Dell OpenManage Power Center REST API Reference Guide

This white paper outlines the usage of the REST API commands in OpenManage Power Center.

Dell Engineering
May 2016

A Dell Technical White Paper

Revisions

Date	Description
September 2014	Initial release
May 2016	Updated with new REST APIs

Table of contents

Executive summary

Dell™ OpenManage™ Power Center (OMPC) management console provides increased visibility over power consumption, anomalies, and utilization in a more precise manner. This enables increased control over data center, improved rack density, faster response times, greater accuracy, and broader decision-making intelligence.

OMPC provides more than 20 Pre-defined templates to generate the reports. This makes it easy for a user to produce any substantive proof or collateral of what the tool monitors. OMPC reporting infrastructure allows you to report on metrics / inventory, to focus on groups / devices as report targets or to sort data.

A Representational State Transfer (REST) Application Program Interface (API) is provided to other applications or scripts using which you can access OMPC report data through a proper authentication bridge.

This document outlines the requirements for using this feature, the integration call procedures, and all types of report supported by OpenManage Power Center.

1 Introduction

This document describes the REST integration with Dell OpenManage Power Center. It provides examples (using python 3.5) of enumerating the device details from OMPC server and gathers the report details through REST APIs. Using these APIs, you can create various reports based on your organization requirement, check reports execution status, or retrieve report results.

2 Key Integration Concepts

2.1 Client Integration Overview

The REST client sends standard HTTP(s) requests to the REST API end- point. Each request is sent using a HTTP verb (for example, PUT, GET, POST, DELETE, HEAD, and OPTIONS) and includes a message body in JSON format.

The response uses a standard HTTP status code. The REST client can be developed in various programming languages (for example, C# and Java) or scripting languages (for example, Python and Perl) that have basic support for HTTPS communication and are capable of constructing and parsing JSON payloads.

2.2 Base URI

The Base URL for REST request:

- https://<HOST_ADDRESS>:<PORT_NUM>/powercenter/api/overview/ (Full form)
- https://<HOST_ADDRESS>:<PORT_NUM>/api/overview/ (Shorter form)

A sample request URL:

https://localhost:8643/api/overview/enumerateEntitySummaries

HOST_ADDRESS	The network address of the machine on which the product is installed. It could be a DNS host name or IP address.
PORT_NUM	The port number that is opened as web application server for OMPC. It should be the port assigned in OMPC installation. Default port is 8643 but use the one that you have configured during the OMPC installation.

The port number can be re-configured, to get the current port that is used, query the value for below registry key:

HKEY_LOCAL_MACHINE\SOFTWARE\Dell\OpenManagePowerCenter\Port

2.3 Security

The REST services will only be exposed through HTTPs to ensure that the common threats associated with HTTP traffic are mitigated.

2.4 Authentication Mechanism

OMPC uses Basic Authentication along with HTTPs to enable authentication of REST requests.

Each request requires an Authorization header in the request, in the format of base-64 encoded string of the combination of following credentials:

- username,
- password,
- accountType: Possible values for this field are:
  - OMPC_ACCOUNT = 0;
  - WIN_LOCAL_ACCOUNT = 1;
  - WIN_DOMAIN_ACCOUNT = 2;
  - LINUX_LOCAL_ACCOUNT = 3;
  - LDAP_ACCOUNT = 4;
- Domain: It is an optional field unless accountType field value is WIN_DOMAIN_ACCOUNT.

Example:

Name=admin, Password=user@123 and AccountType= OMPC_ACCOUNT, no domain specified.

The combination of username, password, and accountType is **admin:user@123:0**, and the base-64 encoded string for this combination is **YWRtaW46dXNlckAxMjM6MA==**

Note:
If the credentials are missing or invalid, a 401 (Authorization failure) error is displayed.
This mechanism is only supported when SSL/TLS is enabled for the transport.

3 Device Enumeration API Interface

3.1 Request Parameters

Name	Type	Description
depth	Integer	Depth of enumeration: -1: Full depth search 1: Search for only immediate child level
path	String	Group path from where enumeration is performed: / Search from root // Search for all devices that are not added to group (Un assigned devices) /// Search for all devices, free or managed /DC/Room1/Row2 Specific group path

Sort Object		Indicates result will be ordered by which field in which order:
		
		"sortObj": [
		{
		"field": "name", ← field to sort
		"order": "1" ← 1 for asc and 2 for desc
		},
		{
		"field": "entityType", ← only first sort field takes effect, this will be ignored
		"order": "1"
		}
		],

Filter Object		Indicates result will be filtered by which field(s) with what kind of condition(s). In case multiple filters are specified, only result(s) that meets all filters could be returned.
		
		"filterObj": [
		{
		"field": "entityType", ← field to filter
		"op": "=", ← condition
		"opTarget": [
		"Server", ← value(s) to match, if multiple items here, relations are "OR"
		]
		},
		{
		"field": "ipAddress", ← another filter
		"op": "=",
		"opTarget": ["10.239.151.133 "]
		}

Name	Type	Description
		] }
Sort Object & Filter Object /Field	String	Indicates target field in sort object and filter object: "assetTag": Asset / service tag "description": Description "entityType": Entity type (See below) "hostname": Host name of the entity "ipAddress": IP address of the entity "link": URL for link and launch "location": Group path of the entity, if the entity belongs to multiple groups, the first path will be returned "name": Name of the entity "serialNumber": Serial number of the entity Valid Entity type values when field = "entityType": "Server" Servers "PDU" PDUs "UPS" UPSs "Chassis" chassis "Unsupported" Unsupported devices "DC" Data centers "ROOM" Rooms "AISLE" Rows "RACK" Racks "CUSTOM" Custom groups
Sort Object /Order	Integer	Sort order, 1:Ascending 2:Descending
Filter Object /Operator	String	Filter operator for comparison: "=" Equals to ">" Greater than "<" Less than ">=" No less than "<=" No greater than

3.2 Sample request

Below request body enumerates from root path with the complete detail, seeking entities whose type is "Server", arranging the result in ascending order by "name".

```
# developed using Python 3.5
import base64
import json
import ssl
import urllib.request
from pprint import pprint
bseurl = 'https://localhost:8643/api/overview/'
encoded_auth = base64.b64encode(b'user:pwd:0', None)
requestdata = b'{"requestObj":{"depth":-1,"path":"/"},' \

b'"criteriaObj":{"paginationObj":{"currentPage":0,"rowCountPerPage":9999,"totalItemsCount":0},' \
                              b'"sortObj":[{"field": "name","order": "1"}],' \
                               b'"filterObj": [{"field": "entityType", "op": "=", "opTarget": ["Server"]}]}}'
if __name__ == "__main__":

    url = bseurl + 'enumerateEntitySummaries'
    req = urllib.request.Request(url, data=requestdata)
    req.add_header('Authorization', encoded_auth)
    req.add_header('Content-Type', 'application/json')
    data = ""
    try:
        cntx = ssl.SSLContext(ssl.PROTOCOL_TLSv1_2)
        with urllib.request.urlopen(req, context=cntx) as f:
            data = json.loads(f.read().decode('utf-8'))
    except Exception as e:
        print('Error:', str(e))

    pprint(data)
    quit()
```

3.3 Sample response body

All Servers under root path are enumerated, result arranged by entity name in ascending order.

Body	Comment
{	
"criteriaObj": {	
"errorObj": null,	
"requestObj": null,	
"responseCode": 0,	
"responseObj": [	Array of entity summaries
{	
"assetTag": "ASSET1234",	Asset Tag
"description": "",	Description
"entityType": "Server",	Entity type
"hostname": "idrac.dell.com.",	Host name for the entity
"ipAddress": "192.168.1.1",	IP address
"link": "https://localhost:8643/index.html#/devices.alldevices/4",	URL can be used to launch the OMPC GUI interface on the target device / group using a browser. When launched, OMPC will authenticate the user through SSO (if configured), reuse of an existing

Body	Comment
	authenticated browser window (if applicable), or the login page and if successful, display the corresponding device filtered in its interface. In case of a missing device, the interface will provide an error message to the user indicating the device could not be found.
"location": "/DC/ROOM/ROW/Rack1/",	Group path of the entity, if the entity belongs to multiple groups, the first path will be returned
"name": "Server - 192.168.1.1",	Name of the entity
"serialNumber": "SRNUM53"	Serial number
"devicePosition": "1"	Device position in rack. If current entity is not a device, this field will be -1.
"size": "3"	Device form factor in U. If current entity is not a device, this field will be -1.
},	
{	
"assetTag": "",	
"description": "",	
"entityType": "Server",	
"hostname": "",	
"ipAddress": "192.168.1.2",	
"link": "https://localhost:8643/index.html#/devices.alldevices/10",	URL for link and launch
"location": "/DC/ROOM/ROW/Rack2/",	
"name": "Server - 192.168.1.2",	
"serialNumber": " SRNUM65"	
"devicePosition": "11"	
"size": "1"	
},	
]	
}	

3.4 Error response

3.4.1 Sample error response body:

Body	Comment
{	
"errorMessage": "",	
"errorObj": {	

"errorMessage": "Request data is empty or the format is invalid.",	Error message
"fldErrors": []	
},	
"requestObj": null,	
"responseCode": -1912339653	Error response code
}	

Error messages / codes:

Error code	Error message
0x8E04033B	Request data is empty or the format is invalid.
0x8E130301	The pagination object is invalid.
0x8E130304	The sort descriptor is invalid.
0x8E130305	The filter descriptor is invalid.
0x8E050306	Invalid depth value to enumerate entities. Valid values are: All levels: -1, Immediate sub level:1.
0x8E050307	Invalid path to enumerate entities.
0x8E12ff04	Unsupported order field name. Check if the OpenManage Power Center server security is compromised.
-	An internal error occurred. Contact Dell Support for help.

4 Report APIs

4.1 Request Parameters

This section introduces the API definition of reporting features for OMPC. Using this set of APIs, users can create various types of reports, check the status of the reports and retrieve the report results when status are ready, or remove target reports.

