AMI DCM RESTful API

REVISION 1.1 – DECMBER 27, 2023
NDA REQUIRED

AMI DCM RESTful API Document

12/27/2023

Revision History

10-17-2023	1.0	Initial Draft of AMI DCM RESTful API Document
12-27-2023	1.1	Formatted document as per template

Disclaimer

Although efforts have been made to assure the accuracy of the information contained here, AMI expressly disclaims liability for any error in this information, and for damages, whether direct, indirect, special, exemplary, consequential or otherwise, that may result from such error, including but not limited to the loss of profits resulting from the use or misuse of the document or information contained therein (even if AMI has been advised of the possibility of such damages). Any questions or comments regarding this document or its contents should be addressed to AMI at marketing@ami.com.

AMI provides this publication "as is" without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability or fitness for a specific purpose.

Some states do not allow disclaimer of express or implied warranties or the limitation or exclusion of liability for indirect, special, exemplary, incidental or consequential damages in certain transactions; therefore, this statement may not apply to you. Also, you may have other rights that vary from jurisdiction to jurisdiction.

This publication could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. AMI may make improvements and/or revisions in the product(s) and/or the program(s) described in this publication at any time.

Requests for technical information about AMI products should be made to your AMI authorized reseller or marketing representative.

Table of Contents

Document Information

Technical Support

AMI provides technical support only for AMI products licensed directly from AMI.

Web Site

We invite you to visit our website at ami.com.

Purpose

This document is intended to be used by Engineers.

Audience

The intended audiences are Engineers.

Chapter 1

1 Introduction

This document is a reference for DCM RESTful APIs which covers the following functionalities:

1.1 Datacenter management

- Add a datacenter
- Update a datacenter
- Delete a datacenter
- Get a datacenter list
- Get basic information of a datacenter
- Get power data of a datacenter
- Get thermal data of a datacenter
- Get health data of a datacenter
- Get inventory data of a datacenter
- Get a room list of a datacenter
- Get server firmware outlier of a data center
- Get maximum inlet temperature of a data center
- Get device health statistics of a data center
- Get cooling anomaly status of a data center
- Get power data summary of a data center
- Get hotspots of a data center
- Get power capacity of a data center
- Get carbon emission of a data center
- Get carbon emission projection of a data center
- Get annual renewable energy ratio

1.2 Room management

- Add a room
- Update a room
- Delete a room
- Get a room list
- Get basic information of a room
- Get power data of a room
- Get thermal data of a room
- Get health data of a room
- Get inventory data of a room
- Get a row list of a room
- Get carbon emission of a room

1.3 Row management

- Add a row
- Update a row
- Delete a row
- Get a row list

- Get basic information of a row
- Get power data of a row
- Get thermal data of a row
- Get health data of a row
- Get inventory data of a row
- Get a rack list of a row
- Get carbon emission of a row

1.4 Rack management

- Add a rack
- Update a rack
- Delete a rack
- Get a rack list
- Get basic information of a rack
- Get power data of a rack
- Get thermal data of a rack
- Get health data of a rack
- Get inventory data of a rack
- Get a device list of a rack
- Get carbon emission of a rack

1.5 Device management

- Add a device
- Update a device
- Delete a device
- Get a device list
- Get basic information of a device
- Get power data of a device
- Get thermal data of a device
- Get health data of a device
- Get inventory data of a device
- Get a blade list of a chassis
- Get real time SNMP walk data
- Get SSD smart data
- Get real time sensor data of a device
- Get real time UPS data of a device
- Get real time PDU data of a device
- Get real time health data of a device
- Get real time component faults of a device
- Control a device

1.6 Group management

- Add a group
- Update a group
- Delete a group
- Get a group list
- Get basic information of a group
- Get power data of a group
- Get thermal data of a group
- Get health data of a group

- Get inventory data of a group
- Get a device list of a group
- Add a device into a group
- Remove a device from a group
- Get carbon emission of a group
- Get carbon emission projection of a group
- Get annual renewable energy ratio

1.7 Power policy

- Add a policy
- Get policies
- Get a policy
- Update a policy
- Delete a policy

1.8 Configuration management

- Get miscellaneous configurations
- update miscellaneous configurations
- Get predefined event configurations
- Update predefined event configurations
- Get emergency power reduction rooms
- Add an emergency power reduction room
- Delete an emergency power reduction room
- Export CPU telemetry data
- Add a user
- Get users
- Get a user
- Update a user
- Delete a user
- Add a custom attribute
- Update a custom attribute
- Delete a custom attribute
- Get a custom attribute
- Get custom attributes

1.9 Discovery management

- Add a discovery task
- Get discovery tasks
- Get a discovery task
- Control a discovery task
- Delete a discovery task

1.10 Batch hierarchy operation

- Add a batch add task
- Get batch add tasks
- Get a batch add task
- Control a batch add task
- Delete a batch add task
- Update/delete entities

1.11 Analysis

- Get unhealthy SSDs
- Get unhealthy devices

1.12 Event management

- Add a notification
- Update a notification
- Get notifications
- Get a notification
- Delete a notification
- Add an event handler
- Get event handlers
- Get an event handler
- Delete an event handler
- Get historical events

1.13 Data service

- Get measurement data of an entity
- Get latest measurement data of entities
- Get power capacity of an entity
- Get space capacity of an entity
- Get weight capacity of an entity
- Get power status history of entities
- Get low utilization servers
- Analyze low utilization servers

1.14 Provisioning management

- Add a provisioning task
- Get provisioning tasks
- Get a provisioning task
- Control a provisioning task
- Delete a provisioning task

Chapter 2

2 Authentication

- Basic authentication is supported. Link username and password with a colon, with base64 encoding and put the result string in the standard "Authorization" http header. This authentication is only used for DCM Console accounts. AD and LDAP accounts cannot be supported.
- Customized HTTP header
  As it is inconvenient to support AD and LDAP accounts, we add customized HTTP headers to transfer authentication information:

Field	Attribute	Description
dcmUserName	Required	User name.
dcmUserPassword	Required	Password.
dcmAccountType	Required	Account type: 0- Console user 1- AD user 2- LDAP user
dcmDomain	Optional	It means the AD domain for AD user.

- If the "Authorization" http header is present, the RESTful API server will use basic authentication. Otherwise, customized http header will be checked for authentication.
- Tenant users will not be supported for these RESTful APIs.
- For each RESTful API, the authentication header should be presented.
- HTTPS should be enabled to ensure transport security.

Chapter 3

3 Version

The APIs will be maintained with no broken changes. So, the clients do not need to present any version information at the moment.

Chapter 4

4 URI

/rest/datacenters
/rest/datacenters/{dcId}
/rest/datacenters/{dcId}/power
/rest/datacenters/{dcId}/thermal
/rest/datacenters/{dcId}/health
/rest/datacenters/{dcId}/inventory
/rest/datacenters/{dcId}/serverFirmwareOutlier
/rest/datacenters/{dcId}/maxInletTemperature
/rest/datacenters/{dcId}/deviceHealthStatistics
/rest/datacenters/{dcId}/coolingAnomalyStatus
/rest/datacenters/{dcId}/powerDataSummary
/rest/datacenters/{dcId}/hotspots
/rest/datacenters/{dcId}/powerCapacity
/rest/datacenters/{dcId}/topInletTemperatureRooms
/rest/datacenters/{dcId}/carbonEmission
/rest/datacenters/{dcId}/carbonEmissionProjection
/rest/datacenters/{dcId}/annualRenewableEnergyRatio

/rest/rooms
/rest/rooms/{roomId}
/rest/rooms/{roomId}/power
/rest/rooms/{roomId}/thermal
/rest/rooms/{roomId}/health
/rest/rooms/{roomId}/inventory
/rest/rooms/{roomId}/carbonEmission

/rest/rows
/rest/rows/{rowId}
/rest/rows/{rowId}/power
/rest/rows/{rowId}/thermal
/rest/rows/{rowId}/health
/rest/rows/{rowId}/inventory
/rest/rows/{rowId}/carbonEmission

/rest/racks
/rest/racks/{rackId}
/rest/racks/{rackId}/power
/rest/racks/{rackId}/thermal
/rest/racks/{rackId}/health
/rest/racks/{rackId}/inventory
/rest/racks/{rackId}/carbonEmission

/rest/devices
/rest/devices/{deviceId}
/rest/devices/{deviceId}/power
/rest/devices/{deviceId}/thermal
/rest/devices/{deviceId}/health
/rest/devices/{deviceId}/inventory
/rest/devices/{deviceId}/blades
/rest/devices/{deviceId}/SSDSmartData
/rest/devices/{deviceId}/realTimeSNMPWalkData
/rest/devices/{deviceId}/realTimeSensorData
/rest/devices/{deviceId}/realTimeUPSData
/rest/devices/{deviceId}/realTimePDUData

```
/rest/devices/{deviceId}/realTimeHealthData
/rest/devices/{deviceId}/realTimeComponentFaults
/rest/devices/{deviceId}/control
/rest/devices/{deviceId}/outlets
/rest/devices/{deviceId}/outlets/{outletId}
/rest/devices/{deviceId}/carbonEmission

/rest/groups
/rest/groups/{groupId}
/rest/groups/{groupId}/power
/rest/groups/{groupId}/thermal
/rest/groups/{groupId}/health
/rest/groups/{groupId}/inventory
/rest/groups/{groupId}/devices
/rest/groups/{groupId}/carbonEmission
/rest/groups/{groupId}/carbonEmissionProjection
/rest/groups/{groupId}/annualRenewableEnergyRatio

/rest/policies
/rest/policies/{policyId}

/rest/configuration/misc
/rest/configuration/predefinedEvents
/rest/configuration/EPRRooms
/rest/configuration/cpuTelemetry
/rest/configuration/SiAASConfig
/rest/configuration/SiAASTelemetry
/rest/configuration/users
/rest/configuration/users/{userId}
/rest/configuration/customAttributes

/rest/discoveryTasks
/rest/discoveryTasks/{taskId}
/rest/discoveryTasks/{taskId}/control

/rest/batchAddTasks
/rest/batchAddTasks/{taskId}
/rest/batchAddTasks/{taskId}/control

/rest/entities

/rest/event/notifications
/rest/event/notifications/{notificationId}
/rest/event/eventHandlers
/rest/event/eventHandlers/{eventHandlerId}
/rest/event/historicalEvents

/rest/analysis/unhealthySSDs
/rest/analysis/unhealthyDevices

/rest/data/measurementData
/rest/data/latestMeasurementData
/rest/data/powerStatusHistory

/rest/provisioningTasks
/rest/provisioningTasks/{taskId}
/rest/provisioningTasks/{taskId}/control

/rest/data/lowUtilizationServers
/rest/data/lowUtilizationServers/analyze
```

The flat URI structure in DCM RESTful APIs is used to reduce the URI complexity.

Chapter 5

5 HATEOAS

HATEOAS (Hypertext As The Engine Of Application State) will be supported. Links will be in the response to help navigate between APIs. For example, after sending a GET request to URL "/rest/datacenters", you will receive the following response:


```
{
    "links": [
        {
            "rel": "self",
            "href": "https://localhost:8643/DcmConsole/rest/datacenters"
        }
    ],
    "content": [
        {
            "id": 46,
            "name": "dc-1",
            "powerCapacity": 0,
            "pue": 0,
            "links": [
                {
                    "rel": "self",
                    "href":
"https://localhost:8643/DcmConsole/rest/datacenters/46"
                },
                {
                    "rel": "power",
                    "href":
"https://localhost:8643/DcmConsole/rest/datacenters/46/power{?latest,startDateTi
me,endDateTime,page,itemsPerPage,granularity}"
                },
                {
                    "rel": "thermal",
                    "href":
"https://localhost:8643/DcmConsole/rest/datacenters/46/thermal{?latest,startDate
Time,endDateTime,page,itemsPerPage,granularity}"
                },
                {
                    "rel": "health",
                    "href":
"https://localhost:8643/DcmConsole/rest/datacenters/46/health"
                },
                {
                    "rel": "rooms",
                    "href":
"https://localhost:8643/DcmConsole/rest/rooms?parentId=46"
                }
            ]
        }
    ]
}
```


The field "rel" represents the relationship between the API and the current API. Below is the list all "rel" in DCM:

Rel	Description
Self	The API itself.
Power	The power API.
Thermal	The temperature API.
Health	The health data API.
Rooms	The API to get rooms under the datacenter.
Rows	The API to get rows under the room.
Racks	The API to get racks under the row.
Devices	The API to get devices under the rack or group.
blades	The API to get blades under the chassis device.
add	The API to add a new resource.
update	The API to update the resource.
delete	The API to delete the resource.
serverFirmwareOutlier	The API to get server firmware outlier.
maxInletTemperature	The API to get maximum inlet temperature.
deviceHealthStatistics	The API to get device health statistics information.
coolingAnomaly	The API to get cooling anomaly status.
powerDataSummary	The API to get power data summary.
control	The API to apply a special control action to the resource.
hotspots	The API to get hotspots.
powerCapacity	The API to get power capacity.
realTimeSNMPWalkData	The API to get SNMP walk data.
realTimeSensorData	The API to get real time sensor data.
realTimeUPSData	The API to get real time UPS data.
realTimePDUData	The API to get real time PDU data.
realTimeHealthData	The API to get real time health data.
realTimeComponentFaults	The API to get real time component faults.
SSDSmartData	The API to get SSD smart data.
measurementData	The API to get measurement data.
topInletTemperatureRooms	The API to get top N inlet temperature rooms
inventory	The API to get inventory information.

