value": "90"
        },
        {
            "SettingId": 2,
            "OperatorId": 1,
            "Value": "2"
        }]
}
}
```

Table 91. SettingId Enumeration

Enumeration Value	Description
1	Report duration
2	Aggregation period
3	Metric type

Table 92. Report duration Enumeration

Enumeration Value	Description
1	1 Day
7	7 Days
15	15 Days
30	1 Month
90	3 Months
180	6 Months
365	1 Year

Table 93. Aggregation Period Enumeration

Enumeration Value	Description
1	1 Hour
2	1 Day

Table 94. Metric Type Enumeration

Enumeration Value	Description
3	Power
7	Temperature