4.2 Adding Reports

Below APIs could be used to add and run report of a specific type:

#	API	Report
01	<BASE_URL>/powerHoarders	Power Hoarders
02	<BASE_URL>/powerFrugal	Power Frugal
03	<BASE_URL>/powerData	Power Data
04	<BASE_URL>/powerHeadroom	Power Headroom
05	<BASE_URL>/generalInventory	General Inventory
06	<BASE_URL>/powerHoardersRack	Power Hoarders Rack
07	<BASE_URL>/powerFrugalRack	Power Frugal Rack
08	<BASE_URL>/rawMonitoringData	Raw Monitoring Data
09	<BASE_URL>/compareReport	Comparison Report
10	<BASE_URL>/thermalData	Thermal Data
11	<BASE_URL>/powerUtilization	Power Utilization
12	<BASE_URL>/powerThresholdViolations	Power Threshold Violations
13	<BASE_URL>/powerCapViolations	Power Cap Violations
14	<BASE_URL>/powerCapSettings	Power Cap Settings
15	<BASE_URL>/thresholdSettings	Threshold Settings
16	<BASE_URL>/rackFragmentationHoarders	Rack Fragmentation Hoarders
17	<BASE_URL>/rackSpaceHoarders	Rack Space Hoarders
18	<BASE_URL>/rackSpaceFrugal	Rack Space Frugal
19	<BASE_URL>/highestTemperature	Highest Temperature
20	<BASE_URL>/lowestTemperature	Lowest Temperature
21	<BASE_URL>/eventsReport	Events Report
22	<BASE_URL>/pduOutletAssignment	PDU Outlet Assignment
23	<BASE_URL>/pduSensor	PDU Sensor
24	<BASE_URL>/thermalEventPolicy	Thermal Event Policy

4.3 Request fields

Common request fields for Add Report APIs are listed:

Type	Name	Description
Integer	aggPeriodType	Report aggregation type. Valid types are: 1 - HOUR, 2 - DAY, 3 - WEEK, 4 – MONTH

Type	Name	Description
		If aggregation period is disabled, set both aggregation type and value to -1. Otherwise error will be returned. Note:- For "Raw Monitoring" report, if aggregation period is disabled by setting type/value=-1,-1, a default aggregation period of 1 hour will be applied.
Integer	aggPeriodValue	Aggregation period value, depends on aggregation period type. e.g. when aggregation period type = 2, aggregation period value = 1, result data are aggregated on a base of 1 day, and aggregation period type = 1, aggregation period value = 12 indicates an aggregation period of 12 hours. Aggregation period should not be greater than the time range specified by start time/end time, e.g. when the time range specified is 1 day (24 hours) but the aggregation period is 25 hours, error will be returned.
Array	attributes	Attributes setting for the report: Field Name: display name; Field Key: internal name; Required: Could this attribute be included for this report? Mandatory: Is this attribute mandatory or optional? Needed: Will this attribute be displayed? To use default attribute setting please specify empty attributes. To customize attributes for current report type check "Get System Report Type Summary".
String	startDateTime	Target start time of the report data, should be UTC time in below format: "2016-05-01T00:00:00.000Z" Not all reports require start/end date time. If required, should align on hour boundary except for "Events" report (can specify minute and second).
String	endDateTime	Target end time of the report data, should be UTC time in below format: "2016-05-02T00:00:00.000Z" Not all reports require start/end date time. If required, should align on hour boundary except for "Events" report (can specify minute and second).
Integer	maxResults	Maximum result record count allowed, -1 for returning all records
Array	entityList	Target entity list for the report. Entity could be specified by any one of hostname, IP address, and service tag (serial number). Do not specify more than 1 info at the same time.
Array	entityGroupList	Target entity group list for the report. Group is specified by path.
Integer	percentPower	Used in "Power Utilization" report, device(s) with power utilization percentage >= this value could be reported. Value range is 1~100
Array of integers	severity	Used in "Events" report, events with severity level in this array could be reported. Could be combination of 1:Critical; 2:Warning; 3:Informative
Boolean	selectAllDevices	Indicates whether all devices would be selected, "true" for yes and "false" for no. Only applicable for reports that require target devices / groups. Can be combined with "selectAllGroups" flag.

Type	Name	Description
Boolean	selectAllGroups	Indicates whether all devices would be selected, "true" for yes and "false" for no. Only applicable for reports that require target devices / groups. Can be combined with "selectAllDevices" flag.

4.4 Get System Report Type Summary

GetSystemReportTypeSummary returns predefined report template for a specific report type. Referring to **the attributes list** users can customize the attributes for this report type according to below rule:

1. Mandatory attribute's fields should not be changed ("required" and "needed" must be 1);
2. For non-mandatory attribute:
    - If needed=0, required should also be 0 since this field is not relevant to this report;
    - If needed=1, the attribute is relevant to this report type and optional in output; required=1 indicates that this attribute is to be included and required=0 indicates that this attribute is not to be included.

API usage:

POST <BASE_URL>/getSystemReportTypeSummary?reportType=<ReportType>

Parameters:

Type	Name	Description
String	reportType	Target report type, one of: "powerHoarders" "powerFrugal" "powerData" "powerHeadroom" "generalInventory" "powerHoardersRack" "powerFrugalRack" "rawMonitoringData" "compareReport" "thermalData" "powerUtilization" "powerThresholdViolations" "powerCapViolations" "powerCapSettings" "thresholdSettings" "rackFragmentationHoarders" "rackSpaceHoarders" "rackSpaceFrugal" "highestTemperature" "lowestTemperature" "eventsReport" "pduOutletAssignment" "pduSensor" "thermalEventPolicy"

Sample Request:

```
# developed using Python 3.5
import base64
import json
import ssl
import urllib.request
from pprint import pprint

bseurl = 'https://localhost:8643/powercenter/api/report/getSystemReportTypeSummary?reportType=powerHoarders'
encoded_auth = base64.b64encode(b'user:pwd:0', None)

if __name__ == "__main__":

          req = urllib.request.Request(bseurl)
          req.add_header('Authorization', encoded_auth)
          req.add_header('Content-Type', 'application/json')
          data = ""
          try:
                    cntx = ssl.SSLContext(ssl.PROTOCOL_TLSv1_2)
                    with urllib.request.urlopen(req, context=cntx) as f:
                              data = json.loads(f.read().decode('utf-8'))
          except Exception as e:
                    print('Error:', str(e))
                    quit()
          pprint(data)
          quit()
```

4.5 Usage

The usual usage of using Report REST API is listed below:

1. Call getSystemReportTypeSummary to get template of a specified report type;
2. Add & run a report:
   In request header:
   - set authentication header;

   In request body:
   - specify the the attribute list in request body referring to the result of getSystemReportTypeSummary, leave attributes field empty to use default attributes' setting;
   - specify time range (start / end);
   - specify aggregation type and value (if aggregation period allowed);
   - specify specific fields (severity, percentPower) for related report types;

   Send the request and get checkStatus link / report id from the response;
3. Call checkStatus with report id to Check report status till it becomes "Completed" or "Error", fetch the "getReportResult" link;
4. Call getReportResult with report id to get the report result, specify proper page index and row count per page to present the result in pages;
5. In case of error response, try to locate the root cause with the error message / code returned and retry after adjusting input;

4.6 Report Status

CheckStatus will return the status of specific report. The possible report status are:

- Running
- Deleted
- Pending
- Erroneous
- Completed

When the status for a report is “Completed”, the result is displayed.

4.7 Report Result

GetReportResult is used for returning the results of specific report instance from last running.
To avoid large amount of data two parameters for pagination added. A typical combination is currentPage=0 and rowCountPerPage=10 calling this API for the first time. Then onwards increment the current page.

API usage:

POST <BASE_URL>/getReportResult?reportId=<Id>&currentPage=<page>& rowCountPerPage=<count>

4.8 Removing Report

Remove report will delete the target report with specific report id.

API usage:

POST <BASE_URL>/removeReport?reportId=<Id>

5 Reports

5.1 Power Hoarders Report

Below API is used for consuming "Power Hoarders" report.

API usage:

POST <BASE_URL>/powerHoarders

Sample Request:

Used	Body	Description
✓	"aggPeriodType": -1,	Aggregation period disabled. Must be -1
✓	"aggPeriodValue": -1,	Aggregation period disabled. Must be -1
✓	"attributes": [],	Use default of customize attributes
✓	"startDateTime": "2016-05-01T00:00:00.000Z",	Start date time should be earlier than end date time and current date time
✓	"endDateTime": "2016-05-02T00:00:00.000Z",	End date time should fall between start date time and current date time
✓	"maxResults": -1,	Max result record count allowed, -1 for returning all records
✗	"entityList": [],	not applicable
✗	"entityGroupList": [],	not applicable
✗	"percentPower": 1,	not applicable
✗	"severity": [],	not applicable
✗	"selectAllDevices": false,	Selecting all devices not applicable, do not use or set value to false
✗	"selectAllGroups": false,	Selecting all groups not applicable, do not use or set value to false

Sample Code:

```
# developed using Python 3.5
import base64
import datetime
import json
import ssl
import urllib.request
from pprint import pprint

bseurl = 'https://localhost:8643/powercenter/api/report/'
encoded_auth = base64.b64encode(b'user:pwd:0', None)

day = datetime.timedelta(days=1)
reportStartDate = (datetime.datetime.utcnow().date() - day).strftime('%Y-%m-%d') + 'T00:00:00.000Z'
reportEndDate = datetime.datetime.utcnow().date().strftime('%Y-%m-%d') + 'T00:00:00.000Z'

def getdata(reqobj):
    reqobj.add_header('Authorization', encoded_auth)
    reqobj.add_header('Content-Type', 'application/json')
    try:
        cntx = ssl.SSLContext(ssl.PROTOCOL_TLSv1_2)
        with urllib.request.urlopen(reqobj, context=cntx) as f:
            result = json.loads(f.read().decode('utf-8'))
            return result
    except Exception as e:
        print('Error:', str(e))
        quit()
if __name__ == "__main__":

    url = bseurl + 'powerHoarders'
    post_data = str.encode('{"requestObj": '
                '{"description": "Devices which use the most amount of power",'
                '"startDateTime": "' + reportStartDate + '",'
                '"endDateTime": "' + reportEndDate + '",'
                '"aggPeriodType": -1,'
                '"aggPeriodValue": -1,'
                '"maxResults": -1,'
                '"entityList": [],'
                '"entityGroupList": [],'
                '"percentPower": 0,'
                '"severity": [],'
                '"selectAllDevices": false,'
                '"selectAllGroups": false,'
                '"attributes": ['
                '{"fieldName": "Name","key": "name","mandatory": 1,"needed": 1,"required": 1},'
                '{"fieldName": "Rank","key": "rank","mandatory": 0,"needed": 1,"required": 1},'
                '{"fieldName": "Hostname","key": "hostname","mandatory": 1,"needed": 1,"required": 1},'
                '{"fieldName": "Device Type","key": "deviceType","mandatory": 1,"needed": 1,"required": 1},'
                '{"fieldName": "Model","key": "model","mandatory": 1,"needed": 1,"required": 1},'
                '{"fieldName": "Service Tag","key": "serviceTag","mandatory": 1,"needed": 1,"required": 1},'
                '{"fieldName": "Location","key": "location","mandatory": 1,"needed": 1,"required": 1},'
                '{"fieldName": "Average Power","key": "avgPower","mandatory": 1,"needed": 1,"required": 1},'
                '{"fieldName": "Average Temperature","key": "avgTemp","mandatory": 0,"needed": 1,"required": 1},'
                '{"fieldName": "Minimum Temperature","key": "minTemp","mandatory": 0,"needed": 1,"required": 1},'
                '{"fieldName": "Maximum Temperature","key": "maxTemp","mandatory": 0,"needed": 1,"required": 1},'
                '{"fieldName": "EPR Status","key": "eprState","mandatory": 0,"needed": 1,"required": 1},'
                '{"fieldName": "Connection State","key": "connectionState","mandatory": 0,"needed": 1,"required": 1},'
                '{"fieldName": "Health State","key": "healthState","mandatory": 0,"needed": 1,"required": 1},'
                '{"fieldName": "Time Stamp","key": "timeStamp","mandatory": 0,"needed": 1,"required": 1},'
                '{"fieldName": "Device Ip","key": "deviceIp","mandatory": 0,"needed": 1,"required": 1},'
                '{"fieldName": "Memory Power Consumption","key": "memPowerConsumption","mandatory": 0,"needed": 1,"required": 1},'
                '{"fieldName": "CPU Power Consumption","key": "cpuPowerConsumption","mandatory": 0,"needed": 1,"required": 1},'
                '{"fieldName": "CPU Utilization","key": "cpuUtilization","mandatory": 0,"needed": 1,"required": 1},'
                '{"fieldName": "Memory Bandwidth Utilization","key": "memUtilization","mandatory": 0,"needed": 1,"required": 1},'
                '{"fieldName": "I/O Utilization","key": "ioUtilization","mandatory": 0,"needed": 1,"required": 1}]}}')

    reportstatus = getdata(urllib.request.Request(url, data=post_data))
    statusUrl = ""
    if  reportstatus['errorObj'] is None:
        statusUrl =  reportstatus['responseObj']['link']
    else:
        print( reportstatus['errorObj'])
        quit()
    reportresult = getdata(urllib.request.Request(statusUrl))
    while  reportresult['responseObj']['message'] != 'Completed':
         reportresult = getdata(urllib.request.Request(statusUrl))
    if  reportresult['responseObj']['message'] == 'Error':
        print('Error while creating the report')
        quit()
    resultUrl =  reportresult['responseObj']['link']
    resultUrl += '&currentPage=0&rowCountPerPage=1000'
    reportdata = getdata(urllib.request.Request(resultUrl))
    pprint(reportdata)
    quit()
```

Note: For rest of the reports only url and post_data will be provided in sample code section.