Chapter 6

6 Error handling

- Standard HTTP status code will be used which means:

Status Code	Usage
200(OK)	The request succeeded.
201(Created)	When a new resource is created or updated, 201 will be returned. The link to the new resource will be in the Location header.
202(Accepted)	If an accepted request will not be finished before return, 202 will be used.
204(No Content)	No content success status response code indicates that the request has succeeded.
400(Bad Request)	DCM cannot or will not process the request due to a client error.
401(Unauthorized)	If the clients do not present the required credential information in the header or the authentication failed, it will be returned.
403(Forbidden)	When the credential is correct, but the permission mismatches, it will be returned.
404(Not Found)	When the request resource does not exist, it will be returned.
405(Method not allowed)	When the HTTP method is not supported, this code will be returned. For example, try to POST a request to create power of a data center, the supported methods will be returned in the Allow header.
415(Unsupported Media Type)	Only "application/json" is accepted in the request header. If POST or PUT data in http body is in wrong media type, this code will be returned.
500(Internal Server Error)	Normally this kind of error should be handled in the server application. When unexpected exception is raised, this error code will be returned.
501(Not Implemented)	It is used for those APIs which have declaration without implementation.

When a HTTP code with 400, 401 and 403 is returned, the error information will be displayed in two fields in the response body:

```
{
"errorCode": String,
"errorMessage": String
}
```

- Error code is a string which will be described in each API.
- Error message will be an English string to describe the detailed error information.
- Except the errors defined for each API, the following generic errors should also be caught:

Error Code	Description
SYNTAX_ERROR	Syntax error in request data.
INVALID_LICENSE	The license is invalid.
LICENSE_EXPIERED	The license is expired.
AUTH_FAILURE	Authentication failure.
INVALID_AUTHORIZATION_HEADER	Invalid authorization header.
USER_NAME_IS_REQUIRED	User name is required.
ACCOUNT_TYPE_IS_REQUIRED	Password is required.
ACCOUNT_TYPE_IS_INVALID	Account type is required.

DOMAIN_IS_REQUIRED	Domain is required.
TENANT_USER_NOT_SUPPORTED	Tenant user is not supported.
NO_PRIVILEDGE	No privilege.
INTERNAL_ERROR	Internal error.

Chapter 7

7 Client stub generation

DCM Console provides an openAPI definition JSON file located in the root folder of DCM Consolewhich allows access via the URL https://localhost:8643/DcmConsole/dcmAPI.json.

Some tools like Swagger Codegen support to generate a client stub based on the openAPI JSON file. For an example, the following command can generate a client stub in CSHARP language:

```
java -jar swagger-codegen-cli-2.2.1.jar generate -i
https://localhost:8643/DcmConsole/dcmAPI.json
-Dio.swagger.parser.util.RemoteUrl.trustAll=true -l csharp
```

Below is an example to call the stub in CSHARP:

```
static void Main(string[] args) {
    DccontrollerApi instance = new DccontrollerApi(
        "https://localhost:8643/rest");
    String userName = "dcmadmin";
    String password = "mypassowrd";
    String encodingCredential = Convert.ToBase64String(ASCIIEncoding.ASCII
        .GetBytes(userName + ":" + password));
    instance.Configuration.DefaultHeader
        .Add("Authorization", "Basic " + encodingCredential);

    try {
        // Query all data centers
        ResourcesResourceDcInfo ret = instance.GetAllDcUsingGET();
        foreach(ResourceDcInfo dc in ret.Content) {
            Console.WriteLine(dc.Id);
            Console.WriteLine(dc.Name);
            Console.WriteLine(dc.PowerCapacity);
            Console.WriteLine(dc.Pue);
            foreach(Link link in dc.Links) {
                Console.WriteLine(link.Href);
            }
        }
    } catch (Exception e) {
        Console.WriteLine(e.Message);
    }
}
```

For the tool Swagger Codegen, please refer to https://swagger.io/docs/open-source-tools/swagger-codegen/.

Chapter 8

8 APIs

8.1 General Guide

- The supported content type is "application/json".
- The requested input information will be submitted as request parameters in the URL for all HTTP GET requests.
- Except HTTP GET request and some special HTTP headers, the input information will be transferred in the HTTP body and response will also be included in the body.
- The Host header must be set to the hostname or IP address of the DCM server. Otherwise, INVALID_PARAMETER will be returned.
- In order to handle different time zones, all date time in request should follow the format ISO_OFFSET_DATE_TIME (i.e. "yyyy-MM-dd'T'HH:mm:ssXXX"). For example, 2019-08-02T11:00:00+08:00. That the symbol "+" needs to be encoded as "%2B" when you send http requests.

8.2 Add a data center

8.2.1 Request URL

/rest/datacenters

8.2.2 HTTP request method

POST

8.2.3 Request example

```
{
  "name": "dc1",
  "description": "description",
  "powerCapacity": 10000,
  "pue": 1.8,
  "electricityRate": 1.1
}
```

8.2.4 Request field description

Field	Type	Attribute	Description
name	String	Required	The name of the data center.
description	String	Optional	The description for the data center.
powerCapacity	Integer	Optional	The total power capacity of the data center. The unit is W.
pue	Double	Optional	The Power Usage Effectiveness of the data center.
electricityRate	Double	Optional	The electricity rate of the data center. If the value is set, it will overwrite the global electricity rate setting for the data center.
cefConfig	Struct	Optional	Carbon emission factor configuration struct which has two fields. One is cef, the other unit.
cefConfig.cef	Array	Optional	Carbon emission factor two-dimensional array. The first dimension is 12, which represents the month. The second dimension is 24, which means 24 hours.

cefConfig.uint	Integer	Optional	Carbon emission factor unit. The value 0 means the unit is kilogram, and the value 1 means the unit is pound.

8.2.5 Response example


```
{
  "id": 84,
  "name": "dc1",
  "description": "description",
  "powerCapacity": 10000,
  "pue": 1.8,
  "electricityRate": 1.1,
  "links": [
    {
      "rel": "self",
      "href": "https://localhost:8643/DcmConsole/rest/datacenters/84"
    },
    {
      "rel": "power",
      "href":
"https://localhost:8643/DcmConsole/rest/datacenters/84/power{?latest,startDateTi
me,endDateTime,page,itemsPerPage,granularity}"
    },
    {
      "rel": "thermal",
      "href":
"https://localhost:8643/DcmConsole/rest/datacenters/84/thermal{?latest,startDate
Time,endDateTime,page,itemsPerPage,granularity}"
    },
    {
      "rel": "health",
      "href": "https://localhost:8643/DcmConsole/rest/datacenters/84/health"
    },
    {
      "rel": "rooms",
      "href": "https://localhost:8643/DcmConsole/rest/rooms?parentId=84"
    },
    {
      "rel": "update",
      "href": "https://localhost:8643/DcmConsole/rest/datacenters/84"
    },
    {
      "rel": "delete",
      "href": "https://localhost:8643/DcmConsole/rest/datacenters/84"
    }
  ]
}
```


8.2.6 Response field description

Field	Type	Description
id	Integer	The id of the data center.

Refer to the chapter "Add a data center" for other field definitions.

8.2.7 Errors

Except the generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
NAME_IS_REQUIRED	The name field is required.

INVALID_NAME	The name is invalid.
INVALID_POWER_CAPACITY	The power capacity field value is invalid.
INVALID_PUE	The PUE field value is invalid.
INVALID_ELECTRICITY_RATE	Invalid electricity rate.

8.3 Get data centers

8.3.1 Request URL

/rest/datacenters

8.3.2 HTTP request method

GET

8.3.3 Request example

https://localhost:8643/DcmConsole/rest/datacenters

8.3.4 Request field description

N/A.

8.3.5 Response example


```
{
  "links": [
    {
      "rel": "self",
      "href": "https://localhost:8643/DcmConsole/rest/datacenters"
    },
    {
      "rel": "add",
      "href": "https://localhost:8643/DcmConsole/rest/datacenters"
    }
  ],
  "content": [
    {
      "id": 84,
      "name": "dc1",
      "description": "description",
      "powerCapacity": 10000,
      "pue": 1.8,
      "links": [
        {
          "rel": "self",
          "href": "https://localhost:8643/DcmConsole/rest/datacenters/84"
        },
        {
          "rel": "power",
          "href":
"https://localhost:8643/DcmConsole/rest/datacenters/84/power{?latest,startDateTi
me,endDateTime,page,itemsPerPage,granularity}"
        },
        {
          "rel": "thermal",
          "href":
"https://localhost:8643/DcmConsole/rest/datacenters/84/thermal{?latest,startDate
Time,endDateTime,page,itemsPerPage,granularity}"
        },
        {
          "rel": "health",
```

```
            "href": "https://localhost:8643/DcmConsole/rest/datacenters/84/health"
          },
          {
            "rel": "rooms",
            "href": "https://localhost:8643/DcmConsole/rest/rooms?parentId=84"
          },
          {
            "rel": "update",
            "href": "https://localhost:8643/DcmConsole/rest/datacenters/84"
          },
          {
            "rel": "delete",
            "href": "https://localhost:8643/DcmConsole/rest/datacenters/84"
          },
          {
            "rel": "serverFirmwareOutlier",
            "href":
"https://localhost:8643/DcmConsole/rest/datacenters/84/serverFirmwareOutlier"
          },
          {
            "rel": "maxInletTemperature",
            "href":
"https://localhost:8643/DcmConsole/rest/datacenters/84/maxInletTemperature"
          },
          {
            "rel": "deviceHealthStatistics",
            "href":
"https://localhost:8643/DcmConsole/rest/datacenters/84/deviceHealthStatistics"
          },
          {
            "rel": "coolingAnomaly",
            "href":
"https://localhost:8643/DcmConsole/rest/datacenters/84/coolingAnomaly"
          },
          {
            "rel": "powerDataSummary",
            "href":
"https://localhost:8643/DcmConsole/rest/datacenters/84/powerDataSummary"
          }
        ]
      },
      {
        "id": 46,
        "name": "dc2",
        "powerCapacity": 0,
        "pue": 2,
        "links": [
          {
            "rel": "self",
            "href": "https://localhost:8643/DcmConsole/rest/datacenters/46"
          },
          {
            "rel": "power",
            "href":
"https://localhost:8643/DcmConsole/rest/datacenters/46/power{?latest,startDateTi
me,endDateTime,page,itemsPerPage,granularity}"
          },
          {
            "rel": "thermal",
            "href":
"https://localhost:8643/DcmConsole/rest/datacenters/46/thermal{?latest,startDate
Time,endDateTime,page,itemsPerPage,granularity}"
          },
```

```
        {
          "rel": "health",
          "href": "https://localhost:8643/DcmConsole/rest/datacenters/46/health"
        },
        {
          "rel": "rooms",
          "href": "https://localhost:8643/DcmConsole/rest/rooms?parentId=46"
        },
        {
          "rel": "update",
          "href": "https://localhost:8643/DcmConsole/rest/datacenters/46"
        },
        {
          "rel": "delete",
          "href": "https://localhost:8643/DcmConsole/rest/datacenters/46"
        },
        {
          "rel": "serverFirmwareOutlier",
          "href":
"https://localhost:8643/DcmConsole/rest/datacenters/46/serverFirmwareOutlier"
        },
        {
          "rel": "maxInletTemperature",
          "href":
"https://localhost:8643/DcmConsole/rest/datacenters/46/maxInletTemperature"
        },
        {
          "rel": "deviceHealthStatistics",
          "href":
"https://localhost:8643/DcmConsole/rest/datacenters/46/deviceHealthStatistics"
        },
        {
          "rel": "coolingAnomaly",
          "href":
"https://localhost:8643/DcmConsole/rest/datacenters/46/coolingAnomaly"
        },
        {
          "rel": "powerDataSummary",
          "href":
"https://localhost:8643/DcmConsole/rest/datacenters/46/powerDataSummary"
        }
      ]
    }
  ]
}
```


8.3.6 Response field description

Field	Type	Description
links	Array	The links to the API.
content	Array	All data centers.

