# OpenManage Enterprise Power Manager

## RESTful API Guide version 1.1

June 2020
Rev. A00

DELL EMC

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

For more information about REST, see publically-available resources.

**Topics:**



# Introduction to Power Manager

Dell EMC OpenManage Enterprise Power Manager is an extension to the Dell EMC OpenManage Enterprise (OME) console and uses fine-grained instrumentation to provide increased visibility to power consumption, anomalies, and utilization. Also, Power Manager 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 or 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.

# 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
```

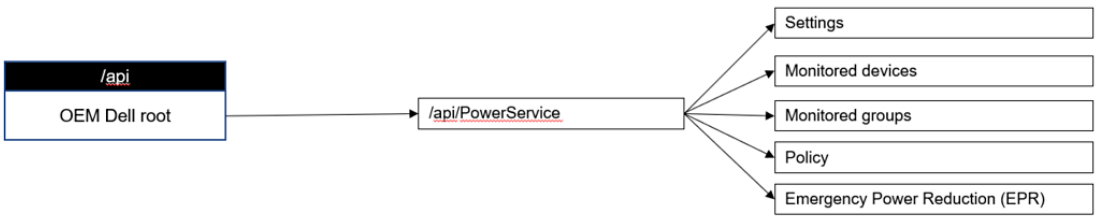


**Figure 1. Resource tree for Power Manager—Modular OEM scheme**

## 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 |

**Table 1. Authentication and authorization requirements (continued)**

| Actions | Authentication required | Authorization required |
|---|---|---|
| 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 | | |
| GROUP_MANAGEMENT | Y | Y | |
| INVENTORY_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.<br>`<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.<br>`<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.<br>`<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.<br>`<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`<br>**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<br>• 204 – OK with no message body<br>• 206 – OK with partial message body |
| POST | • 201 – Resource created (operation complete)<br>• 202 – Resource accepted (operation pending) |
| PUT | • 202 – Accepted (operation pending)<br>• 204 – Success (operation complete) |
| DELETE | • 202 – Accepted (operation pending)<br>• 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 |
| 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 |

**Table 6. Examples of response headers (continued)**

| Response Header | Description | Example |
|---|---|---|
| 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.<br>• *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:<br>• deny – no rendering within a frame<br>• sameorigin – no rendering if origin mismatch<br>• allow-from – allow from specified location<br>• 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:

- 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
            },
            {
                "@odata.type": "#PowerService.Settings",
                "Id": 9,
                "Name": "RESET_WSMAN_POWER_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:

- 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
},
{
    "Id": 9,
    "Value": 2
}
]
}
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 |
| 9 | Reset WSMAN Power 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 |

### Table 16. Reset WSMAN Power Metric Data Enumeration

| Enumeration Value | Description |
|---|---|
| 1 | Enabled |
| 2 | Disabled |

# 5

# Monitored Devices

**Topics:**



## /api/PowerService/Actions/PowerService.AddDevices

This URI is to add the device(s) to Power Manager.

Supported versions of Power Manager:

- 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 17. 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:

- 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 18. 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 19. Filters**

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

Supported versions of Power Manager:

- 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,
            "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 20. 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. |
| 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 21. License Status Enumeration

| Enumeration Value | Description |
|---|---|
| 1 | Active |
| 2 | Expired |
| 3 | Not Applicable |

### Table 22. 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

### Table 23. Enumeration

| Type | State | Enumeration Value |
|---|---|---|
| Health | Critical | 4000 |
| | Warning | 3000 |
| | Normal | 1000 |
| | Unknown | 2000 |
| | | |
| Power | Online | 17 |
| | Offline | 18 |
| | Unknown | 1 |
| | | |
| Connection | Connected | true |

**Table 23. Enumeration (continued)**

| Type | State | Enumeration Value |
|---|---|---|
| | Disconnected | false |
| | | |
| Device Type | Compute | 1000 |
| | Chassis | 2000 |
| | | |
| Managed | Error | 1000 |
| | Managed | 3000 |
| | Monitored | 4000 |
| | Managed with Alerts | 6000 |
| | Proxied | 7000 |

# 6

# Monitored Groups

**Topics:**



# /api/PowerService/Actions/PowerService.AddGroups

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

Supported versions of Power Manager:

- 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:

- 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:

- 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:

- 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. |

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

This URI represents all the Power Manager capable devices.

Supported versions of Power Manager:

- 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 30. 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 31. 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 32. Filters

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

Supported versions of Power Manager:

- 1.1
- 1.0

# GET method for 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"
      "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 33. Attributes

| Attribute name | Description |
|---|---|
| Id | Group Id |
| ParentId | Parent group Id |
| 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 34. Filters