5.2 Power Frugal Report

Below API is used for consuming “Power Frugal” report.

API usage:

POST <BASE_URL>/powerFrugal

Sample Request:

Used	Body	Description
✓	"aggPeriodType": -1,	Aggregation period disabled. Must be -1
✓	"aggPeriodValue": -1,	Aggregation period disabled. Must be -1
✓	"attributes": [],	Use default of customize attributes
✓	"startDateTime": "2016-05-01T00:00:00.000Z",	Start date time should be earlier than end date time and current date time
✓	"endDateTime": "2016-05-02T00:00:00.000Z",	End date time should fall between start date time and current date time
✓	"maxResults": -1,	Max result record count allowed, -1 for returning all records
✗	"entityList": [],	not applicable
✗	"entityGroupList": [],	not applicable
✗	"percentPower": 1,	not applicable
✗	"severity": [],	not applicable
✗	“selectAllDevices”: false,	Selecting all devices not applicable, do not use or set value to false
✗	“selectAllGroups”: false,	Selecting all groups not applicable, do not use or set value to false

Sample Code:

```
    url = bseurl + 'powerFrugal'
    post_data = str.encode(
        '{"requestObj": '
        '{"description": "Devices which use the least amount of power(lowest average)",'
        '"startDateTime": "' + reportStartDate + '",'
        '"endDateTime": "' + reportEndDate + '",'
        '"aggPeriodType": -1,'
        '"aggPeriodValue": -1,'
        '"maxResults": -1,'
        '"entityList": [],'
        '"entityGroupList": [],'
        '"percentPower": 0,'
        '"severity": [],'
        '"selectAllDevices": false,'
        '"selectAllGroups": false,'
        '"attributes": ['
        '{"fieldName": "Name","key": "name","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Rank","key": "rank","mandatory": 0,"needed": 1,"required": 1},'
        '{"fieldName": "Hostname","key": "hostname","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Device Type","key": "deviceType","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Model","key": "model","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Service Tag","key": "serviceTag","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Location","key": "location","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Average Power","key": "avgPower","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Average Temperature","key": "avgTemp","mandatory": 0,"needed": 1,"required": 1},'
        '{"fieldName": "Minimum Temperature","key": "minTemp","mandatory": 0,"needed": 1,"required": 1},'
        '{"fieldName": "Maximum Temperature","key": "maxTemp","mandatory": 0,"needed": 1,"required": 1},'
        '{"fieldName": "EPR Status","key": "eprState","mandatory": 0,"needed": 1,"required": 1},'
        '{"fieldName": "Connection State","key": "connectionState","mandatory": 0,"needed": 1,"required": 1},'
        '{"fieldName": "Health State","key": "healthState","mandatory": 0,"needed": 1,"required": 1},'
```

```
        '{"fieldName": "Time Stamp","key": "timeStamp","mandatory": 0,"needed": 1,"required": 1},'
        '{"fieldName": "Device Ip","key": "deviceIp","mandatory": 0,"needed": 1,"required": 1},'
        '{"fieldName": "Memory Power Consumption","key": "memPowerConsumption","mandatory": 0,"needed":
1,"required": 1},'
        '{"fieldName": "CPU Power Consumption","key": "cpuPowerConsumption","mandatory": 0,"needed": 1,"required":
1},'
        '{"fieldName": "CPU Utilization","key": "cpuUtilization","mandatory": 0,"needed": 1,"required": 1},'
        '{"fieldName": "Memory Bandwidth Utilization","key": "memUtilization","mandatory": 0,"needed":
1,"required": 1},'
        '{"fieldName": "I/O Utilization","key": "ioUtilization","mandatory": 0,"needed": 1,"required": 1}]}}')
```

5.3 Power Data Report

Below API is used for consuming "Power Data" report.

API usage:

POST <BASE_URL>/powerData

Sample Request:

Used	Body	Description
✓	"aggPeriodType": 1,	Aggregation period could be enabled.
✓	"aggPeriodValue": 1,	Aggregation period could be enabled.
✓	"attributes": [],	Use default of customize attributes
✓	"startDateTime": "2016-05-01T00:00:00.000Z",	Start date time should be earlier than end date time and current date time
✓	"endDateTime": "2016-05-02T00:00:00.000Z",	End date time should fall between start date time and current date time
✓	"maxResults": -1,	Max result record count allowed, -1 for returning all records
✓	"entityList": [{ "hostName": "idrac1-dell.com", "ipAddress": "", "serviceTagOrSerialNum": ""}],	Specify target entity list. If "selectAllDevices" or "selectAllGroups" is true, this field is ignored, keep it empty. If both "selectAllDevices" and "selectAllGroups" are false, must provide non-empty entity group list and/or entity list. Providing empty list or invalid group / entity will result in error.
✓	"entityGroupList": [{"groupPath": "/DC" }],	Specify target entity group list. If "selectAllDevices" or "selectAllGroups" is true, this field is ignored, keep it empty. If both "selectAllDevices" and "selectAllGroups" are false, must provide non-empty entity group list and/or entity list. Providing empty list or invalid group / entity will result in error.
✗	"percentPower": 1,	not applicable
✗	"severity": [],	not applicable
✓	"selectAllDevices": false,	"true" to select all devices
✓	"selectAllGroups": false,	"true" to select all groups

Sample Code:

```
    url = bseurl + 'powerData'
    post_data = str.encode(
        '{"requestObj": '
        '{"description": "Device/Group MIN/MAX/average and instantaneous power numbers",'
        '"startDateTime": "' + reportStartDate + '",'
        '"endDateTime": "' + reportEndDate + '",'
        '"aggPeriodType": -1,'
        '"aggPeriodValue": -1,'
        '"maxResults": -1,'
        '"entityList": [{"hostName": "","ipAddress": "192.168.2.1","serviceTagOrSerialNum": ""},'
        '{"hostName": "","ipAddress": "192.168.2.2","serviceTagOrSerialNum": ""}],'
        '"entityGroupList": [{"groupPath": "/DC_001/Room_001/Row(100)/Rack(100)"}],'
        '"percentPower": 0,'
        '"severity": [],'
        '"selectAllDevices": false,'
        '"selectAllGroups": false,'
        '"attributes": ['
        '{"fieldName": "Name","key": "name","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Hostname","key": "hostname","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Device Type","key": "deviceType","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Model","key": "model","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Service Tag","key": "serviceTag","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Location","key": "location","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Average Power","key": "avgPower","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Instantaneous Power","key": "instantPower","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Minimum Power","key": "minPower","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Maximum Power","key": "maxPower","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Connection State","key": "connectionState","mandatory": 0,"needed": 1,"required": 1},'
        '{"fieldName": "Health State","key": "healthState","mandatory": 0,"needed": 1,"required": 1},'
        '{"fieldName": "Time Stamp","key": "timeStamp","mandatory": 0,"needed": 1,"required": 1},'
        '{"fieldName": "Device Ip","key": "deviceIp","mandatory": 0,"needed": 1,"required": 1},'
        '{"fieldName": "Memory Power Consumption","key": "memPowerConsumption","mandatory": 0,"needed": 1,"required":
1},'
        '{"fieldName": "CPU Power Consumption","key": "cpuPowerConsumption","mandatory": 0,"needed": 1,"required": 1},'
        '{"fieldName": "CPU Utilization","key": "cpuUtilization","mandatory": 0,"needed": 1,"required": 1},'
        '{"fieldName": "Memory Bandwidth Utilization","key": "memUtilization","mandatory": 0,"needed": 1,"required":
1},'
        '{"fieldName": "I/O Utilization","key": "ioUtilization","mandatory": 0,"needed": 1,"required": 1}]}}')
```

5.4 Power Headroom Report

Below API is used for consuming "Power Headroom" report.

API usage:

POST <BASE_URL>/PowerHeadroom

Sample Request:

Used	Body	Description
✓	"aggPeriodType": 1,	Aggregation period could be enabled.
✓	"aggPeriodValue": 1,	Aggregation period could be enabled.
✓	"attributes": [],	Use default of customize attributes
✓	"startDateTime": "2016-05-01T00:00:00.000Z",	Start date time should be earlier than end date time and current date time
✓	"endDateTime": "2016-05-02T00:00:00.000Z",	End date time should fall between start date time and current date time
✓	"maxResults": -1,	Max result record count allowed, -1 for returning all records
✗	"entityList": [],	not applicable

✓	"entityGroupList": [
	{
	"groupPath": "/DC"
	}
	],
		Specify target entity group list.
		If "selectAllGroups" is true, this field is ignored, keep it empty.
		If "selectAllGroups" is false, must provide non-empty entity group list. Providing empty list or invalid group / entity will result in error.

✗	"percentPower": 1,	not applicable
✗	"severity": [],	not applicable
✗	"selectAllDevices": false,	not applicable
✓	"selectAllGroups": false,	"true" to select all groups

Sample Code:

```
    url = bseurl + 'powerHeadroom'
    post_data = str.encode(
        '{"requestObj": '
        '{"description": "Overall power usage and available headroom",'
        '"startDateTime": "' + reportStartDate + '",'
        '"endDateTime": "' + reportEndDate + '",'
        '"aggPeriodType": -1,'
        '"aggPeriodValue": -1,'
        '"maxResults": -1,'
        '"entityList": [],'
        '"entityGroupList": [{"groupPath": "/DC_001/Room_001/Row(100)/Rack(100)"}],'
        '"percentPower": 0,'
        '"severity": [],'
        '"selectAllDevices": false,'
        '"selectAllGroups": false,'
        '"attributes": ['
        '{"fieldName": "Name","key": "name","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Location","key": "location","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Average Power","key": "avgPower","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Time Stamp","key": "timeStamp","mandatory": 0,"needed": 1,"required": 1},'
        '{"fieldName": "Stranded Power","key": "strandedPower","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Allocated Power","key": "allocatedPower","mandatory": 0,"needed": 1,"required": 1},'
        '{"fieldName": "Peak Power Consumed","key": "consumedPower","mandatory": 0,"needed": 1,"required": 1},'
        '{"fieldName": "Average Temperature","key": "avgTemp","mandatory": 0,"needed": 1,"required": 1},'
        '{"fieldName": "Maximum Temperature","key": "maxTemp","mandatory": 0,"needed": 1,"required": 1},'
        '{"fieldName": "Minimum Temperature","key": "minTemp","mandatory": 0,"needed": 1,"required": 1}]}}')
```

5.5 General Inventory Report

Below API is used for consuming "General Inventory" report.