Refer to the chapter "Add a data center" for other field definitions.

8.3.7 Errors

Generic errors defined in chapter "Error handling".

8.4 Get a data center

8.4.1 Request URL

/rest/datacenters/{dcId}

The parameter "dcId" is the ID of the datacenter.

8.4.2 HTTP request method

GET

8.4.3 Request example

https://localhost:8643/DcmConsole/rest/datacenters/1

8.4.4 Request field description

N/A.

8.4.5 Response example


```
{
  "id": 1,
  "name": "dc-1",
  "powerCapacity": 0,
  "pue": 2,
  "links": [
    {
      "rel": "self",
      "href": "https://localhost:8643/DcmConsole/rest/datacenters/46"
    },
    {
      "rel": "power",
      "href":
"https://localhost:8643/DcmConsole/rest/datacenters/46/power{?latest,startDateTi
me,endDateTime,page,itemsPerPage,granularity}"
    },
    {
      "rel": "thermal",
      "href":
"https://localhost:8643/DcmConsole/rest/datacenters/46/thermal{?latest,startDate
Time,endDateTime,page,itemsPerPage,granularity}"
    },
    {
      "rel": "health",
      "href": "https://localhost:8643/DcmConsole/rest/datacenters/46/health"
    },
    {
      "rel": "rooms",
      "href": "https://localhost:8643/DcmConsole/rest/rooms?parentId=46"
    },
    {
      "rel": "update",
      "href": "https://localhost:8643/DcmConsole/rest/datacenters/46"
    },
    {
      "rel": "delete",
      "href": "https://localhost:8643/DcmConsole/rest/datacenters/46"
    },
    {
      "rel": "serverFirmwareOutlier",
      "href":
"https://localhost:8643/DcmConsole/rest/datacenters/46/serverFirmwareOutlier"
    },
    {
```

```
      "rel": "maxInletTemperature",
      "href":
"https://localhost:8643/DcmConsole/rest/datacenters/46/maxInletTemperature"
    },
    {
      "rel": "deviceHealthStatistics",
      "href":
"https://localhost:8643/DcmConsole/rest/datacenters/46/deviceHealthStatistics"
    },
    {
      "rel": "coolingAnomaly",
      "href":
"https://localhost:8643/DcmConsole/rest/datacenters/46/coolingAnomaly"
    },
    {
      "rel": "powerDataSummary",
      "href":
"https://localhost:8643/DcmConsole/rest/datacenters/46/powerDataSummary"
    }
  ]
}
```


8.4.6 Response field description

Field	Type	Description
cefConfig	Struct	Carbon emission factor configuration struct which has two fields. One is cef, the other unit.
cefConfig.cef	Array	Carbon emission factor two-dimensional array. The first dimension is 12, which represents the month. The second dimension is 24, which means 24 hours.
cefConfig.uint	Integer	Carbon emission factor unit. The value 0 means the unit is kilogram, and the value 1 means the unit is pound.

For other fields, refer to the chapter "Add a data center".

8.4.7 Errors

Except the generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The data center id does not exist.

8.5 Get power of a data center

8.5.1 Request URL

```
/rest/datacenters/{dcId}/power
```

The parameter "dcId" is the ID of the datacenter.

8.5.2 HTTP request method

GET

8.5.3 Request example

https://localhost:8643/DcmConsole/rest/datacenters/1/power?latest=true,
https://localhost:8643/DcmConsole/rest/datacenters/1/power?latest=false&&startDateTime=2019-01-17T10:00%2B08:00&&endDateTime=2019-01-17 T10:00%2B08:00&&granularity=monitoringPeriod&&page=0&&itemsPerPage=10

8.5.4 Request field description

Field	Type	Attribute	Description

latest	Boolean	Optional	The value "true" means to get the latest power. In this case, all other parameters will be ignored. The default value will be "true".
startDateTime	String	Optional	The start date time.
endDateTime	String	Optional	The end date time. NULL value means the current time.
page	Integer	Optional	The result supports pagination. This parameter means the page number starting from 0.
itemsPerPage	Integer	Optional	The number of the items on one page.
granularity	String	Optional	The possible values: monitoringPeriod The current monitoring interval. For example, if the current monitoring period is 3 minutes, the granularity will be 3 minutes. 1hour 1 hour. 6hours 6 hours. 24hours 24 hours. 7 days 7 days. 14days 14 days. By default, if the time range exceeds 1 week, the minimal granularity should be 1 hour.

8.5.5 Response example

```
{
  "totalItems": 121,
  "itemsPerPage": 2,
  "page": 0,
  "dataList": [
    {
      "value": 2816,
      "dateTime": "2019-01-17T10:00:00+08:00"
    },
    {
      "value": 2820,
      "dateTime": "2019-01-17T10:03:00+08:00"
    }
  ]
}
```

8.5.6 Response field description

Field	Type	Description
totalItems	Integer	The total items when pagination parameters are specified in the request.
itemsPerPage	Integer	The item count per page.
page	Integer	The current page.
dataList	Array	The item list.
dateTime	String	The timestamp of the data. The format is "yyyy-MM-dd'T'HH:mm:ssXXX".
value	Integer	The value of the data.

8.5.7 Errors

Except the generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The data center id does not exist.
INVALID_PAGE	The page value is invalid.
INVALID_ITEMS_PER_PAGE	Invalid item number per page.
INVALID_GRANULARITY	Invalid granularity.
START_TIME_IS_REQUIRED	Start time is missing or the value is invalid.
START_TIME_BEFORE_END_TIME	Start time is not before the end time.
INVALID_PARAMETER	Invalid parameter.

8.6 Get thermal of a data center

8.6.1 Request URL

/rest/datacenters/{dcId}/thermal

The parameter "dcId" is the ID of the datacenter.

8.6.2 HTTP request method

GET

8.6.3 Request example

https://localhost:8643/DcmConsole/rest/datacenters/1/thermal?latest=true,
https://localhost:8643/DcmConsole/rest/datacenters/1/thermal?latest=false&&startDateTime=2019-01-17T10:00%2B08:00&&endDateTime=2019-01-17T 10:00%2B08:00&&page=0&&itemsPerPage=10&&granularity=monitoringPeriod

8.6.4 Request field description

Field	Type	Attribute	Description
latest	Boolean	Optional	The value "true" means to get the latest power. In this case, other parameters will be ignored. The default value will be "true".
startDateTime	String	Optional	The start date time.
endDateTime	String	Optional	The end date time. NULL value means the current time.
page	Integer	Optional	The result supports pagination. This parameter means the page number starting from 0.
itemsPerPage	Integer	Optional	The number of items on one page.
granularity	String	Optional	The possible values: monitoringPeriod The current monitoring interval. For example, if the current monitoring period is 3 minutes, the granularity will be 3 minutes. 1hour 1 hour. 6hours 6 hours. 24hours 24 hours. 14days 14 days.

			By default, if the time range exceeds 1 week, the minimal granularity should be 1 hour.

8.6.5 Response example

```
{
  "totalItems": 2,
  "itemsPerPage": 2,
  "page": 0,
  "dataList": [
    {
      "dateTime": "2019-01-17T10:00:00+08:00",
      "value": 30
    },
    {
      "dateTime": "2019-01-17T10:03:00+08:00",
      "value": 29
    }
  ]
}
```

8.6.6 Response field description

Field	Type	Description
totalItems	Integer	The total items when pagination parameters are specified in the request.
itemsPerPage	Integer	The item count per page.
page	Integer	The current page.
dataList	Array	The item list.
dateTime	String	The date time of the data. The format is "yyyy-MM-dd'T'HH:mm:ssXXX".
value	Integer	The value of the data.

8.6.7 Errors

Except the generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The data center id does not exist.
INVALID_PAGE	The page value is invalid.
INVALID_ITEMS_PER_PAGE	Invalid item number per page.
INVALID_GRANULARITY	Invalid granularity.
START_TIME_IS_REQUIRED	Start time is missing or the value is invalid.
START_TIME_BEFORE_END_TIME	Start time is not before the end time.
INVALID_PARAMETER	Invalid parameter.

8.7 Get health of a data center

8.7.1 Request URL

```
/rest/datacenters/{dcId}/health
```

The parameter "dcId" is the ID of the datacenter.

8.7.2 HTTP request method

GET

8.7.3 Request example

https://localhost:8643/DcmConsole/rest/datacenters/1/health

8.7.4 Request field description

N/A.

8.7.5 Response example


```
{
  "unsupported": 8,
  "unknown": 1,
  "fault": 0,
  "normal": 1,
  "components": [
    {
      "component": "SYSTEM",
      "count": 0
    },
    {
      "component": "STORAGE",
      "count": 0
    },
    {
      "component": "BATTERY",
      "count": 0
    },
    {
      "component": "VOLTAGE",
      "count": 0
    },
    {
      "component": "PROCESSOR",
      "count": 0
    },
    {
      "component": "FPGA",
      "count": 0
    },
    {
      "component": "IO_MODULE",
      "count": 0
    },
    {
      "component": "MANAGEMENT_MODULE",
      "count": 0
    },
    {
      "component": "POWER_SUPPLY",
      "count": 0
    },
    {
      "component": "FAN",
      "count": 0
    },
    {
      "component": "MEMORY",
      "count": 0
    },
    {
```

```
        "component": "TEMPERATURE",
        "count": 0
      },
      {
        "component": "OPTANE_DIMM",
        "count": 0
      },
      {
        "component": "BLADE",
        "count": 0
      }
    ]
  }
```

8.7.6 Response field description

Field	Type	Description
unsupported	Integer	The number of unsupported devices.
unknown	Integer	The number of unknown devices.
fault	Integer	The number of fault devices.
normal	Integer	The number of normal devices.
components	Array	Component health data array.
component	String	The component name. Possible values include but not limited: SYSTEM STORAGE BATTERY VOLTAGE PROCESSOR FPGA IO_MODULE MANAGEMENT_MODULE POWER_SUPPLY FAN MEMORY TEMPERATURE OPTANE_DIMM BLADE
count	Integer	The fault component counter.

8.7.7 Errors

Except the generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The data center id does not exist.

8.8 Get inventory of a data center

8.8.1 Request URL

```
/rest/datacenters/{dcId}/inventory
```

The parameter "dcId" is the ID of the datacenter.

8.8.2 HTTP request method

GET

8.8.3 Request example

https://localhost:8643/DcmConsole/rest/datacenters/1/inventory

8.8.4 Request field description

N/A.

8.8.5 Response example


```
{
  "totalDeviceCount": 3,
  "supportedDeviceCount": 1,
  "details": {
    "resourceCategories": [
      {
        "categoryType": "PROCESSOR",
        "categoryProperties": [
          {
            "categoryPropertyName": "ProcessorCount",
            "categoryPropertyValue": "4"
          }
        ]
      },
      {
        "categoryType": "MEMORY",
        "categoryProperties": [
          {
            "categoryPropertyName": "TotalCapacityGiB",
            "categoryPropertyValue": "16.0"
          }
        ]
      },
      {
        "categoryType": "STORAGE",
        "categoryProperties": [
          {
            "categoryPropertyName": "DiskTotalCapacityGB",
            "categoryPropertyValue": "0.0"
          }
        ]
      }
    ]
  }
}
```


8.8.6 Response field description

Field	Type	Description
totalDeviceCount	Integer	The number of devices.
supportedDeviceCount	Integer	The number of devices with the capability to get inventory data.
details	Object	The detailed inventory data.
resourceCategories	Array	The resource categories.
categoryType	String	Category type. Possible values include but not limited to: STORAGE PROCESSOR MEMORY
categoryProperties	Array	The properties of this category.

categoryPropertyName	String	The category property name.
categoryPropertyValue	String	The category property value.