| Field | Operator |
|---|---|
| Id | eq |
| Name | contains |

List of all the sort options that this method supports:

- Name
- DevicesInWorkingSet
- AddedOn

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

This URI represents all the devices of a monitored group

### Table 35. Filters

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

Supported versions of Power Manager:

- 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 36. Attributes

| Attribute name | Description |
|---|---|
| Id | Device Id |
| DeviceName | Device Name |
| ServiceTag | Device Service Tag |

**Table 36. Attributes (continued)**

| Attribute name | Description |
|---|---|
| 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 | 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:

- 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 37. Attributes**

| Attribute | Description |
|---|---|
| EntityId | ID of device. |
| LowerBound | Device Lower power bound. |
| 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 38. Filters**

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

Supported versions of Power Manager:

- 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 39. 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 40. 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 41. Filters

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

### Table 42. Type

| Enumeration Value | Description |
|---|---|
| 1 | Static policy |
| 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:

- 1.1

# 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 43. 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. |

**Table 43. Attributes (continued)**

| Attribute Name | Description |
|---|---|
| PolicyDetails | Link to the detailed view of the policy. |
| IsAssociatedToGroup | The policy is associated to group or not. |

**Table 44. 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:

- 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 45. 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. |
| 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:

- 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 46. 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:

- 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 47. Attributes**

| Attribute name | Description |
|---|---|
| Id | ID of the policy. |
| Type | Type of the policy.<br>**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 48. 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:

- 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 49. 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 will be 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 50. Output Attributes**

| Attribute name | Description |
|---|---|
| Id | Policy Id |

**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:

- 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 51. 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 will be 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 till 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 52. 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 creating 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:

- 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 53. 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:

- 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 54. 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:

- 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 55. 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:

- 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 56. Attributes**

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

### Table 57. EPR Type Enumeration

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

### Table 58. 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:

- 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 59. Attributes

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

### Table 60. EPR Type Enumeration

| Enumeration Value | Description |
|---|---|
| 1 | Power Throttle Down (0 power cap value). |

**Table 60. EPR Type Enumeration (continued)**

| Enumeration Value | Description |
|---|---|
| 2 | Power Off. |

**Table 61. 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:

- 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 62. 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 63. 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 64. 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 65. EprPowerDown

| Value | Description |
|---|---|
| False | Throttle |
| True | Shutdown |

# /api/PowerService/Actions/PowerService.EnableEPR

This URI enables you to enable EPR on the target.

Supported versions of Power Manager:

- 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 66. 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:

- 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 67. Attributes**

| Attribute Name | Description |
|---|---|
| Id | EPR Ids to be disabled. |

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

# 9

# Report Service

**Topics:**

- /api/ReportService/ReportDefs

# /api/ReportService/ReportDefs

This method creates a custom report.

Supported versions of Power Manager:

- 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 68. SettingId Enumeration

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

### Table 69. 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 70. Aggregation Period Enumeration

| Enumeration Value | Description |
|---|---|
| 1 | 1 Hour |
| 2 | 1 Day |

### Table 71. Metric Type Enumeration

| Enumeration Value | Description |
|---|---|
| 3 | Power |
| 7 | Temperature |