API usage:

POST <BASE_URL>/generalInventory

Sample Request:

Used	Body	Description
✓	"aggPeriodType": -1,	Aggregation period disabled. Must be -1
✓	"aggPeriodValue": -1,	Aggregation period disabled. Must be -1
✓	"attributes": [],	Use default of customize attributes
✓	"startDateTime": "2016-05-01T00:00:00.000Z",	Start date time should be earlier than end date time and current date time

✓	"endDateTime": "2016-05-02T00:00:00.000Z",	End date time should fall between start date time and current date time
✓	"maxResults": -1,	Max result record count allowed, -1 for returning all records

✓	"entityList": [
	{
	"hostName": "idrac1-dell.com",
	"ipAddress": "",
	"serviceTagOrSerialNum": ""
	}
	],
		Specify target entity list.
		
		If "selectAllDevices" or "selectAllGroups" is true, this field is ignored, keep it empty.
		If both "selectAllDevices" and "selectAllGroups" are false, must provide non-empty entity group list and/or entity list. Providing empty list or invalid group / entity will result in error.

✓	"entityGroupList": [
	{
	"groupPath": "/DC"
	}
	],
		Specify target entity group list.
		
		If "selectAllDevices" or "selectAllGroups" is true, this field is ignored, keep it empty.
		If both "selectAllDevices" and "selectAllGroups" are false, must provide non-empty entity group list and/or entity list. Providing empty list or invalid group / entity will result in error.

✗	"percentPower": 1,	not applicable
✗	"severity": [],	not applicable
✓	"selectAllDevices": false,	"true" to select all devices
✓	"selectAllGroups": false,	"true" to select all groups

Sample Code:

```
    url = bseurl + 'generalInventory'
    post_data = str.encode(
        '{"requestObj": '
        '{"description": "General inventory report",'
        '"startDateTime": "' + reportStartDate + '",'
        '"endDateTime": "' + reportEndDate + '",'
        '"aggPeriodType": -1,'
        '"aggPeriodValue": -1,'
        '"maxResults": -1,'
        '"entityList": [{"hostName": "","ipAddress": "192.168.2.1","serviceTagOrSerialNum": ""},'
        '{"hostName": "","ipAddress": "192.168.2.2","serviceTagOrSerialNum": ""}],'
        '"entityGroupList": [{"groupPath": "/BDC_001/Room_001/Row(100)/Rack(100)"}],'
        '"percentPower": 0,'
        '"severity": [],'
        '"selectAllDevices": false,'
        '"selectAllGroups": false,'
        '"attributes": ['
        '{"fieldName": "Name","key": "name","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Hostname","key": "hostname","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Device Type","key": "deviceType","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Model","key": "model","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Service Tag","key": "serviceTag","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Location","key": "location","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Average Power","key": "avgPower","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Instantaneous Power","key": "instantPower","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Minimum Power","key": "minPower","mandatory": 0,"needed": 1,"required": 1},'
        '{"fieldName": "Maximum Power","key": "maxPower","mandatory": 0,"needed": 1,"required": 1},'
        '{"fieldName": "Connection State","key": "connectionState","mandatory": 0,"needed": 1,"required": 1},'
        '{"fieldName": "Health State","key": "healthState","mandatory": 0,"needed": 1,"required": 1},'
        '{"fieldName": "Time Stamp","key": "timeStamp","mandatory": 0,"needed": 1,"required": 1},'
        '{"fieldName": "Device Ip","key": "deviceIp","mandatory": 0,"needed": 1,"required": 1},'
        '{"fieldName": "Memory Power Consumption","key": "memPowerConsumption","mandatory": 0,"needed": 1,"required":
1},'
        '{"fieldName": "CPU Power Consumption","key": "cpuPowerConsumption","mandatory": 0,"needed": 1,"required":
1},'
        '{"fieldName": "CPU Utilization","key": "cpuUtilization","mandatory": 0,"needed": 1,"required": 1},'
        '{"fieldName": "Memory Bandwidth Utilization","key": "memUtilization","mandatory": 0,"needed": 1,"required":
1},'
        '{"fieldName": "I/O Utilization","key": "ioUtilization","mandatory": 0,"needed": 1,"required": 1},'
        '{"fieldName": "Stranded Power","key": "strandedPower","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Allocated Power","key": "allocatedPower","mandatory": 0,"needed": 1,"required": 1},'
        '{"fieldName": "Peak Power Consumed","key": "consumedPower","mandatory": 0,"needed": 1,"required": 1},'
        '{"fieldName": "Average Temperature","key": "avgTemp","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Maximum Temperature","key": "maxTemp","mandatory": 0,"needed": 1,"required": 1},'
        '{"fieldName": "Minimum Temperature","key": "minTemp","mandatory": 0,"needed": 1,"required": 1}]}}')
```

5.6 Power Hoarders Rack Report

Below API is used for consuming "Power Hoarders Rack" report.

API usage:

POST <BASE_URL>/powerHoardersRack

Sample Request:

Used	Body	Description
✓	"aggPeriodType": -1,	Aggregation period disabled. Must be -1
✓	"aggPeriodValue": -1,	Aggregation period disabled. Must be -1
✓	"attributes": [],	Use default of customize attributes
✓	"startDateTime": "2016-05-01T00:00:00.000Z",	Start date time should be earlier than end date time and current date time
✓	"endDateTime": "2016-05-02T00:00:00.000Z",	End date time should fall between start date time and current date time
✓	"maxResults": -1,	Max result record count allowed, -1 for returning all records

✗	"entityList": [],	not applicable
✗	"entityGroupList": [],	not applicable
✗	"percentPower": 1,	not applicable
✗	"severity": [],	not applicable
✗	“selectAllDevices” : false,	Selecting all devices not applicable, do not use or set value to false
✗	“selectAllGroups” : false,	Selecting all devices not applicable, do not use or set value to false

Sample Code:

```
    url = bseurl + 'powerHoardersRack'
    post_data = str.encode(
        '{"requestObj": '
        '{"description": "Racks which have highest utilization of power",'
        '"startDateTime": "' + reportStartDate + '",'
        '"endDateTime": "' + reportEndDate + '",'
        '"aggPeriodType": -1,'
        '"aggPeriodValue": -1,'
        '"maxResults": -1,'
        '"entityList": [],'
        '"entityGroupList": [],'
        '"percentPower": 0,'
        '"severity": [],'
        '"selectAllDevices": false,'
        '"selectAllGroups": false,'
        '"attributes": ['
        '{"fieldName": "Name","key": "name","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Rank","key": "rank","mandatory": 0,"needed": 1,"required": 1},'
        '{"fieldName": "Location","key": "location","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Average Power","key": "avgPower","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Stranded Power","key": "strandedPower","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Allocated Power","key": "allocatedPower","mandatory": 0,"needed": 1,"required": 1},'
        '{"fieldName": "Peak Power Consumed","key": "consumedPower","mandatory": 0,"needed": 1,"required": 1}]}}')
```

5.7 Power Frugal Rack Report

Below API is used for consuming “Power Frugal Rack” report.

API usage:

POST <BASE_URL>/powerFrugalRack

Sample Request:

Used	Body	Description
✓	"aggPeriodType": -1,	Aggregation period disabled. Must be -1
✓	"aggPeriodValue": -1,	Aggregation period disabled. Must be -1
✓	"attributes": [],	Use default of customize attributes
✓	"startDateTime": "2016-05-01T00:00:00.000Z",	Start date time should be earlier than end date time and current date time
✓	"endDateTime": "2016-05-02T00:00:00.000Z",	End date time should fall between start date time and current date time
✓	"maxResults": -1,	Max result record count allowed, -1 for returning all records
✗	"entityList": [],	not applicable
✗	"entityGroupList": [],	not applicable

✗	"percentPower": 1,	not applicable
✗	"severity": [],	not applicable
✗	“selectAllDevices” : false,	Selecting all devices not applicable, do not use or set value to false
✗	“selectAllGroups” : false,	Selecting all devices not applicable, do not use or set value to false

Sample Code:

```
    url = bseurl + 'powerFrugalRack'
    post_data = str.encode(
        '{"requestObj": '
        '{"description": "Racks which have lowest utilization of power",'
        '"startDateTime": "' + reportStartDate + '",'
        '"endDateTime": "' + reportEndDate + '",'
        '"aggPeriodType": -1,'
        '"aggPeriodValue": -1,'
        '"maxResults": -1,'
        '"entityList": [],'
        '"entityGroupList": [],'
        '"percentPower": 0,'
        '"severity": [],'
        '"selectAllDevices": false,'
        '"selectAllGroups": false,'
        '"attributes": ['
        '{"fieldName": "Name","key": "name","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Rank","key": "rank","mandatory": 0,"needed": 1,"required": 1},'
        '{"fieldName": "Location","key": "location","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Average Power","key": "avgPower","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Stranded Power","key": "strandedPower","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Allocated Power","key": "allocatedPower","mandatory": 0,"needed": 1,"required": 1},'
        '{"fieldName": "Peak Power Consumed","key": "consumedPower","mandatory": 0,"needed": 1,"required": 1}]}}')
```

5.8 Raw Monitoring Data Report

Below API is used for consuming “Raw Monitoring Data” report.

API usage:

POST <BASE_URL>/rawMonitoringData

Sample Request:

Used	Body	Description
✓	"aggPeriodType": -1,	Aggregation period could be enabled.
✓	"aggPeriodValue": -1,	Aggregation period could be enabled.
✓	"attributes": [],	Use default of customize attributes
✓	"startDateTime": "2016-05-01T00:00:00.000Z",	Start date time should be earlier than end date time and current date time
✓	"endDateTime": "2016-05-02T00:00:00.000Z",	End date time should fall between start date time and current date time
✓	"maxResults": -1,	Max result record count allowed, -1 for returning all records
✓	"entityList": [{ "hostName": "idrac1-dell.com", "ipAddress": "", "serviceTagOrSerialNum": ""}],	Specify target entity list. If "selectAllDevices" is true, this field is ignored, keep it empty.