8.8.7 Errors

Except the generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The id does not exist.

8.9 Get server firmware outlier of a data center

8.9.1 Request URL

/rest/datacenters/{dcId}/serverFirmwareOutlier

The parameter "dcId" is the ID of the datacenter. The value "-1" means all of datacenters.

8.9.2 HTTP request method

GET

8.9.3 Request example

https://localhost:8643/DcmConsole/rest/datacenters/1/serverFirmwareOutlier

8.9.4 Request field description

N/A.

8.9.5 Response example

```
{
  "content": [
    {
      "model": "Supermicro CMM",
      "version": "3.35",
      "count": 2
    },
    {
      "model": "Node Manager 4.0 Intel Corporation - S2600WFT",
      "version": "1.35",
      "count": 3
    }
  ]
}
```

8.9.6 Response field description

Field	Type	Description
content	Array	The item list.
model	String	Device model.
version	String	The anomaly version.
count	Integer	The anomaly device count with the version.

8.9.7 Errors

Except the generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The data center id does not exist.

8.10 Get maximum inlet temperature of a data center

8.10.1 Request URL

```
/rest/datacenters/{dcId}/maxInletTemperature
```

The parameter "dcId" is the ID of the datacenter. The value "-1" means all of datacenters.

8.10.2 HTTP request method

GET

8.10.3 Request example

https://localhost:8643/DcmConsole/rest/datacenters/1/maxInletTemperature

8.10.4 Request field description

N/A.

8.10.5 Response example

```
{
  "maxTemperature": 37,
  "dcName": "dc-1"
}
```

8.10.6 Response field description

Field	Type	Description
maxTemperature	Integer	The maximum inlet temperature of the specified data center. The unit is Celsius.
dcName	String	The data center name which only exists when the parameter "dcId" is not "-1".

8.10.7 Errors

Except the generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The data center id does not exist.

8.11 Get top N high inlet temperature rooms of a data center

8.11.1 Request URL

```
/rest/datacenters/{dcId}/topInletTemperatureRooms
```

The parameter "dcId" is the ID of the datacenter. The value "-1" means all of datacenters.

8.11.2 HTTP request method

GET

8.11.3 Request example

https://localhost:8643/DcmConsole/rest/dataceneters/1/topInletTemperatureRooms?top=2

8.11.4 Request field description

Field	Type	Attribute	Description
top	Integer	Required	The top number.

8.11.5 Response example

```
{
  "links": [
    {
```

```
      "rel": "self",
      "href":
"https://localhost:8643/DcmConsole/rest/dataceneters/1/topInletTemperatureRooms?
top=3"
    }
  ],
  "content": [
    {
      "roomId": "2",
      "roomName": "room1",
      "maxTempValue": "30",
      "dcName": "dc-1",
      "dcId": "1"
    },
    {
      "roomId": "3",
      "roomName": "room2",
      "maxTempValue": "30",
      "dcName": "dc-1",
      "dcId": "1"
    }
  ]
}
```


8.11.6 Response field description

Field	Type	Description
roomId	Integer	The room id.
roomName	String	The room name.
maxTempValue	Integer	The current maximum inlet temperature value. The unit is Celsius.
dcName		The data center name.
dcId		The data center id.

8.11.7 Errors

Except the generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The data center id does not exist.
INVALID_TOP_PARAMETER	The value of parameter "top" is invalid.

8.12 Get device health statistics of a data center

8.12.1 Request URL

/rest/datacenters/{dcId}/deviceHealthStatistics

The parameter "dcId" is the ID of the datacenter. The value "-1" means all datacenters.

8.12.2 HTTP request method

GET

8.12.3 Request example

https://localhost:8643/DcmConsole/rest/datacenters/1/deviceHealthStatistics

8.12.4 Request field description

N/A.

8.12.5 Response example

```
{
  "unsupported": 3,
  "unknown": 1,
  "fault": 1,
  "normal": 100
}
```

8.12.6 Response field description

Field	Type	Description
unsupported	Integer	The number of devices without health monitoring capability.
unknown	Integer	The number of devices with health data monitoring capability but the health data are not retrieved.
fault	Object	The number of devices with unhealthy data.
normal	Array	The number of devices with healthy data.

8.12.7 Errors

Except the generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The data center id does not exist.

8.13 Get cooling anomaly status of a data center

8.13.1 Request URL

```
/rest/datacenters/{dcId}/coolingAnomaly
```

The parameter "dcId" is the ID of the datacenter. The value "-1" means all of datacenters.

8.13.2 HTTP request method

GET

8.13.3 Request example

https://localhost:8643/DcmConsole/rest/datacenters/1/coolingAnomaly

8.13.4 Request field description

N/A.

8.13.5 Response example

```
{
  "content": [
    {
      "roomId": 2,
      "roomName": "room-2",
      "dateTime": "2019-01-17T10:03:00+08:00",
      "maxTemperature": 28,
      "minTemperature": 24,
      "avgTemperature": 26,
      "maxTemperatureDelta": 2,
      "minTemperatureDelta": -2,
      "avgTemperatureDelta": 1
```

```
    },
    {
      "roomId": 3,
      "roomName": "room-3",
      "dateTime": "2019-01-17T10:03:00+08:00",
      "maxTemperature": 28,
      "minTemperature": 24,
      "avgTemperature": 26,
      "maxTemperatureDelta": 2,
      "minTemperatureDelta": -2,
      "avgTemperatureDelta": 1
    }
  ]
}
```


8.13.6 Response field description

Field	Type	Description
content	Array	The data list.
roomId	Integer	The id of the room with cooling anomaly.
roomName	String	The name of the room with cooling anomaly.
dateTime	String	The date time of the data. The format is "yyyy-MM-dd'T'HH:mm:ssXXX".
maxTemparature	Integer	The maximum temperature of the room. The unit is Celsius.
minTemparature	Integer	The minimum temperature of the room. The unit is Celsius.
avgTemperature	Integer	The average temperature of the room. The unit is Celsius.
maxTemperatureDelta	Integer	The change of the maximum temperature. Positive value means increase, and negative value means decrease.
minTemperatureDelta	Integer	The change of the minimum temperature. Positive value means increase, and negative value means decrease.
avgTemperatureDelta	Integer	The change of the average temperature. Positive value means increase, and negative value means decrease.

8.13.7 Errors

Except the generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The data center id does not exist.

8.14 Get power data summary of a data center

8.14.1 Request URL

/rest/datacenters/{dcId}/powerDataSummary

The parameter "dcId" is the ID of the datacenter. The value "-1" means all datacenters.

8.14.2 HTTP request method

GET

8.14.3 Request example

https://localhost:8643/DcmConsole/rest/datacenters/1/powerDataSummary?top=4&&startDateTime=2019-01-17T10:00%2B08:00&&endDateTime=2019-01-17T 10:00%2B08:00"

8.14.4 Request field description

Field	Type	Attribute	Description
top	Integer	Required	The number of top power consumption devices.
startDateTime	String	Required	The start date time.
endDateTime	String	Optional	The end date time. If this field is not specified, we will use the current time.

8.14.5 Response example


```
{
  "PeakPower": 10000,
  "topServerPowerDatas": [
    {
      "id": 10,
      "name": "server-1",
      "dateTime": "2019-01-17T10:03:00+08:00",
      "power": 1000
    },
    {
      "id": 11,
      "name": "server-2",
      "dateTime": "2019-01-17T10:03:00+08:00",
      "power": 900
    }
  ],
  "topChassisPowerDatas": [
    {
      "id": 20,
      "name": "chassis-1",
      "dateTime": "2019-01-17T10:03:00+08:00",
      "power": 1000
    },
    {
      "id": 21,
      "name": "chassis -2",
      "dateTime": "2019-01-17T10:03:00+08:00",
      "power": 900
    }
  ]
}
```


8.14.6 Response field description

Field	Type	Description
PeakPower	Integer	The peak power of IT equipment in the datacenter or all datacenters.
topServerPowerDatas	Array	The top power consumption servers.
topChassisPowerDatas	Array	The top power consumption chassis.
id	Integer	The device id.
name	String	The device name.
power	Integer	The IT equipment power of the device.
dateTime	String	The date time of the data. The format is "yyyy-MM-dd'T'HH:mm:ssXXX".

8.14.7 Errors

Except the generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The datacenter id does not exist.
TOP_PARAMTER_IS_REQUIRED	The parameter "top" is required.
INVALID_TOP_PARAMETER	The value of parameter "top" is invalid.
START_TIME_IS_REQUIRED	Start time is missing or the value is invalid.
START_TIME_BEFORE_END_TIME	Start time is not before the end time.
INVALID_PARAMETER	Invalid parameter.

8.15 Get hotspots of a data center

8.15.1 Request URL

/rest/datacenters/{dcId}/hotspots

The parameter "dcId" is the ID of the datacenter. The value "-1" means all of datacenters.

8.15.2 HTTP request method

GET

8.15.3 Request example

https://localhost:8643/DcmConsole/rest/datacenters/1/hotspots?&&page=0&&itemsPerPage=10&&sortField=value&&sortOrder=desc

8.15.4 Request field description

Field	Type	Attribute	Description
page	Integer	Optional	The result supports pagination. This parameter means the page number starting from 0. The default value is 0.
itemsPerPage	Integer	Optional	The number of items on one page. The default value is 10.
sortField	String	Optional	The possible values: name value path
sortOrder		Optional	The possible values: asc desc

8.15.5 Response example


```
{
  "totalItems": 2,
  "itemsPerPage": 2,
  "page": 0,
  "dataList": [
    {
      "deviceId": "1",
      "entityName": "server-1",
      "entityPath": [
        "4",
        "3",
        "2",
        "1"
      ],
```

```
      "entityPathStr": "dc > room > row > rack",
      "dateTime": "2019-01-17T10:00:00+08:00",
      "value": 30
    },
    {
      "deviceId": "2",
      "entityName": "server-2",
      "entityPath": [
        "4",
        "3",
        "2",
        "1"
      ],
      "entityPathStr": "dc > room > row > rack",
      "dateTime": "2019-01-17T10:00:00+08:00",
      "value": 30
    }
  ]
}
```


8.15.6 Response field description

Field	Type	Description
totalItems	Integer	The total items when pagination parameters are specified in the request.
itemsPerPage	Integer	The item count per page.
page	Integer	The current page.
dataList	Array	The item list.
dateTime	String	The date time of the data. The format is "yyyy-MM-dd'T'HH:mm:ssXXX".
value	Integer	The temperature value in Celsius.

8.15.7 Errors

Except the generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The data center id does not exist.
INVALID_PAGE	The page value is invalid.
INVALID_ITEMS_PER_PAGE	Invalid item number per page.
INVALID_SORT_FIELD	Invalid sort field.
INVALID_SORT_ORDER	Invalid sort order.

8.16 Get power capacity of a data center

8.16.1 Request URL

/rest/datacenters/{dcId}/powerCapacity

The parameter "dcId" is the ID of the datacenter. The value "-1" means all datacenters.

8.16.2 HTTP request method

GET

8.16.3 Request example

https://localhost:8643/DcmConsole/rest/datacenters/1/powerCapacity

8.16.4 Request field description

N/A.

8.16.5 Response example

```
{
  "powerCapacity": "100000",
  "powerUsed": "10000"
}
```

8.16.6 Response field description

Field	Type	Description
powerCapacity	Integer	The power capacity of the specified data center. The unit is watt.
powerUsed	Integer	The used power of the specified data center. The unit is watt.

8.16.7 Errors

Except the generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The data center id does not exist.

8.17 Get rooms of a data center

Refer to the chapter "Get rooms".