		If "selectAllDevices" is false, must provide non-empty entity list. Providing empty list or invalid entity will result in error.
✘	"entityGroupList": [],	not applicable
✘	"percentPower": 1,	not applicable
✘	"severity": [],	not applicable
✓	“selectAllDevices” : false,	“true” to select all devices
✘	“selectAllGroups” : false,	not applicable, do not use or set value to false

Sample Code:

```
    url = bseurl + 'rawMonitoringData'
    post_data = str.encode(
        '{"requestObj": '
        '{"description": "Raw monitoring data",'
        '"startDateTime": "' + reportStartDate + '",'
        '"endDateTime": "' + reportEndDate + '",'
        '"aggPeriodType": -1,'
        '"aggPeriodValue": -1,'
        '"maxResults": -1,'
        '"entityList": [{"hostName": "","ipAddress": "192.168.2.1","serviceTagOrSerialNum": ""},'
        '{"hostName": "","ipAddress": "192.168.2.2","serviceTagOrSerialNum": ""}],'
        '"entityGroupList": [],'
        '"percentPower": 0,'
        '"severity": [],'
        '"selectAllDevices": false,'
        '"selectAllGroups": false,'
        '"attributes": ['
        '{"fieldName": "Name","key": "name","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Hostname","key": "hostname","mandatory": 0,"needed": 1,"required": 1},'
        '{"fieldName": "Device Type","key": "deviceType","mandatory": 0,"needed": 1,"required": 1},'
        '{"fieldName": "Model","key": "model","mandatory": 0,"needed": 1,"required": 1},'
        '{"fieldName": "Service Tag","key": "serviceTag","mandatory": 0,"needed": 1,"required": 1},'
        '{"fieldName": "Location","key": "location","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Average Power","key": "avgPower","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Instantaneous Power","key": "instantPower","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Minimum Power","key": "minPower","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Maximum Power","key": "maxPower","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Connection State","key": "connectionState","mandatory": 0,"needed": 1,"required": 1},'
        '{"fieldName": "Health State","key": "healthState","mandatory": 0,"needed": 1,"required": 1},'
        '{"fieldName": "EPR Status","key": "eprState","mandatory": 0,"needed": 1,"required": 1},'
        '{"fieldName": "Time Stamp","key": "timeStamp","mandatory": 0,"needed": 1,"required": 1},'
        '{"fieldName": "Device Ip","key": "deviceIp","mandatory": 0,"needed": 1,"required": 1},'
        '{"fieldName": "Memory Power Consumption","key": "memPowerConsumption","mandatory": 0,"needed": 1,"required":
1},'
        '{"fieldName": "CPU Power Consumption","key": "cpuPowerConsumption","mandatory": 0,"needed": 1,"required":
1},'
        '{"fieldName": "CPU Utilization","key": "cpuUtilization","mandatory": 0,"needed": 1,"required": 1},'
        '{"fieldName": "Memory Bandwidth Utilization","key": "memUtilization","mandatory": 0,"needed": 1,"required":
1},'
        '{"fieldName": "I/O Utilization","key": "ioUtilization","mandatory": 0,"needed": 1,"required": 1},'
        '{"fieldName": "Average Temperature","key": "avgTemp","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Maximum Temperature","key": "maxTemp","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Minimum Temperature","key": "minTemp","mandatory": 1,"needed": 1,"required": 1}]}}')
```

5.9 Comparison Report

Below API is used for consuming "Comparison" report.

API usage:

POST <BASE_URL>/compareReport

Sample Request:

Used	Body	Description
✓	"aggPeriodType": -1,	Aggregation period could be enabled.
✓	"aggPeriodValue": -1,	Aggregation period could be enabled.
✓	"attributes": [],	Use default of customize attributes
✓	"startDateTime": "2016-05-01T00:00:00.000Z",	Start date time should be earlier than end date time and current date time
✓	"endDateTime": "2016-05-02T00:00:00.000Z",	End date time should fall between start date time and current date time
✓	"maxResults": -1,	Max result record count allowed, -1 for returning all records
✓	"entityList": [{"hostName": "idrac1-dell.com", "ipAddress": "","serviceTagOrSerialNum": ""}, {"hostName": "idrac2-dell.com","ipAddress": "", "serviceTagOrSerialNum": ""}],	Must provide non-empty entity group list and/or entity list. Providing empty list or invalid group / entity will result in error. Total number of entities / groups should be no less than 2 and to a max of 3.
✓	"entityGroupList": [{"groupPath": "/DC1"}, {"groupPath": "/DC2"}],	Must provide non-empty entity group list and/or entity list. Providing empty list or invalid group / entity will result in error. Total number of entities / groups should be no less than 2 and to a max of 3.
✗	"percentPower": 1,	not applicable
✗	"severity": [],	not applicable
✗	"selectAllDevices" : false,	not applicable, do not use or set value to false
✗	"selectAllGroups" : false,	not applicable, do not use or set value to false

Sample Code:

```
    url = bseurl + 'compareReport'
    post_data = str.encode(
        '{"requestObj": '
        '{"description": "Device/Group comparison report",'
        '"startDateTime": "' + reportStartDate + '",'
        '"endDateTime": "' + reportEndDate + '",'
        '"aggPeriodType": -1,'
        '"aggPeriodValue": -1,'
        '"maxResults": -1,'
        '"entityList": [{"hostName": "","ipAddress": "192.168.2.1","serviceTagOrSerialNum": ""},'
        '{"hostName": "","ipAddress": "192.168.2.2","serviceTagOrSerialNum": ""},'
        '{"hostName": "","ipAddress": "192.168.2.3","serviceTagOrSerialNum": ""}],'
        '"entityGroupList": [],'
        '"percentPower": 0,'
        '"severity": [],'
        '"selectAllDevices": false,'
        '"selectAllGroups": false,'
        '"attributes": ['
        '{"fieldName": "Name","key": "name","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Hostname","key": "hostname","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Device Type","key": "deviceType","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Model","key": "model","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Service Tag","key": "serviceTag","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Location","key": "location","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Average Power","key": "avgPower","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Instantaneous Power","key": "instantPower","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Minimum Power","key": "minPower","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Maximum Power","key": "maxPower","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Peak Power","key": "peakPower","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "IT Energy","key": "energyIt","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "IT Energy Cost","key": "energyItCost","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Cooling Energy","key": "energyCooling","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Cooling Energy Cost","key": "energyCoolingCost","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Total Energy","key": "energyTotal","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Total Energy Cost","key": "energyTotalCost","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Memory Power Consumption","key": "memPowerConsumption","mandatory": 0,"needed":
1,"required": 1},'
        '{"fieldName": "CPU Power Consumption","key": "cpuPowerConsumption","mandatory": 0,"needed": 1,"required":
1},'
        '{"fieldName": "CPU Utilization","key": "cpuUtilization","mandatory": 0,"needed": 1,"required": 1},'
        '{"fieldName": "Memory Bandwidth Utilization","key": "memUtilization","mandatory": 0,"needed":
1,"required": 1},'
        '{"fieldName": "I/O Utilization","key": "ioUtilization","mandatory": 0,"needed": 1,"required": 1},'
        '{"fieldName": "Average Temperature","key": "avgTemp","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Maximum Temperature","key": "maxTemp","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Minimum Temperature","key": "minTemp","mandatory": 1,"needed": 1,"required": 1}]}}')
```

5.10 Thermal Data Report

Below API is used for consuming “Thermal Data” report.

API usage:

POST <BASE_URL>/thermalData

Sample Request:

Used	Body	Description
✓	"aggPeriodType": -1,	Aggregation period disabled. Must be -1
✓	"aggPeriodValue": -1,	Aggregation period disabled. Must be -1
✓	"attributes": [],	Use default of customize attributes
✓	"startDateTime": "2016-05-01T00:00:00.000Z",	Start date time should be earlier than end date time and current date time
✓	"endDateTime": "2016-05-02T00:00:00.000Z",	End date time should fall between start date time and current date time
✓	"maxResults": -1,	Max result record count allowed, -1 for returning all records

✓	"entityList": [{ "hostName": "idrac1-dell.com", "ipAddress": "", "serviceTagOrSerialNum": ""}],	Specify target entity list. If "selectAllDevices" or "selectAllGroups" is true, this field is ignored, keep it empty. If both "selectAllDevices" and "selectAllGroups" are false, must provide non-empty entity group list and/or entity list. Providing empty list or invalid group / entity will result in error.
✓	"entityGroupList": [{"groupPath": "/DC1"}, {"groupPath": "/DC2"}],	Specify target entity group list. If "selectAllDevices" or "selectAllGroups" is true, this field is ignored, keep it empty. If both "selectAllDevices" and "selectAllGroups" are false, must provide non-empty entity group list and/or entity list. Providing empty list or invalid group / entity will result in error.
✗	"percentPower": 1,	not applicable
✗	"severity": [],	not applicable
✓	"selectAllDevices" : false,	"true" to select all devices
✓	"selectAllGroups" : false,	"true" to select all groups

Sample Code:

```
url = bseurl + 'thermalData'
post_data = str.encode(
    '{"requestObj": '
    '{"description": "Device/Group MIN/MAX/average temperature values",'
    '"startDateTime": "' + reportStartDate + '",'
    '"endDateTime": "' + reportEndDate + '",'
    '"aggPeriodType": -1,'
    '"aggPeriodValue": -1,'
    '"maxResults": -1,'
    '"entityList": [{"hostName": "","ipAddress": "192.168.2.1","serviceTagOrSerialNum": ""},'
    '{"hostName": "","ipAddress": "192.168.2.2","serviceTagOrSerialNum": ""}],'
    '"entityGroupList": [],'
    '"percentPower": 0,'
    '"severity": [],'
    '"selectAllDevices": false,'
    '"selectAllGroups": false,'
    '"attributes": ['
    '{"fieldName": "Name","key": "name","mandatory": 1,"needed": 1,"required": 1},'
    '{"fieldName": "Hostname","key": "hostname","mandatory": 1,"needed": 1,"required": 1},'
    '{"fieldName": "Device Type","key": "deviceType","mandatory": 1,"needed": 1,"required": 1},'
    '{"fieldName": "Model","key": "model","mandatory": 1,"needed": 1,"required": 1},'
    '{"fieldName": "Service Tag","key": "serviceTag","mandatory": 1,"needed": 1,"required": 1},'
    '{"fieldName": "Location","key": "location","mandatory": 1,"needed": 1,"required": 1},'
    '{"fieldName": "Connection State","key": "connectionState","mandatory": 0,"needed": 1,"required": 1},'
    '{"fieldName": "Health State","key": "healthState","mandatory": 0,"needed": 1,"required": 1},'
    '{"fieldName": "Time Stamp","key": "timeStamp","mandatory": 0,"needed": 1,"required": 1},'
    '{"fieldName": "Device Ip","key": "deviceIp","mandatory": 0,"needed": 1,"required": 1},'
    '{"fieldName": "Average Temperature","key": "avgTemp","mandatory": 1,"needed": 1,"required": 1},'
    '{"fieldName": "Maximum Temperature","key": "maxTemp","mandatory": 1,"needed": 1,"required": 1},'
    '{"fieldName": "Minimum Temperature","key": "minTemp","mandatory": 1,"needed": 1,"required": 1}]}}')
```

5.11 Power Utilization Report

Below API is used for consuming " Power Utilization" report.

API usage:

POST <BASE_URL>/powerUtilization

Sample Request:

Used	Body	Description
✓	"aggPeriodType": -1,	Aggregation period disabled. Must be -1
✓	"aggPeriodValue": -1,	Aggregation period disabled. Must be -1
✓	"attributes": [],	Use default of customize attributes
✓	"startDateTime": "2016-05-01T00:00:00.000Z",	Start date time should be earlier than end date time and current date time
✓	"endDateTime": "2016-05-02T00:00:00.000Z",	End date time should fall between start date time and current date time
✓	"maxResults": -1,	Max result record count allowed, -1 for returning all records
✗	"entityList": [],	not applicable
✗	"entityGroupList": [],	not applicable
✓	"percentPower": 50,	Specify a value between 1 and 100.
✗	"severity": [],	not applicable
✗	"selectAllDevices" : false,	not applicable, do not use or set value to false
✗	"selectAllGroups" : false,	not applicable, do not use or set value to false

Sample Code:

```
    url = bseurl + 'powerUtilization'
    post_data = str.encode(
        '{"requestObj": '
        '{"description": "Devices that use at least specific percent of their power",'
        '"startDateTime": "' + reportStartDate + '",'
        '"endDateTime": "' + reportEndDate + '",'
        '"aggPeriodType": -1,'
        '"aggPeriodValue": -1,'
        '"maxResults": -1,'
        '"entityList": [],'
        '"entityGroupList": [],'
        '"percentPower": 50,'
        '"severity": [],'
        '"selectAllDevices": false,'
        '"selectAllGroups": false,'
        '"attributes": ['
        '{"fieldName": "Rank","key": "rank","mandatory": 0,"needed": 1,"required": 1},'
        '{"fieldName": "Name","key": "name","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Hostname","key": "hostname","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Device Type","key": "deviceType","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Model","key": "model","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Service Tag","key": "serviceTag","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Location","key": "location","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Average Power","key": "avgPower","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Maximum Power","key": "maxPower","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Time Stamp","key": "timeStamp","mandatory": 0,"needed": 1,"required": 1},'
        '{"fieldName": "Percent Power","key": "percentPower","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Memory Power Consumption","key": "memPowerConsumption","mandatory": 0,"needed":
1,"required": 1},'
        '{"fieldName": "CPU Power Consumption","key": "cpuPowerConsumption","mandatory": 0,"needed": 1,"required":
1},'
        '{"fieldName": "CPU Utilization","key": "cpuUtilization","mandatory": 0,"needed": 1,"required": 1},'
        '{"fieldName": "Memory Bandwidth Utilization","key": "memUtilization","mandatory": 0,"needed": 1,"required":
1},'
        '{"fieldName": "I/O Utilization","key": "ioUtilization","mandatory": 0,"needed": 1,"required": 1}]}}')
```

5.12 Power Threshold Violations Report

Below API is used for consuming " Power Threshold Violations" report.

API usage:

POST <BASE_URL>/powerThresholdViolations

Sample Request:

Used	Body	Description
✓	"aggPeriodType": -1,	Aggregation period disabled. Must be -1
✓	"aggPeriodValue": -1,	Aggregation period disabled. Must be -1
✓	"attributes": [],	Use default of customize attributes
✓	"startDateTime": "2016-05-01T00:00:00.000Z",	Start date time should be earlier than end date time and current date time
✓	"endDateTime": "2016-05-02T00:00:00.000Z",	End date time should fall between start date time and current date time
✓	"maxResults": -1,	Max result record count allowed, -1 for returning all records
✗	"entityList": [],	not applicable
✗	"entityGroupList": [],	not applicable
✗	"percentPower": 0,	not applicable
✗	"severity": [],	not applicable

✗	"selectAllDevices" : false,	not applicable, do not use or set value to false
✗	"selectAllGroups" : false,	not applicable, do not use or set value to false

Sample Code:

```
    url = bseurl + 'powerThresholdViolations'
    post_data = str.encode(
        '{"requestObj": '
        '{"description": "Device/Group power threshold violations",'
        '"startDateTime": "' + reportStartDate + '",'
        '"endDateTime": "' + reportEndDate + '",'
        '"aggPeriodType": -1,'
        '"aggPeriodValue": -1,'
        '"maxResults": -1,'
        '"entityList": [],'
        '"entityGroupList": [],'
        '"percentPower": 0,'
        '"severity": [],'
        '"selectAllDevices": false,'
        '"selectAllGroups": false,'
        '"attributes": ['
        '{"fieldName": "Name","key": "name","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Hostname","key": "hostname","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Device Type","key": "deviceType","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Model","key": "model","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Device IP","key": "deviceIp","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Service Tag","key": "serviceTag","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Location","key": "location","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Number of Violations","key": "violationCount","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Average Warning Violation","key": "avgWarningViolationAmount","mandatory": 1,"needed":
1,"required": 1},'
        '{"fieldName": "Average Critical Violation","key": "avgCriticalViolationAmount","mandatory": 1,"needed":
1,"required": 1}]}}')
```

5.13 Power Cap Violations Report

Below API is used for consuming " Power Cap Violations" report.

API usage:

POST <BASE_URL>/powerCapViolations

Sample Request:

Used	Body	Description
✓	"aggPeriodType": -1,	Aggregation period disabled. Must be -1
✓	"aggPeriodValue": -1,	Aggregation period disabled. Must be -1
✓	"attributes": [],	Use default of customize attributes
✓	"startDateTime": "2016-05-01T00:00:00.000Z",	Start date time should be earlier than end date time and current date time
✓	"endDateTime": "2016-05-02T00:00:00.000Z",	End date time should fall between start date time and current date time
✓	"maxResults": -1,	Max result record count allowed, -1 for returning all records
✗	"entityList": [],	not applicable
✗	"entityGroupList": [],	not applicable
✗	"percentPower": 0,	not applicable

✗	"severity": [],	not applicable
✗	“selectAllDevices” : false,	not applicable, do not use or set value to false
✗	“selectAllGroups” : false,	not applicable, do not use or set value to false

Sample Code:

```
    url = bseurl + 'powerCapViolations'
    post_data = str.encode(
        '{"requestObj": '
        '{"description": "Device/Group power cap violations",'
        '"startDateTime": "' + reportStartDate + '",'
        '"endDateTime": "' + reportEndDate + '",'
        '"aggPeriodType": -1,'
        '"aggPeriodValue": -1,'
        '"maxResults": -1,'
        '"entityList": [],'
        '"entityGroupList": [],'
        '"percentPower": 0,'
        '"severity": [],'
        '"selectAllDevices": false,'
        '"selectAllGroups": false,'
        '"attributes": ['
        '{"fieldName": "Name","key": "name","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Hostname","key": "hostname","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Device Type","key": "deviceType","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Model","key": "model","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Service Tag","key": "serviceTag","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Location","key": "location","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Number of Violations","key": "violationCount","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Average Critical Violation","key": "avgCriticalViolationAmount","mandatory": 1,"needed":
1,"required": 1}]}}')
```

5.14 Power Cap Settings Report

Below API is used for consuming “ Power Cap Settings” report.

API usage:

POST <BASE_URL>/powerCapSettings

Sample Request:

Used	Body	Description
✓	"aggPeriodType": -1,	Aggregation period disabled. Must be -1
✓	"aggPeriodValue": -1,	Aggregation period disabled. Must be -1
✓	"attributes": [],	Use default of customize attributes
✗	"startDateTime": "2016-05-01T00:00:00.000Z",	not applicable
✗	"endDateTime": "2016-05-02T00:00:00.000Z",	not applicable
✓	"maxResults": -1,	Max result record count allowed, -1 for returning all records
✗	"entityList": [],	not applicable
✗	"entityGroupList": [],	not applicable
✗	"percentPower": 0,	not applicable
✗	"severity": [],	not applicable

✗	"selectAllDevices" : false,	not applicable, do not use or set value to false
✗	"selectAllGroups" : false,	not applicable, do not use or set value to false

Sample Code:

```
    url = bseurl + 'powerCapSettings'
    post_data = str.encode(
        '{"requestObj": '
        '{"description": "Device/Group power cap settings",'
        '"startDateTime": "",'
        '"endDateTime": "",'
        '"aggPeriodType": -1,'
        '"aggPeriodValue": -1,'
        '"maxResults": -1,'
        '"entityList": [],'
        '"entityGroupList": [],'
        '"percentPower": 0,'
        '"severity": [],'
        '"selectAllDevices": false,'
        '"selectAllGroups": false,'
        '"attributes": ['
        '{"fieldName": "Name","key": "name","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Hostname","key": "hostname","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Device Type","key": "deviceType","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Model","key": "model","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Service Tag","key": "serviceTag","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Location","key": "location","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Policy Name","key": "policyName","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Policy Type","key": "policyType","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Policy Status","key": "policyStatus","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Policy Active","key": "policyActive","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Power Cap","key": "powerCap","mandatory": 1,"needed": 1,"required": 1}]}}')
```

5.15 Threshold Settings Report

Below API is used for consuming "Threshold Settings" report.

API usage:

POST <BASE_URL>/thresholdSettings

Sample Request:

Used	Body	Description
✓	"aggPeriodType": -1,	Aggregation period disabled. Must be -1
✓	"aggPeriodValue": -1,	Aggregation period disabled. Must be -1
✓	"attributes": [],	Use default of customize attributes
✗	"startDateTime": "2016-05-01T00:00:00.000Z",	not applicable
✗	"endDateTime": "2016-05-02T00:00:00.000Z",	not applicable
✓	"maxResults": -1,	Max result record count allowed, -1 for returning all records
✗	"entityList": [],	not applicable
✗	"entityGroupList": [],	not applicable
✗	"percentPower": 0,	not applicable
✗	"severity": [],	not applicable

✗	“selectAllDevices” : false,	not applicable, do not use or set value to false
✗	“selectAllGroups” : false,	not applicable, do not use or set value to false

Sample Code:

```
    url = bseurl + 'thresholdSettings'
    post_data = str.encode(
        '{"requestObj": '
        '{"description": "Device/Group threshold settings",'
        '"startDateTime": "",'
        '"endDateTime": "",'
        '"aggPeriodType": -1,'
        '"aggPeriodValue": -1,'
        '"maxResults": -1,'
        '"entityList": [],'
        '"entityGroupList": [],'
        '"percentPower": 0,'
        '"severity": [],'
        '"selectAllDevices": false,'
        '"selectAllGroups": false,'
        '"attributes": ['
        '{"fieldName": "Name","key": "name","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Hostname","key": "hostname","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Device Type","key": "deviceType","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Model","key": "model","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Service Tag","key": "serviceTag","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Location","key": "location","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Power Threshold","key": "powerThreshold","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Temperature Threshold","key": "tempThreshold","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Event Condition","key": "eventCondition","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Event Severity","key": "severity","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Event Type","key": "eventType","mandatory": 1,"needed": 1,"required": 1}]}}')
```

5.16 Rack Fragmentation Hoarders Report

Below API is used for consuming “Rack Fragmentation Hoarders” report.