8.18 Update a datacenter

8.18.1 Request URL

```
/rest/datacenters/{dcId}
```

"dcId" is the id of the target data center.

8.18.2 HTTP request method

PUT

8.18.3 Request example

```
{
  "name": "dc1",
  "description": "description",
  "powerCapacity": 10000,
  "pue": 1.8
}
```

8.18.4 Request field description

The client can specify one or some of these fields to update.

Field	Type	Attribute	Description
Name	String	Optional	The name of the data center.
description	String	Optional	Description for the data center.
powerCapacity	Integer	Optional	The total power capacity of the data center. The unit is W.
Pue	Double	Optional	The Power Usage Effectiveness of the data center.

electricityRate	Double	Optional	The electricity rate of the data center. If the value is set, it will overwrite the global electricity rate of the data center. The value "-1" means to remove the setting for the data center.
cefConfig	Struct	Optional	Carbon emission factor configuration struct which has two fields. One is cef, the other unit.
cefConfig.cef	Array	Optional	Carbon emission factor two-dimensional array. The first dimension is 12, which represents the month. The second dimension is 24, which means 24 hours.
cefConfig.uint	Integer	Optional	Carbon emission factor unit. The value 0 means the unit is kilogram, and the value 1 means the unit is pound.

8.18.5 Response example


```
{
  "id": 1,
  "name": "dc1",
  "description": "description",
  "powerCapacity": 10000,
  "pue": "1.8",
  "links": [
    {
      "rel": "self",
      "href": "https://localhost:8643/DcmConsole/rest/datacenters/1"
    },
    {
      "rel": "power",
      "href":
"https://localhost:8643/DcmConsole/rest/datacenters/46/power{?latest,startDateTi
me,endDateTime,page,itemsPerPage,granularity}"
    },
    {
      "rel": "thermal",
      "href":
"https://localhost:8643/DcmConsole/rest/datacenters/1/thermal{?latest,startDateT
ime,endDateTime,page,itemsPerPage,granularity}"
    },
    {
      "rel": "health",
      "href": "https://localhost:8643/DcmConsole/rest/datacenters/1/health"
    },
    {
      "rel": "rooms",
      "href": "https://localhost:8643/DcmConsole/rest/rooms?parentId=1"
    },
    {
      "rel": "update",
      "href": "https://localhost:8643/DcmConsole/rest/datacenters/1"
    },
    {
      "rel": "delete",
      "href": "https://localhost:8643/DcmConsole/rest/datacenters/1"
    }
  ]
}
```


8.18.6 Response field description

Refer to the chapter "Add a data center".

8.18.7 Errors

Except the generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The data center id does not exist.
INVALID_NAME	The name is invalid.
INVALID_POWER_CAPACITY	The power capacity field value is invalid.
INVALID_PUE	The PUE field value is invalid.
INVALID_ELECTRICITY_RATE	Invalid electricity rate.

8.19 Delete a data center

8.19.1 Request URL

```
/rest/datacenters/{id}
```

"id" is the id of the target data center.

8.19.2 HTTP request method

DELETE

8.19.3 Request example

https://localhost:8643/DcmConsole/rest/datacenters/1

8.19.4 Request field description

N/A.

8.19.5 Response example

N/A.

8.19.6 Response field description

N/A.

8.19.7 Errors

Except the generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The data center id does not exist.

***Note**: All devices of the data center will be deleted.*

8.20 Add a room

8.20.1 Request URL

```
/rest/rooms
```

8.20.2 HTTP request method

POST

8.20.3 Request example

```
{
  "parentId": 1,
  "name": "room1",
```

```
  "description": "description",
  "powerCapacity": 10000,
  "pue": 1.8
}
```

8.20.4 Request field description

Field	Type	Attribute	Description
parentId	Integer	Required	The id of the parent data center.
Name	String	Required	The name of the room.
description	String	Optional	Description for the room.
powerCapacity	Integer	Optional	The total power capacity of the room. The unit is W.
pue	Double	Optional	The Power Usage Effectiveness of the room.
cefConfig	Struct	Optional	Carbon emission factor configuration structwhich has two fields. One is cef, the other unit.
cefConfig.cef	Array	Optional	Carbon emission factor two-dimensional array. The first dimension is 12, which represents the month. The second dimension is 24, which means 24 hours.
cefConfig.uint	Integer	Optional	Carbon emission factor unit. The value 0 means the unit is kilogram, and the value 1 means the unit is pound.

8.20.5 Response example

```
{
  "parentId": 1,
  "id": 77,
  "name": "room1",
  "description": "dsadsasda",
  "powerCapacity": 10000,
  "pue": 2,
  "links": [
    {
      "rel": "self",
      "href": "https://localhost:8643/DcmConsole/rest/rooms/77"
    },
    {
      "rel": "power",
      "href":
"https://localhost:8643/DcmConsole/rest/rooms/77/power{?latest,startDateTime,end
DateTime,page,itemsPerPage,granularity}"
    },
    {
      "rel": "thermal",
      "href":
"https://localhost:8643/DcmConsole/rest/rooms/77/thermal{?latest,startDateTime,e
ndDateTime,page,itemsPerPage,granularity}"
    },
    {
      "rel": "health",
      "href": "https://localhost:8643/DcmConsole/rest/rooms/77/health"
    },
    {
      "rel": "rows",
      "href": "https://localhost:8643/DcmConsole/rest/rows?parentId=77"
    },
    {
      "rel": "update",
```

```
      "href": "https://localhost:8643/DcmConsole/rest/rooms/77"
    },
    {
      "rel": "delete",
      "href": "https://localhost:8643/DcmConsole/rest/rooms/77"
    }
  ]
}
```

8.20.6 Response field description

Field	Type	Description
id	Integer	The id of the room.

Refer to Request field description for other fields.

8.20.7 Errors

Except the generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
NAME_IS_REQUIRED	The name field is required, or the value is missing.
INVALID_NAME	The name is invalid.
INVALID_POWER_CAPACITY	The power capacity field value is invalid.
INVALID_PUE	The PUE field value is invalid.
INVALID_PARENT_ID	The parent id is not an id of a data center.
TARGET_NOT_EXISTED	The parent id doesn't exist.

8.21 Get rooms

8.21.1 Request URL

```
/rest/rooms?parentId=-1
```

The optional parameter "parentId" is the id of the parent data center. If this parameter is missing or presented with a value -1, all rooms will be returned. Otherwise, only the rooms of the data center will be returned.

8.21.2 HTTP request method

GET

8.21.3 Request example

https://localhost:8643/DcmConsole/rest/rooms

8.21.4 Request field description

N/A

8.21.5 Response example

```
{
  "links": [
    {
      "rel": "self",
      "href": "https://localhost:8643/DcmConsole/rest/rooms?parentId=-1"
```

```
    },
    {
      "rel": "add",
      "href": "https://localhost:8643/DcmConsole/rest/rooms"
    }
  ],
  "content": [
    {
      "parentId": 46,
      "id": 77,
      "name": "room-3",
      "description": "dsadsasda",
      "powerCapacity": 10000,
      "pue": 2,
      "links": [
        {
          "rel": "self",
          "href": "https://localhost:8643/DcmConsole/rest/rooms/77"
        },
        {
          "rel": "power",
          "href":
"https://localhost:8643/DcmConsole/rest/rooms/77/power{?latest,startDateTime,end
DateTime,page,itemsPerPage,granularity}"
        },
        {
          "rel": "thermal",
          "href":
"https://localhost:8643/DcmConsole/rest/rooms/77/thermal{?latest,tartDateTime,en
dDateTime,page,itemsPerPage,granularity}"
        },
        {
          "rel": "health",
          "href": "https://localhost:8643/DcmConsole/rest/rooms/77/health"
        },
        {
          "rel": "rows",
          "href": "https://localhost:8643/DcmConsole/rest/rows?parentId=77"
        },
        {
          "rel": "update",
          "href": "https://localhost:8643/DcmConsole/rest/rooms/77"
        },
        {
          "rel": "delete",
          "href": "https://localhost:8643/DcmConsole/rest/rooms/77"
        }
      ]
    },
    {
      "parentId": 46,
      "id": 47,
      "name": "room",
      "powerCapacity": 10000,
      "pue": 2,
      "links": [
        {
          "rel": "self",
          "href": "https://localhost:8643/DcmConsole/rest/rooms/47"
        },
        {
          "rel": "power",
```

```
            "href":
"https://localhost:8643/DcmConsole/rest/rooms/47/power{?latest,startDateTime,end
DateTime,page,itemsPerPage,granularity}"
          },
          {
            "rel": "thermal",
            "href":
"https://localhost:8643/DcmConsole/rest/rooms/47/thermal{?latest,startDateTime,e
ndDateTime,page,itemsPerPage,granularity}"
          },
          {
            "rel": "health",
            "href": "https://localhost:8643/DcmConsole/rest/rooms/47/health"
          },
          {
            "rel": "rows",
            "href": "https://localhost:8643/DcmConsole/rest/rows?parentId=47"
          },
          {
            "rel": "update",
            "href": "https://localhost:8643/DcmConsole/rest/rooms/47"
          },
          {
            "rel": "delete",
            "href": "https://localhost:8643/DcmConsole/rest/rooms/47"
          }
        ]
      }
    ]
}
```


8.21.6 Response field description

Field	Type	Description
links	Array	The links to the API.
content	Array	All rooms.

Refer to the chapter "Add a room" for other field definitions.

8.21.7 Errors

Except the generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
INVALID_PARENT_ID	The parent id is not an id of a data center.
TARGET_NOT_EXISTED	The parent id doesn't exist.

8.22 Get a room

8.22.1 Request URL

```
/rest/rooms/{roomId}
```

The parameter "roomId" is the ID of the room.

8.22.2 HTTP request method

GET

8.22.3 Request example

https://localhost:8643/DcmConsole/rest/rooms/1

8.22.4 Request field description

N/A.

8.22.5 Response example


```
{
  "parentId": 46,
  "id": 46,
  "name": "room1",
  "powerCapacity": 1000,
  "pue": 2,
  "links": [
    {
      "rel": "self",
      "href": "https://localhost:8643/DcmConsole/rest/rooms/46"
    },
    {
      "rel": "power",
      "href":
"https://localhost:8643/DcmConsole/rest/rooms/46/power{?latest,startDateTime,end
DateTime,page,itemsPerPage,granularity}"
    },
    {
      "rel": "thermal",
      "href":
"https://localhost:8643/DcmConsole/rest/rooms/46/thermal{?latest,startDateTime,e
ndDateTime,page,itemsPerPage,granularity}"
    },
    {
      "rel": "health",
      "href": "https://localhost:8643/DcmConsole/rest/rooms/46/health"
    },
    {
      "rel": "rows",
      "href": "https://localhost:8643/DcmConsole/rest/rows?parentId=46"
    },
    {
      "rel": "update",
      "href": "https://localhost:8643/DcmConsole/rest/rooms/46"
    },
    {
      "rel": "delete",
      "href": "https://localhost:8643/DcmConsole/rest/rooms/46"
    }
  ]
}
```


8.22.6 Response field description

Refer to the chapter "Add a room".

8.22.7 Errors

Except the generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The room id does not exist.

8.23 Get power of a room

8.23.1 Request URL

```
/rest/rooms/{roomId}/power
```

The parameter "roomId" is the ID of the room

8.23.2 HTTP request method

GET

8.23.3 Request example

https://localhost:8643/DcmConsole/rest/rooms/1/power?latest=true,
https://localhost:8643/DcmConsole/rest/rooms/1/power?latest=false&&startDateTime=2019-01-17 10:00%2B08:00&&endDateTime=2019-01-17 10:00%2B08:00&&page=0&&itemsPerPage=10&&granularity=monitoringPeriod

8.23.4 Request field description

Refer to the chapter "Get power of a data center".

8.23.5 Response example

Refer to the chapter "Get power of a data center".

8.23.6 Response field description

Refer to the chapter "Get power of a data center".

8.23.7 Errors

Refer to the chapter "Get power of a data center".

8.24 Get thermal of a room

8.24.1 Request URL

/rest/rooms/{roomId}/thermal

The parameter "roomId" is the ID of the room.