API usage:

POST <BASE_URL>/rackFragmentationHoarders

Sample Request:

Used	Body	Description
✓	"aggPeriodType": -1,	Aggregation period disabled. Must be -1
✓	"aggPeriodValue": -1,	Aggregation period disabled. Must be -1
✓	"attributes": [],	Use default of customize attributes
✗	"startDateTime": "2016-05-01T00:00:00.000Z",	not applicable
✗	"endDateTime": "2016-05-02T00:00:00.000Z",	not applicable
✓	"maxResults": -1,	Max result record count allowed, -1 for returning all records
✗	"entityList": [],	not applicable
✗	"entityGroupList": [],	not applicable
✗	"percentPower": 0,	not applicable
✗	"severity": [],	not applicable

✗	“selectAllDevices” : false,	not applicable, do not use or set value to false
✗	“selectAllGroups” : false,	not applicable, do not use or set value to false

Sample Code:

```
    url = bseurl + 'rackFragmentationHoarders'
    post_data = str.encode(
        '{"requestObj": '
        '{"description": "Racks with most fragmentation",'
        '"startDateTime": "",'
        '"endDateTime": "",'
        '"aggPeriodType": -1,'
        '"aggPeriodValue": -1,'
        '"maxResults": -1,'
        '"entityList": [],'
        '"entityGroupList": [],'
        '"percentPower": 0,'
        '"severity": [],'
        '"selectAllDevices": false,'
        '"selectAllGroups": false,'
        '"attributes": ['
        '{"fieldName": "Name","key": "name","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Rank","key": "rank","mandatory": 0,"needed": 1,"required": 1},'
        '{"fieldName": "Location","key": "location","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Fragmentation Factor","key": "fragmentationFactor","mandatory": 1,"needed": 1,"required":
1},'
        '{"fieldName": "Rack Utilization","key": "rackUtilization","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Rack Utilization Percent","key": "rackUtilizationPercent","mandatory": 1,"needed":
1,"required": 1},'
        '{"fieldName": "Max Gap","key": "maxGap","mandatory": 1,"needed": 1,"required": 1}]}}')
```

5.17 Rack Space Hoarders Report

Below API is used for consuming “Rack Space Hoarders” report.

API usage:

POST <BASE_URL>/rackSpaceHoarders

Sample Request:

Used	Body	Description
✓	"aggPeriodType": -1,	Aggregation period disabled. Must be -1
✓	"aggPeriodValue": -1,	Aggregation period disabled. Must be -1
✓	"attributes": [],	Use default of customize attributes
✗	"startDateTime": "2016-05-01T00:00:00.000Z",	not applicable
✗	"endDateTime": "2016-05-02T00:00:00.000Z",	not applicable
✓	"maxResults": -1,	Max result record count allowed, -1 for returning all records
✗	"entityList": [],	not applicable
✗	"entityGroupList": [],	not applicable
✗	"percentPower": 0,	not applicable
✗	"severity": [],	not applicable
✗	“selectAllDevices” : false,	not applicable, do not use or set value to false

✗	“selectAllGroups” : false,	not applicable, do not use or set value to false

Sample Code:

```
    url = bseurl + 'rackSpaceHoarders'
    post_data = str.encode(
        '{"requestObj": '
        '{"description": "Racks which have highest utilization of space",'
        '"startDateTime": "",'
        '"endDateTime": "",'
        '"aggPeriodType": -1,'
        '"aggPeriodValue": -1,'
        '"maxResults": -1,'
        '"entityList": [],'
        '"entityGroupList": [],'
        '"percentPower": 0,'
        '"severity": [],'
        '"selectAllDevices": false,'
        '"selectAllGroups": false,'
        '"attributes": ['
        '{"fieldName": "Name","key": "name","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Rank","key": "rank","mandatory": 0,"needed": 1,"required": 1},'
        '{"fieldName": "Location","key": "location","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Rack Utilization","key": "rackUtilization","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Rack Utilization Percent","key": "rackUtilizationPercent","mandatory": 1,"needed":
1,"required": 1}]}}')
```

5.18 Rack Space Frugal Report

Below API is used for consuming “Rack Space Frugal” report.

API usage:

POST <BASE_URL>/rackSpaceFrugal

Sample Request:

Used	Body	Description
✓	"aggPeriodType": -1,	Aggregation period disabled. Must be -1
✓	"aggPeriodValue": -1,	Aggregation period disabled. Must be -1
✓	"attributes": [],	Use default of customize attributes
✗	"startDateTime": "2016-05-01T00:00:00.000Z",	not applicable
✗	"endDateTime": "2016-05-02T00:00:00.000Z",	not applicable
✓	"maxResults": -1,	Max result record count allowed, -1 for returning all records
✗	"entityList": [],	not applicable
✗	"entityGroupList": [],	not applicable
✗	"percentPower": 0,	not applicable
✗	"severity": [],	not applicable
✗	“selectAllDevices” : false,	not applicable, do not use or set value to false
✗	“selectAllGroups” : false,	not applicable, do not use or set value to false

Sample Code:

```
    url = bseurl + 'rackSpaceFrugal'
    post_data = str.encode(
        '{"requestObj": '
        '{"description": "Racks which have lowest utilization of space",'
        '"startDateTime": "",'
        '"endDateTime": "",'
        '"aggPeriodType": -1,'
        '"aggPeriodValue": -1,'
        '"maxResults": -1,'
        '"entityList": [],'
        '"entityGroupList": [],'
        '"percentPower": 0,'
        '"severity": [],'
        '"selectAllDevices": false,'
        '"selectAllGroups": false,'
        '"attributes": ['
        '{"fieldName": "Name","key": "name","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Rank","key": "rank","mandatory": 0,"needed": 1,"required": 1},'
        '{"fieldName": "Location","key": "location","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Rack Utilization","key": "rackUtilization","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Rack Utilization Percent","key": "rackUtilizationPercent","mandatory": 1,"needed":
1,"required": 1}]}}')
```

5.19 Highest Temperature Report

Below API is used for consuming "Highest Temperature" report.

API usage:

POST <BASE_URL>/highestTemperature

Sample Request:

Used	Body	Description
✓	"aggPeriodType": -1,	Aggregation period disabled. Must be -1
✓	"aggPeriodValue": -1,	Aggregation period disabled. Must be -1
✓	"attributes": [],	Use default of customize attributes
✓	"startDateTime": "2016-05-01T00:00:00.000Z",	Start date time should be earlier than end date time and current date time.
✓	"endDateTime": "2016-05-02T00:00:00.000Z",	End date time should fall between start date time and current date time.
✓	"maxResults": -1,	Max result record count allowed, -1 for returning all records
✗	"entityList": [],	not applicable
✗	"entityGroupList": [],	not applicable
✗	"percentPower": 0,	not applicable
✗	"severity": [],	not applicable
✗	"selectAllDevices" : false,	not applicable, do not use or set value to false
✗	"selectAllGroups" : false,	not applicable, do not use or set value to false

Sample Code:

```
    url = bseurl + 'highestTemperature'
    post_data = str.encode(
        '{"requestObj": '
        '{"description": "Devices that are the hottest",'
        '"startDateTime": "' + reportStartDate + '",'
        '"endDateTime": "' + reportEndDate + '",'
        '"aggPeriodType": -1,'
        '"aggPeriodValue": -1,'
        '"maxResults": -1,'
        '"entityList": [],'
        '"entityGroupList": [],'
        '"percentPower": 0,'
        '"severity": [],'
        '"selectAllDevices": false,'
        '"selectAllGroups": false,'
        '"attributes": ['
        '{"fieldName": "Name","key": "name","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Rank","key": "rank","mandatory": 0,"needed": 1,"required": 1},'
        '{"fieldName": "Hostname","key": "hostname","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Device Type","key": "deviceType","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Model","key": "model","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Service Tag","key": "serviceTag","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Location","key": "location","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Connection State","key": "connectionState","mandatory": 0,"needed": 1,"required": 1},'
        '{"fieldName": "Health State","key": "healthState","mandatory": 0,"needed": 1,"required": 1},'
        '{"fieldName": "Time Stamp","key": "timeStamp","mandatory": 0,"needed": 1,"required": 1},'
        '{"fieldName": "Device Ip","key": "deviceIp","mandatory": 0,"needed": 1,"required": 1},'
        '{"fieldName": "Average Temperature","key": "avgTemp","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Maximum Temperature","key": "maxTemp","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Minimum Temperature","key": "minTemp","mandatory": 1,"needed": 1,"required": 1}]}}')
```

5.20 Lowest Temperature Report

Below API is used for consuming “Lowest Temperature” report.

API usage:

POST <BASE_URL>/lowestTemperature

Sample Request:

Used	Body	Description
✓	"aggPeriodType": -1,	Aggregation period disabled. Must be -1
✓	"aggPeriodValue": -1,	Aggregation period disabled. Must be -1
✓	"attributes": [],	Use default of customize attributes
✓	"startDateTime": "2016-05-01T00:00:00.000Z",	Start date time should be earlier than end date time and current date time.
✓	"endDateTime": "2016-05-02T00:00:00.000Z",	End date time should fall between start date time and current date time.
✓	"maxResults": -1,	Max result record count allowed, -1 for returning all records
✗	"entityList": [],	not applicable
✗	"entityGroupList": [],	not applicable
✗	"percentPower": 0,	not applicable
✗	"severity": [],	not applicable
✗	“selectAllDevices” : false,	not applicable, do not use or set value to false
✗	“selectAllGroups” : false,	not applicable, do not use or set value to false

Sample Code:

```
    url = bseurl + 'lowestTemperature'
    post_data = str.encode(
        '{"requestObj": '
        '{"description": "Devices that are the coolest",'
        '"startDateTime": "' + reportStartDate + '",'
        '"endDateTime": "' + reportEndDate + '",'
        '"aggPeriodType": -1,'
        '"aggPeriodValue": -1,'
        '"maxResults": -1,'
        '"entityList": [],'
        '"entityGroupList": [],'
        '"percentPower": 0,'
        '"severity": [],'
        '"selectAllDevices": false,'
        '"selectAllGroups": false,'
        '"attributes": ['
        '{"fieldName": "Name","key": "name","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Rank","key": "rank","mandatory": 0,"needed": 1,"required": 1},'
        '{"fieldName": "Hostname","key": "hostname","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Device Type","key": "deviceType","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Model","key": "model","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Service Tag","key": "serviceTag","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Location","key": "location","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Connection State","key": "connectionState","mandatory": 0,"needed": 1,"required": 1},'
        '{"fieldName": "Health State","key": "healthState","mandatory": 0,"needed": 1,"required": 1},'
        '{"fieldName": "Time Stamp","key": "timeStamp","mandatory": 0,"needed": 1,"required": 1},'
        '{"fieldName": "Device Ip","key": "deviceIp","mandatory": 0,"needed": 1,"required": 1},'
        '{"fieldName": "Average Temperature","key": "avgTemp","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Maximum Temperature","key": "maxTemp","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Minimum Temperature","key": "minTemp","mandatory": 1,"needed": 1,"required": 1}]}}')
```

5.21 Events Report

Below API is used for consuming “Events” report.

API usage:

POST <BASE_URL>/eventsReport

Sample Request:

Used	Body	Description
✓	"aggPeriodType": -1,	Aggregation period disabled. Must be -1
✓	"aggPeriodValue": -1,	Aggregation period disabled. Must be -1
✓	"attributes": [],	Use default of customize attributes
✓	"startDateTime": "2016-05-01T00:00:00.000Z",	Start date time should be earlier than end date time and current date time. Can specify minute and second.
✓	"endDateTime": "2016-05-02T00:00:00.000Z",	End date time should fall between start date time and current date time. Can specify minute and second.
✓	"maxResults": -1,	Max result record count allowed, -1 for returning all records
✗	"entityList": [],	not applicable
✗	"entityGroupList": [],	not applicable
✗	"percentPower": 0,	not applicable
✓	"severity": [1,2,3],	Could be 1, 2, 3 or combination of the 3 values.

✘	“selectAllDevices” : false,	not applicable, do not use or set value to false
✘	“selectAllGroups” : false,	not applicable, do not use or set value to false

Sample Code:

```
    url = bseurl + 'eventsReport'
    post_data = str.encode(
        '{"requestObj": '
        '{"description": "Events in last n days with certain severity",'
        '"startDateTime": "' + reportStartDate + '",'
        '"endDateTime": "' + reportEndDate + '",'
        '"aggPeriodType": -1,'
        '"aggPeriodValue": -1,'
        '"maxResults": -1,'
        '"entityList": [],'
        '"entityGroupList": [],'
        '"percentPower": 0,'
        '"severity": [1,2,3],'
        '"selectAllDevices": false,'
        '"selectAllGroups": false,'
        '"attributes": ['
        '{"fieldName": "Name","key": "name","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Time Stamp","key": "timeStamp","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Event Severity","key": "severity","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Event Type","key": "eventType","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Event Acknowledged By","key": "eventAcknowledgedBy","mandatory": 0,"needed": 1,"required": 1},'
        '{"fieldName": "Event Description","key": "eventDescription","mandatory": 1,"needed": 1,"required": 1}]}}')
```

5.22 PDU Outlet Assignment Report

Below API is used for consuming “PDU Outlet Assignment” report.

API usage:

POST <BASE_URL>/pduOutletAssignment

Sample Request:

Used	Body	Description
✓	"aggPeriodType": -1,	Aggregation period disabled. Must be -1
✓	"aggPeriodValue": -1,	Aggregation period disabled. Must be -1
✓	"attributes": [],	Use default of customize attributes
✘	"startDateTime": "2016-05-01T00:00:00.000Z",	not applicable
✘	"endDateTime": "2016-05-02T00:00:00.000Z",	not applicable
✓	"maxResults": -1,	Max result record count allowed, -1 for returning all records
✓	"entityList": [{"hostName": "","ipAddress": "192.186.2.20", "serviceTagOrSerialNum": "" }],	Specify target PDU list. If "selectAllDevices" is true, this field is ignored, keep it empty. If "selectAllDevices" is false, must provide non-empty entity list. Providing empty list or invalid group / entity will result in error.

		Note: if you provided devices that are not PDUs or do not have Outlet Power capability, error will not be returned, those devices will be filtered and no relevant results will be generated.
✓	"entityGroupList": [{"groupPath": "/DC_001/Room_001/Row(100)/Rack(100)"}],	Specify target entity group list. If "selectAllGroups" is true, this field is ignored, keep it empty.If "selectAllGroups" is false, must provide non-empty entity group list. Providing empty list or invalid group / entity will result in error.
✘	"percentPower": 0,	not applicable
✘	"severity": [],	not applicable
✓	"selectAllDevices" : false,	"true" to select all PDUs with Outlet Power capability
✓	"selectAllGroups" : false,	"true" to select all groups (for all PDUs with Outlet Power capability associated with groups)

Sample Code:

```
    url = bseurl + 'pduOutletAssignment'
    post_data = str.encode(
        '{"requestObj": '
        '{"description": "PDU Outlet Assignment report",'
        '"startDateTime": "",'
        '"endDateTime": "",'
        '"aggPeriodType": -1,'
        '"aggPeriodValue": -1,'
        '"maxResults": -1,'
        '"entityList": [],'
        '"entityGroupList": [{"groupPath": "/DC_001"}],'
        '"percentPower": 0,'
        '"severity": [],'
        '"selectAllDevices": false,'
        '"selectAllGroups": false,'
        '"attributes": ['
        '{"fieldName": "PDU Name","key": "pduName","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "PDU Location","key": "pduLocation","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "PDU Model","key": "pduModel","mandatory": 0,"needed": 1,"required": 1},'
        '{"fieldName": "PDU IP Address","key": "pduIp","mandatory": 0,"needed": 1,"required": 1},'
        '{"fieldName": "PDU Serial Number or Service Tag","key": "pduServiceTag","mandatory": 0,"needed":
1,"required": 1},'
        '{"fieldName": "Total Outlet Count","key": "pduOutletCount","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Max Rating","key": "pduMaxRating","mandatory": 0,"needed": 1,"required": 1},'
        '{"fieldName": "Outlet Index","key": "pduOutletIndex","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Name","key": "name","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Hostname","key": "hostname","mandatory": 0,"needed": 1,"required": 1},'
        '{"fieldName": "Device Type","key": "deviceType","mandatory": 0,"needed": 1,"required": 1},'
        '{"fieldName": "Model","key": "model","mandatory": 0,"needed": 1,"required": 1},'
        '{"fieldName": "Serial Number or Service Tag","key": "serviceTag","mandatory": 0,"needed": 1,"required":
1},'
        '{"fieldName": "Location","key": "location","mandatory": 0,"needed": 1,"required": 1},'
        '{"fieldName": "Device Ip","key": "deviceIp","mandatory": 0,"needed": 1,"required": 1}]}}')
```

5.23 PDU Sensor Report

Below API is used for consuming “PDU Sensor” report.

API usage:

POST <BASE_URL>/pduSensor

Sample Request:

Used	Body	Description
✓	"aggPeriodType": -1,	Aggregation period disabled. Must be -1
✓	"aggPeriodValue": -1,	Aggregation period disabled. Must be -1
✓	"attributes": [],	Use default of customize attributes
✗	"startDateTime": "2016-05-01T00:00:00.000Z",	not applicable
✗	"endDateTime": "2016-05-02T00:00:00.000Z",	not applicable
✓	"maxResults": -1,	Max result record count allowed, -1 for returning all records
✓	"entityList": [{"hostName": "","ipAddress": "192.186.2.20", "serviceTagOrSerialNum": "" }],	Specify target PDU list. If "selectAllDevices" is true, this field is ignored, keep it empty. If "selectAllDevices" is false, must provide non-empty entity list. Providing empty list or invalid group / entity will result in error. Note: if you provid devices that are not PDUs or do not have sensors attached, error will not be returned, those devices will be filtered and no relevant results will be generated.
✓	"entityGroupList": [{"groupPath": "/DC_001/Room_001/Row(100)/Rack(100)"}],	Specify target entity group list. If "selectAllGroups" is true, this field is ignored, keep it empty.If "selectAllGroups" is false, must provide non-empty entity group list. Providing empty list or invalid group / entity will result in error.
✗	"percentPower": 0,	not applicable
✗	"severity": [],	not applicable
✓	“selectAllDevices” : false,	“true” to select all PDUs with sensor
✓	“selectAllGroups” : false,	“true” to select all groups (for all PDUs environment sensor attached under them)

Sample Code:

```
    url = bseurl + 'pduSensor'
    post_data = str.encode(
        '{"requestObj": '
        '{"description": "PDU environment sensors information",'
        '"startDateTime": "",'
        '"endDateTime": "",'
        '"aggPeriodType": -1,'
        '"aggPeriodValue": -1,'
        '"maxResults": -1,'
        '"entityList": [],'
        '"entityGroupList": [{"groupPath": "/BDC_001"}],'
        '"percentPower": 0,'
        '"severity": [],'
        '"selectAllDevices": false,'
        '"selectAllGroups": false,'
        '"attributes": ['
        '{"fieldName": "PDU Name","key": "pduName","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "PDU Location","key": "pduLocation","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "PDU Model","key": "pduModel","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "PDU IP Address","key": "pduIp","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Sensor Name","key": "sensorName","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Sensor Type","key": "sensorType","mandatory": 1,"needed": 1,"required": 1},'
        '{"fieldName": "Sensor Serial Number","key": "sensorSerialNumber","mandatory": 0,"needed": 1,"required": 1},'
        '{"fieldName": "Reading Unit","key": "sensorReadingUnit","mandatory": 0,"needed": 1,"required": 1},'
        '{"fieldName": "Present Reading","key": "sensorPresentReading","mandatory": 0,"needed": 1,"required": 1},'
        '{"fieldName": "Time Recorded","key": "timeRecorded","mandatory": 0,"needed": 1,"required": 1}]}}')
```

5.24 Thermal Event Policy Report

Below API is used for consuming “Thermal Event Policy” report.

API usage:

POST <BASE_URL>/thermalEventPolicy

Sample Request:

Used	Body	Description
✓	"aggPeriodType": -1,	Aggregation period disabled. Must be -1
✓	"aggPeriodValue": -1,	Aggregation period disabled. Must be -1
✓	"attributes": [],	Use default of customize attributes
✗	"startDateTime": "",	not applicable
✗	"endDateTime": "",	not applicable
✓	"maxResults": -1,	Max result record count allowed, -1 for returning all records
✗	"entityList": [],	not applicable
✗	"entityGroupList": [],	not applicable
✗	"percentPower": 0,	not applicable
✗	"severity": [],	not applicable
✗	“selectAllDevices” : false,	not applicable
✗	“selectAllGroups” : false,	not applicable

Sample Code:

```
url = bseurl + 'thermalEventPolicy'
post_data = str.encode(
    '{"requestObj": '
    '{"description": "Current thermal event based policy settings",'
    '"startDateTime": "",'
    '"endDateTime": "",'
    '"aggPeriodType": -1,'
    '"aggPeriodValue": -1,'
    '"maxResults": -1,'
    '"entityList": [],'
    '"entityGroupList": [],'
    '"percentPower": 0,'
    '"severity": [],'
    '"selectAllDevices": false,'
    '"selectAllGroups": false,'
    '"attributes": ['
    '{"fieldName": "Name","key": "name","mandatory": 1,"needed": 1,"required": 1},'
    '{"fieldName": "Location","key": "location","mandatory": 1,"needed": 1,"required": 1},'
    '{"fieldName": "Temperature Threshold","key": "tempThreshold","mandatory": 1,"needed": 1,"required": 1},'
    '{"fieldName": "Policy Name","key": "policyName","mandatory": 1,"needed": 1,"required": 1},'
    '{"fieldName": "Policy Type","key": "policyType","mandatory": 1,"needed": 1,"required": 1},'
    '{"fieldName": "Policy Status","key": "policyStatus","mandatory": 1,"needed": 1,"required": 1},'
    '{"fieldName": "Policy Active","key": "policyActive","mandatory": 1,"needed": 1,"required": 1}]}}')
```

A Additional resources

You can visit Support.dell.com which is focused on meeting your needs with proven services and support.

You can also visit DellTechCenter.com/powercenter, an IT Community where you can connect with Dell Customers and Dell employees for sharing knowledge, best practices, and information about Dell products and installations.

For more information, visit www.dell.com/powercenter