8.24.2 HTTP request method

GET

8.24.3 Request example

https://localhost:8643/DcmConsole/rest/rooms/1/thermal?latest=true,
https://localhost:8643/DcmConsole/rest/rooms/1/thermal?latest=false&&startDateTime=2019-01-17 10:00%2B08:00&&endDateTime=2019-01-17 10:00%2B08:00&&page=0&&itemsPerPage=10&&granularity=monitoringPeriod

8.24.4 Request field description

Refer to the chapter "Get thermal of a data center".

8.24.5 Response example

Refer to the chapter "Get thermal of a data center".

8.24.6 Response field description

Refer to the chapter "Get thermal of a data center".

8.24.7 Errors

Refer to the chapter "Get thermal of a data center".

8.25 Get health of a room

8.25.1 Request URL

/rest/rooms/{roomId}/health

The parameter "roomId" is the ID of the room.

8.25.2 HTTP request method

GET

8.25.3 Request example

https://localhost:8643/DcmConsole/rest/rooms/1/health

8.25.4 Request field description

N/A.

8.25.5 Response example

Refer to the chapter "Get health of a data center".

8.25.6 Response field description

Refer to the chapter "Get health of a data center".

8.25.7 Errors

Except the generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The room id does not exist.

8.26 Get inventory of a room

8.26.1 Request URL

/rest/rooms/{roomId}/inventory

The parameter "roomId" is the ID of the room.

8.26.2 HTTP request method

GET

8.26.3 Request example

https://localhost:8643/DcmConsole/rest/rooms/1/inventory

8.26.4 Request field description

N/A.

8.26.5 Response example

Refer to the chapter "Get inventory of a data center".

8.26.6 Response field description

Refer to the chapter "Get inventory of a data center".

8.26.7 Errors

Except the generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The id does not exist.

8.27 Get rows of a room

Refer to the chapter "Get rows".

8.28 Update a room

8.28.1 Request URL

```
/rest/rooms/{id}
"id" is the id of the target room.
```

8.28.2 HTTP request method

PUT

8.28.3 Request example

```
{
  "parentId": "1",
  "name": "room1",
  "description": "description",
  "powerCapacity": 10000,
  "pue": 1.8
}
```

8.28.4 Request field description

The client can update one or some of these fields .

Field	Type	Attribute	Description
parentId	Integer	Optional	The id of the parent. If the parent Id is different from the current one, the room will be moved to the new parent.
Name	String	Optional	The name of the room.
Description	String	Optional	Description for the room.
powerCapacity	Integer	Optional	The total power capacity of the room. The unit is W.
Pue	Float	Optional	The Power Usage Effectiveness of the room.
cefConfig	Struct	Optional	Carbon emission factor configuration structwhich has two fields. One is cef, the other unit.
cefConfig.cef	Array	Optional	Carbon emission factor two-dimensional array. The first dimension is 12, which represents the month. The second dimension is 24, which means 24 hours.
cefConfig.uint	Integer	Optional	Carbon emission factor unit. The value 0 means the unit is kilogram, and the value 1 means the unit is pound.

8.28.5 Response example

```
{
  "parentId": 46,
  "id": 46,
  "name": "dc1",
  "powerCapacity": 1000,
  "pue": 2,
  "links": [
    {
      "rel": "self",
      "href": "https://localhost:8643/DcmConsole/rest/rooms/46"
    },
    {
      "rel": "power",
      "href":
"https://localhost:8643/DcmConsole/rest/rooms/46/power{?latest,startDateTime,end
DateTime,page,itemsPerPage,granularity}"
```

```
    },
    {
      "rel": "thermal",
      "href":
"https://localhost:8643/DcmConsole/rest/rooms/46/thermal{?latest,startDateTime,e
ndDateTime,page,itemsPerPage,granularity}"
    },
    {
      "rel": "health",
      "href": "https://localhost:8643/DcmConsole/rest/rooms/46/health"
    },
    {
      "rel": "rows",
      "href": "https://localhost:8643/DcmConsole/rest/rows?parentId=46"
    },
    {
      "rel": "update",
      "href": "https://localhost:8643/DcmConsole/rest/rooms/46"
    },
    {
      "rel": "delete",
      "href": "https://localhost:8643/DcmConsole/rest/rooms/46"
    }
  ]
}
```


8.28.6 Response field description

Refer to the chapter "Get a room".

8.28.7 Errors

Except the generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The room id or the parent data center id does not exist.
INVALID_PARENT_ID	The parent id is not a data center id.
INVALID_NAME	The name is invalid.
INVALID_POWER_CAPACITY	The power capacity field value is invalid.
INVALID_PUE	The PUE field value is invalid.

8.29 Delete a room

8.29.1 Request URL

```
/rest/rooms/{id}
```

"id" is the id of the target room.

8.29.2 HTTP request method

DELETE

8.29.3 Request example

https://localhost:8643/DcmConsole/rest/rooms/1

8.29.4 Request field description

N/A.

8.29.5 Response example

N/A.

8.29.6 Response field description

N/A.

8.29.7 Errors

Except the generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The room id does not exist.

Note: *All devices in the room will be deleted.*

8.30 Add a row

8.30.1 Request URL

```
/rest/rows
```

8.30.2 HTTP request method

POST

8.30.3 Request example

```
{
  "parentId": 1,
  "name": "row1",
  "description": "description",
  "powerCapacity": 10000,
  "weightCapacity": 10000
}
```

8.30.4 Request field description

Field	Type	Attribute	Description
parentId	Integer	Required	The id of the parent room.
Name	String	Required	The name of the row.
description	String	Optional	Description for the row.
powerCapacity	Integer	Optional	The total power capacity of the row. The unit is W.
weightCapacity	Integer	Optional	The weight capacity. The unit is KG.

8.30.5 Response example

```
{
  "parentId": 1,
  "id": 77,
  "name": "row1",
  "description": "dsadsasda",
  "powerCapacity": 10000,
  "links": [
    {
      "rel": "self",
      "href": "https://localhost:8643/DcmConsole/rest/rows/77"
    },
    {
      "rel": "power",
```

```
      "href":
"https://localhost:8643/DcmConsole/rest/rows/77/power{?latest,startDateTime,endD
ateTime,page,itemsPerPage,granularity}"
    },
    {
      "rel": "thermal",
      "href":
"https://localhost:8643/DcmConsole/rest/rows/77/thermal{?latest,startDateTime,en
dDateTime,page,itemsPerPage,granularity}"
    },
    {
      "rel": "health",
      "href": "https://localhost:8643/DcmConsole/rest/rows/77/health"
    },
    {
      "rel": "racks",
      "href": "https://localhost:8643/DcmConsole/rest/racks?parentId=77"
    },
    {
      "rel": "update",
      "href": "https://localhost:8643/DcmConsole/rest/rows/77"
    },
    {
      "rel": "delete",
      "href": "https://localhost:8643/DcmConsole/rest/rows/77"
    }
  ]
}
```


8.30.6 Response field description

Field	Type	Description
id	Integer	The id of the row.

Refer to Request field description for other fields.

8.30.7 Errors

Except the generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
NAME_IS_REQUIRED	The name field is required.
INVALID_NAME	The name is invalid.
INVALID_POWER_CAPACITY	The power capacity field value is invalid.
INVALID_WEIGHT_CAPACITY	The weight capacity value is invalid.
INVALID_PARENT_ID	The parent id is not a room id.
TARGET_NOT_EXISTED	The parent id doesn't exist.

8.31 Get rows

8.31.1 Request URL

```
/rest/rows?parentId=-1
```

The optional parameter "parentId" is the id of the parent room. If this parameter is missing or presented with a value -1, all rows will be returned. Otherwise, only the rows of the room will be returned.

8.31.2 HTTP request method

GET

8.31.3 Request example

https://localhost:8643/DcmConsole/rest/rows

8.31.4 Request field description

N/A

8.31.5 Response example


```
{
  "links": [
    {
      "rel": "self",
      "href": "https://localhost:8643/DcmConsole/rest/rows?parentId=-1"
    },
    {
      "rel": "add",
      "href": "https://localhost:8643/DcmConsole/rest/rows"
    }
  ],
  "content": [
    {
      "parentId": 46,
      "id": 77,
      "name": "row-3",
      "description": "dsadsasda",
      "powerCapacity": 10000,
      "links": [
        {
          "rel": "self",
          "href": "https://localhost:8643/DcmConsole/rest/rows/77"
        },
        {
          "rel": "power",
          "href":
"https://localhost:8643/DcmConsole/rest/rows/77/power{?latest,startDateTime,endD
ateTime,page,itemsPerPage,granularity}"
        },
        {
          "rel": "thermal",
          "href":
"https://localhost:8643/DcmConsole/rest/rows/77/thermal{?latest,startDateTime,en
dDateTime,page,itemsPerPage,granularity}"
        },
        {
          "rel": "health",
          "href": "https://localhost:8643/DcmConsole/rest/rows/77/health"
        },
        {
          "rel": "racks",
          "href": "https://localhost:8643/DcmConsole/rest/racks?parentId=77"
        },
        {
          "rel": "update",
          "href": "https://localhost:8643/DcmConsole/rest/rows/77"
        },
        {
          "rel": "delete",
          "href": "https://localhost:8643/DcmConsole/rest/rows/77"
        }
```

```
        ]
      },
      {
        "parentId": 46,
        "id": 47,
        "name": "row2",
        "powerCapacity": 10000,
        "links": [
          {
            "rel": "self",
            "href": "https://localhost:8643/DcmConsole/rest/rows/47"
          },
          {
            "rel": "power",
            "href":
"https://localhost:8643/DcmConsole/rest/rows/47/power{?latest,startDateTime,endD
ateTime,page,itemsPerPage,granularity}"
          },
          {
            "rel": "thermal",
            "href":
"https://localhost:8643/DcmConsole/rest/rows/47/thermal{?latest,startDateTime,en
dDateTime,page,itemsPerPage,granularity}"
          },
          {
            "rel": "health",
            "href": "https://localhost:8643/DcmConsole/rest/rows/47/health"
          },
          {
            "rel": "racks",
            "href": "https://localhost:8643/DcmConsole/rest/racks?parentId=47"
          },
          {
            "rel": "update",
            "href": "https://localhost:8643/DcmConsole/rest/rows/47"
          },
          {
            "rel": "delete",
            "href": "https://localhost:8643/DcmConsole/rest/rows/47"
          }
        ]
      }
    ]
}
```


8.31.6 Response field description

Field	Type	Description
links	Array	The links to the API.
content	Array	All rows.

Refer to the chapter "Add a row" for other field definitions.

8.31.7 Errors

Except the generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
INVALID_PARENT_ID	The parent id is not a room id.
TARGET_NOT_EXISTED	The parent id doesn't exist.

8.32 Get a row

8.32.1 Request URL

/rest/rows/{rowId}

The parameter "rowId" is the ID of the row.

8.32.2 HTTP request method

GET

8.32.3 Request example

https://localhost:8643/DcmConsole/rest/rows/1

8.32.4 Request field description

N/A.

8.32.5 Response example


```
{
  "parentId": 46,
  "id": 46,
  "name": "row1",
  "powerCapacity": 1000,
  "links": [
    {
      "rel": "self",
      "href": "https://localhost:8643/DcmConsole/rest/rows/46"
    },
    {
      "rel": "power",
      "href":
"https://localhost:8643/DcmConsole/rest/rows/46/power{?latest,startDateTime,endD
ateTime,page,itemsPerPage,granularity}"
    },
    {
      "rel": "thermal",
      "href":
"https://localhost:8643/DcmConsole/rest/rows/46/thermal{?latest,startDateTime,en
dDateTime,page,itemsPerPage,granularity}"
    },
    {
      "rel": "health",
      "href": "https://localhost:8643/DcmConsole/rest/rows/46/health"
    },
    {
      "rel": "racks",
      "href": "https://localhost:8643/DcmConsole/rest/racks?parentId=46"
    },
    {
      "rel": "update",
      "href": "https://localhost:8643/DcmConsole/rest/rows/46"
    },
    {
      "rel": "delete",
      "href": "https://localhost:8643/DcmConsole/rest/rows/46"
    }
  ]
}
```


8.32.6 Response field description

Refer to the chapter "Add a row".

8.32.7 Errors

Except the generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The row id does not exist.

8.33 Get power of a row

8.33.1 Request URL

/rest/rows/{rowId}/power

The parameter "rowId" is the ID of the row

8.33.2 HTTP request method

GET

8.33.3 Request example

https://localhost:8643/DcmConsole/rest/rows/1/power?latest=true,
https://localhost:8643/DcmConsole/rest/rows/1/power?latest=false&&startDateTime=2019-01-17 10:00%2B08:00&&endDateTime=2019-01-17 10:00%2B08:00&&page=0&&itemsPerPage=10&&granularity=monitoringPeriod

8.33.4 Request field description

Refer to the chapter "Get power of a data center".

8.33.5 Response example

Refer to the chapter "Get power of a data center".

8.33.6 Response field description

Refer to the chapter "Get power of a data center".

8.33.7 Errors

Refer to the chapter "Get power of a data center".

8.34 Get thermal of a row

8.34.1 Request URL

/rest/rows/{rowId}/thermal

The parameter "rowId" is the ID of the row.

8.34.2 HTTP request method

GET

8.34.3 Request example

https://localhost:8643/DcmConsole/rest/rows/1/thermal?latest=true,
https://localhost:8643/DcmConsole/rest/rows/1/thermal?latest=false&&startDateTime=2019-01-17 10:00%2B08:00&&endDateTime=2019-01-17 10:00%2B08:00&&page=0&&itemsPerPage=10&&granularity=monitoringPeriod

8.34.4 Request field description

Refer to the chapter "Get thermal of a data center".

8.34.5 Response example

Refer to the chapter "Get thermal of a data center".

8.34.6 Response field description

Refer to the chapter "Get thermal of a data center".

8.34.7 Errors

Refer to the chapter "Get thermal of a data center".

8.35 Get health status of a row

8.35.1 Request URL

/rest/rows/{rowId}/health

The parameter "rowId" is the ID of the row.

8.35.2 HTTP request method

GET

8.35.3 Request example

https://localhost:8643/DcmConsole/rest/rows/1/health

8.35.4 Request field description

N/A.

8.35.5 Response example

Refer to the chapter "Get health status of a data center".

8.35.6 Response field description

Refer to the chapter "Get health status of a data center".

8.35.7 Errors

Except the generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The row id does not exist.

8.36 Get inventory of a row

8.36.1 Request URL

/rest/rows/{rowId}/inventory

The parameter "rowId" is the ID of the row.

8.36.2 HTTP request method

GET

8.36.3 Request example

https://localhost:8643/DcmConsole/rest/rows/1/inventory

8.36.4 Request field description

N/A.

8.36.5 Response example

Refer to the chapter "Get inventory of a data center".

8.36.6 Response field description

Refer to the chapter "Get inventory of a data center".

8.36.7 Errors

Except the generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The id does not exist.

8.37 Get racks of a row

Refer to the chapter "Get racks".

8.38 Update a row

8.38.1 Request URL

```
/rest/rows/{id}
```

"id" is the id of the target row.

8.38.2 HTTP request method

PUT

8.38.3 Request example

```
{
  "parentId": "1",
  "name": "row1",
  "description": "description",
  "powerCapacity": 10000
}
```

8.38.4 Request field description

The client can specify one or some of these fields to update.

Field	Type	Attribute	Description
parentId	Integer	Optional	The id of the parent. If the parent Id is different from the current one, the row will be moved to the new parent.
Name	String	Optional	The name of the row.
description	String	Optional	Description for the row.
powerCapacity	Integer	Optional	The total power capacity of the row. The unit is W.
weightCapacity	Double	Optional	The weight capacity.

8.38.5 Response example

```
{
  "parentId": 46,
  "id": 46,
  "name": "row1",
  "powerCapacity": 1000,
  "links": [
    {
      "rel": "self",
      "href": "https://localhost:8643/DcmConsole/rest/rows/46"
    },
    {
      "rel": "power",
      "href":
"https://localhost:8643/DcmConsole/rest/rows/46/power{?latest,startDateTime,endD
ateTime,page,itemsPerPage,granularity}"
    },
```

```
    {
      "rel": "thermal",
      "href":
"https://localhost:8643/DcmConsole/rest/rows/46/thermal{?latest,startDateTime,en
dDateTime,page,itemsPerPage,granularity}"
    },
    {
      "rel": "health",
      "href": "https://localhost:8643/DcmConsole/rest/rows/46/health"
    },
    {
      "rel": "rows",
      "href": "https://localhost:8643/DcmConsole/rest/rows?parentId=46"
    },
    {
      "rel": "update",
      "href": "https://localhost:8643/DcmConsole/rest/rows/46"
    },
    {
      "rel": "delete",
      "href": "https://localhost:8643/DcmConsole/rest/rows/46"
    }
  ]
}
```


8.38.6 Response field description

Refer to the chapter "Get a row".

8.38.7 Errors

Except the generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The row id or the parent room id does not exist.
INVALID_PARENT_ID	The parent id is not a room id.
INVALID_NAME	The name is invalid.
INVALID_POWER_CAPACITY	The power capacity field value is invalid.
INVALID_WEIGHT_CAPACITY	The weight capacity value is invalid.

8.39 Delete a row

8.39.1 Request URL

```
/rest/rows/{id}
```

"id" is the id of the target row.

8.39.2 HTTP request method

DELETE

8.39.3 Request example

https://localhost:8643/DcmConsole/rest/rows/1

8.39.4 Request field description

N/A.

8.39.5 Response example

N/A.

8.39.6 Response field description

N/A.

8.39.7 Errors

Except the generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The row id does not exist.

***Note**: All devices of the row will be deleted.*

8.40 Add a rack

8.40.1 Request URL

```
/rest/racks
```

8.40.2 HTTP request method

POST

8.40.3 Request example

```
{
  "parentId": 1,
  "name": "rack1",
  "description": "description",
  "powerCapacity": 10000,
  "spaceCapacity": 42,
  "gridX": 1,
  "gridY": 1,
  "pduPowerAsRackPower": "false",
  "weight": 100,
  "miscWeight": 50,
  "weightCapacity": 1000,
  "slotOrder": 0
}
```

8.40.4 Request field description

Field	Type	Attribute	Description
parentId	Integer	Required	The id of the parent data center.
Name	String	Required	The name of the rack.
description	String	Optional	Description for the rack.
powerCapacity	Integer	Required	The total power capacity of the rack. The unit is W.
spaceCapacity	Integer	Required	The total space capacity of the rack. The unit is U.
gridX	Integer	Optional	The grid horizontal coordinate.
gridY	Integer	Optional	The grid vertical coordinate.
pduPowerAsRackPower	Boolean	Optional	If this field value is true, the rack power will be retrieved from the PDU of the rack. If the field is not presented, the default value is false.
weight	Integer	Optional	Weight. The unit is KG.
miscWeight	Integer	Optional	Misc weight.
weightCapacity	Integer	Optional	Weight capacity.

slotOrder	Integer	Optional	Slot order. 0 means "Bottom-Up". 1 means "Top-Down".

8.40.5 Response example


```
{
  "parentId": 1,
  "id": 77,
  "name": "rack1",
  "description": "dsadsasda",
  "powerCapacity": 10000,
  "spaceCapacity": 42,
  "gridX": 1,
  "gridY": 1,
  "pduPowerAsRackPower": false,
  "slotOrder": 0,
  "links": [
    {
      "rel": "self",
      "href": "https://localhost:8643/DcmConsole/rest/racks/77"
    },
    {
      "rel": "power",
      "href":
"https://localhost:8643/DcmConsole/rest/racks/77/power{?latest,startDateTime,end
DateTime,page,itemsPerPage,granularity}"
    },
    {
      "rel": "thermal",
      "href":
"https://localhost:8643/DcmConsole/rest/racks/77/thermal{?latest,startDateTime,e
ndDateTime,page,itemsPerPage,granularity}"
    },
    {
      "rel": "health",
      "href": "https://localhost:8643/DcmConsole/rest/racks/77/health"
    },
    {
      "rel": "rows",
      "href": "https://localhost:8643/DcmConsole/rest/rows?parentId=77"
    },
    {
      "rel": "update",
      "href": "https://localhost:8643/DcmConsole/rest/racks/77"
    },
    {
      "rel": "delete",
      "href": "https://localhost:8643/DcmConsole/rest/racks/77"
    }
  ]
}
```


8.40.6 Response field description

Field	Type	Description
id	Integer	The id of the rack.

Refer to Request field description for other fields.

8.40.7 Errors

Except the generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description

NAME_IS_REQUIRED	The name field is required, or the value is missing.
INVALID_NAME	The name is invalid.
POWER_CAPACITY_IS_REQUIRED	Power capacity is required.
INVALID_POWER_CAPACITY	The power capacity field value is invalid.
SPACE_CAPACITY_IS_REQUIRED	Space capacity is required.
INVALID_SPACE_CAPACITY	The space capacity is invalid.
INVALID_GRID_X_VALUE	Invalid grid x value.
INVALID_GRID_Y_VALUE	Invalid grid y value.
INVALID_WEIGHT	Invalid weight.
INVALID_MISC_WEIGHT	Invalid misc weight.
INVALID_WEIGHT_CAPACITY	Invalid weight capacity.
INVALID_PARENT_ID	The parent id is not a row id.
TARGET_NOT_EXISTED	The parent id doesn't exist.
INVALID_SLOT_ORDER	Invalid slot order.

8.41 Get racks

8.41.1 Request URL

/rest/racks?parentId=-1

The optional parameter "parentId" is the id of the parent row. If this parameter is not missing or presented with a value -1, all racks will be returned. Otherwise, only the racks of the row will be returned.

8.41.2 HTTP request method

GET

8.41.3 Request example

https://localhost:8643/DcmConsole/rest/racks

8.41.4 Request field description

N/A

8.41.5 Response example


```
{
  "links": [
    {
      "rel": "self",
      "href": "https://localhost:8643/DcmConsole/rest/racks?parentId=-1"
    },
    {
      "rel": "add",
      "href": "https://localhost:8643/DcmConsole/rest/racks"
    }
  ],
  "content": [
    {
      "parentId": 1,
      "id": 77,
```

```
        "name": "rack1",
        "description": "dsadsasda",
        "powerCapacity": 10000,
        "spaceCapacity": 42,
        "gridX": 1,
        "gridY": 1,
        "pduPowerAsRackPower": false,
        "slotOrder": 0,
        "links": [
          {
            "rel": "self",
            "href": "https://localhost:8643/DcmConsole/rest/racks/77"
          },
          {
            "rel": "power",
            "href":
"https://localhost:8643/DcmConsole/rest/racks/77/power{?latest,startDateTime,end
DateTime,page,itemsPerPage,granularity}"
          },
          {
            "rel": "thermal",
            "href":
"https://localhost:8643/DcmConsole/rest/racks/77/thermal{?latest,startDateTime,e
ndDateTime,page,itemsPerPage,granularity}"
          },
          {
            "rel": "health",
            "href": "https://localhost:8643/DcmConsole/rest/racks/77/health"
          },
          {
            "rel": "devices",
            "href": "https://localhost:8643/DcmConsole/rest/devices?parentId=77"
          },
          {
            "rel": "update",
            "href": "https://localhost:8643/DcmConsole/rest/racks/77"
          },
          {
            "rel": "delete",
            "href": "https://localhost:8643/DcmConsole/rest/racks/77"
          }
        ]
      },
      {
        "parentId": 1,
        "id": 47,
        "name": "rack1",
        "description": "dsadsasda",
        "powerCapacity": 10000,
        "spaceCapacity": 42,
        "gridX": 1,
        "gridY": 1,
        "pduPowerAsRackPower": false,
        "slotOrder": 0,
        "links": [
          {
            "rel": "self",
            "href": "https://localhost:8643/DcmConsole/rest/racks/47"
          },
          {
            "rel": "power",
```

```
            "href":
"https://localhost:8643/DcmConsole/rest/racks/47/power{?latest,startDateTime,end
DateTime,page,itemsPerPage,granularity}"
          },
          {
            "rel": "thermal",
            "href":
"https://localhost:8643/DcmConsole/rest/racks/47/thermal{?latest,startDateTime,e
ndDateTime,page,itemsPerPage,granularity}"
          },
          {
            "rel": "health",
            "href": "https://localhost:8643/DcmConsole/rest/racks/47/health"
          },
          {
            "rel": "devices",
            "href": "https://localhost:8643/DcmConsole/rest/devices?parentId=47"
          }
        ]
      }
    ]
}
```


8.41.6 Response field description

Field	Type	Description
links	Array	The links to the API.
content	Array	All racks.

Refer to the chapter "Add a rack" for other field definitions.

8.41.7 Errors

Except the generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
INVALID_PARENT_ID	The parent id is not a row id.
TARGET_NOT_EXISTED	The parent id doesn't exist.
INVALID_SLOT_ORDER	Invalid slot order.

8.42 Get a rack

8.42.1 Request URL

```
/rest/racks/{rackId}
```

The parameter "rackId" is the ID of the rack.

8.42.2 HTTP request method

GET

8.42.3 Request example

https://localhost:8643/DcmConsole/rest/racks/1

8.42.4 Request field description

N/A.

8.42.5 Response example


```
{
  "parentId": 1,
```

```
  "id": 46,
  "name": "rack1",
  "description": "dsadsasda",
  "powerCapacity": 10000,
  "spaceCapacity": 42,
  "gridX": 1,
  "gridY": 1,
  "pduPowerAsRackPower": "false",
  "slotOrder": 0,
  "links": [
    {
      "rel": "self",
      "href": "https://localhost:8643/DcmConsole/rest/racks/46"
    },
    {
      "rel": "power",
      "href":
"https://localhost:8643/DcmConsole/rest/racks/46/power{?latest,startDateTime,end
DateTime,page,itemsPerPage,granularity}"
    },
    {
      "rel": "thermal",
      "href":
"https://localhost:8643/DcmConsole/rest/racks/46/thermal{?latest,startDateTime,e
ndDateTime,page,itemsPerPage,granularity}"
    },
    {
      "rel": "health",
      "href": "https://localhost:8643/DcmConsole/rest/racks/46/health"
    },
    {
      "rel": "devices",
      "href": "https://localhost:8643/DcmConsole/rest/devices?parentId=46"
    },
    {
      "rel": "update",
      "href": "https://localhost:8643/DcmConsole/rest/racks/46"
    },
    {
      "rel": "delete",
      "href": "https://localhost:8643/DcmConsole/rest/racks/46"
    }
  ]
}
```


8.42.6 Response field description

Refer to the chapter "Add a rack".

8.42.7 Errors

Except the generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The rack id does not exist.
INVALID_SLOT_ORDER	Invalid slot order.

8.43 Get power of a rack

8.43.1 Request URL

```
/rest/racks/{rackId}/power
```

The parameter "rackId" is the ID of the rack

8.43.2 HTTP request method

GET

8.43.3 Request example

https://localhost:8643/DcmConsole/rest/racks/1/power?latest=true,
https://localhost:8643/DcmConsole/rest/racks/1/power?latest=false&&startDateTime=2019-01-17T10:00%2B08:00&&endDateTime=2019-01-17T 10:00%2B08:00&& granularity=monitoringPeriod&&page=0&&itemsPerPage=10

8.43.4 Request field description

Refer to the chapter "Get power of a data center".

8.43.5 Response example

Refer to the chapter "Get power of a data center".

8.43.6 Response field description

Refer to the chapter "Get power of a data center".

8.43.7 Errors

Refer to the chapter "Get power of a data center".

8.44 Get thermal of a rack

8.44.1 Request URL

/rest/racks/{rackId}/thermal

The parameter "rackId" is the ID of the rack.

8.44.2 HTTP request method

GET

8.44.3 Request example

https://localhost:8643/DcmConsole/rest/racks/1/thermal?latest=true,
https://localhost:8643/DcmConsole/rest/racks/1/thermal?latest=false&&startDateTime=2019-01-17T10:00%2B08:00&&endDateTime=2019-01-17T 10:00%2B08:00&& granularity=monitoringPeriod&&page=0&&itemsPerPage=10

8.44.4 Request field description

Refer to the chapter "Get thermal of a data center".

8.44.5 Response example

Refer to the chapter "Get thermal of a data center".

8.44.6 Response field description

Refer to the chapter "Get thermal of a data center".

8.44.7 Errors

Refer to the chapter "Get thermal of a data center".

8.45 Get health status of a rack

8.45.1 Request URL

/rest/racks/{rackId}/health

The parameter "rackId" is the ID of the rack.

8.45.2 HTTP request method

GET

8.45.3 Request example

https://localhost:8643/DcmConsole/rest/racks/1/health

8.45.4 Request field description

N/A.

8.45.5 Response example

Refer to the chapter "Get health status of a data center".

8.45.6 Response field description

Refer to the chapter "Get health status of a data center".

8.45.7 Errors

Except the generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The rack id does not exist.

8.46 Get inventory of a rack

8.46.1 Request URL

/rest/racks/{rackId}/inventory

The parameter "rackId" is the ID of the rack.

8.46.2 HTTP request method

GET

8.46.3 Request example

https://localhost:8643/DcmConsole/rest/racks/1/inventory

8.46.4 Request field description

N/A.

8.46.5 Response example

Refer to the chapter "Get inventory of a data center".

8.46.6 Response field description

Refer to the chapter "Get inventory of a data center".

8.46.7 Errors

Except the generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The id does not exist.

8.47 Get devices of a rack

Refer to the chapter "Get devices".

8.48 Update a rack

8.48.1 Request URL

```
/rest/racks/{id}
```

"id" is the id of the target rack.

8.48.2 HTTP request method

PUT

8.48.3 Request example

```
{
  "parentId": 1,
  "id": 46,
  "name": "rack1",
  "description": "description1",
  "powerCapacity": 10000,
  "spaceCapacity": 42,
  "gridX": 1,
  "gridY": 1,
  "pduPowerAsRackPower": "false",
  "slotOrder": 0
}
```

8.48.4 Request field description

The client can specify one or some of these fields to update.

Field	Type	Attribute	Description
parentId	Integer	Optional	The id of the parent data center. If the parent id is presented with different value from the existing parent of the rack, the rack will be moved to the new parent.
name	String	Optional	The name of the rack.
description	String	Optional	Description for the rack.
powerCapacity	Integer	Optional	The total power capacity of the rack. The unit is W.
spaceCapacity	Integer	Optional	The total space capacity of the rack. The unit is U.
gridX	Integer	Optional	The grid horizontal coordinate.
gridY	Integer	Optional	The grid vertical coordinate.
pduPowerAsRackPower	Boolean	Optional	If this field value is true, the rack power will be retrieved from the PDU under the rack.
slotOrder	Integer	Optional	Slot order. 0 means "Bottom-Up". 1 means "Top-Down".

8.48.5 Response example

```
{
  "parentId": 1,
  "id": 46,
  "name": "rack1",
  "description": "dsadsasda",
  "powerCapacity": 10000,
  "spaceCapacity": 42,
  "gridX": 1,
  "gridY": 1,
  "pduPowerAsRackPower": "false",
  "slotOrder": 0,
```

```
  "links": [
    {
      "rel": "self",
      "href": "https://localhost:8643/DcmConsole/rest/racks/46"
    },
    {
      "rel": "power",
      "href":
"https://localhost:8643/DcmConsole/rest/racks/46/power{?latest,startDateTime,end
DateTime,page,itemsPerPage,granularity}"
    },
    {
      "rel": "thermal",
      "href":
"https://localhost:8643/DcmConsole/rest/racks/46/thermal{?latest,startDateTime,e
ndDateTime,page,itemsPerPage,granularity}"
    },
    {
      "rel": "health",
      "href": "https://localhost:8643/DcmConsole/rest/racks/46/health"
    },
    {
      "rel": "devices",
      "href": "https://localhost:8643/DcmConsole/rest/devices?parentId=46"
    },
    {
      "rel": "update",
      "href": "https://localhost:8643/DcmConsole/rest/racks/46"
    },
    {
      "rel": "delete",
      "href": "https://localhost:8643/DcmConsole/rest/racks/46"
    }
  ]
}
```


8.48.6 Response field description

Refer to the chapter "Get a rack".

8.48.7 Errors

Except the generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The rack id or the parent id does not exist.
INVALID_PARENT_ID	The parent id is not a row id.
INVALID_NAME	The name is invalid.
INVALID_POWER_CAPACITY	The power capacity field value is invalid.
INVALID_SPACE_CAPACITY	The space capacity is invalid.
INVALID_GRID_X_VALUE	Invalid grid x value.
INVALID_GRID_Y_VALUE	Invalid grid y value.
INVALID_WEIGHT	Invalid weight.
INVALID_MISC_WEIGHT	Invalid misc weight.
INVALID_WEIGHT_CAPACITY	Invalid weight capacity.

INVALID_SLOT_ORDER	Invalid slot order.

8.49 Delete a rack

8.49.1 Request URL

```
/rest/racks/{id}
```

"id" is the id of the target rack.

8.49.2 HTTP request method

DELETE

8.49.3 Request example

https://localhost:8643/DcmConsole/rest/racks/1

8.49.4 Request field description

N/A.

8.49.5 Response example

N/A.

8.49.6 Response field description

N/A.

8.49.7 Errors

Except the generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The rack id does not exist.

***Note**: All devices of the rack will be deleted.*

8.50 Add a device

8.50.1 Request URL

```
/rest/devices
```

8.50.2 HTTP request method

POST

8.50.3 Request example

```
{
  "parentId": 1,
  "name": "device1",
  "address": "192.168.1.1",
  "userName": "user1",
  "password": "password1",
  "deratedPower": 400,
  "protocol": "IPMI",
  "size": 1,
  "location": 1,
  "description": "device1",
  "birthday": "2019-01-17",
  "weight": 100,
  "vendor": "ANY"
}
```

8.50.4 Request field description

Field	Type	Attribute	Description
parentId	Integer	Optional	The id of the parent rack or chassis. For blade, it means the parent chassis ID. If this field is not missing or presented with a value -1, it will be a standalone device.
protocol	String	Required	The possible values: IPMI SSH HTTPS SNMPv1v2c SNMPv3 INBAND_PROTOCOL WS_MAN Unmanaged server It is used to manage the servers without interfaces to monitor power and thermal. Unmanaged network device Unmanaged storage device Unmanaged chassis
Name	String	Required	The name of the device.
address	String	Optional	The IP address or host name of the device. It is useless when the type is unmanaged device.
userName	String	Optional	The user name to connect to the device.
password	String	Optional	The password to connect to the device.
key	String	Optional	The BMC key which is useful only when the type is IPMI.
deratedPower	Integer	Optional	The derated power for none-chassis devices such as servers, PDU, UPS and so on. The unit is W.
namePlatePower	Integer	Optional	The nameplate power of the device.
snmpCommunityString	String	Optional	SNMP community string. It is useful only when the protocol is SNMPV1v2c.
snmpEncryptionPassword	String	Optional	SNMP encryption password. It is useful only when the protocol is SNMPv3.
snmpAuthenticationProtocol	String	Optional	SNMP authentication protocol. It is useful only when the protocol is SNMPv3. Possible values: AuthMD5 AuthSHA1 AuthHMAC128SHA224 AuthHMAC192SHA256 AuthHMAC256SHA384 AuthHMAC384SHA512
snmpEncryptionProtocol	String	Optional	SNMP encryption protocol. It is useful only when the type is SNMPv3. Possible values: PrivDES Priv3DES PrivAES128 PrivAES192 PrivAES256
httpsPort	Integer	Optional	The HTTPs port. The default value is 443.
sshPort	Integer	Optional	The SSH port. The default value is 22.

authenticateEntity	Boolean	Optional	Whether it is need to validate the device. For https, the device certificate needs to be imported into DCM key store if the field value is true. By default, the value is true for most protocols.
size	Integer	Optional	The size of the device. For servers, the default value is 1. For chassis, the default value is 10. The unit is U.
location	Integer	Optional	The slot of the device.
typicalPower	Integer	Optional	The typical power of the device. It is useful only when the type is unmanaged device.
idlePower	Integer	Optional	The idle power of the device. It is useful only when the type is unmanaged device.
peakPower	Integer	Optional	The peak power of the device. It is useful only when the type is unmanaged device.
model	String	Optional	The device model of the device. It is useful only when the type is unmanaged device.
osType	String	Optional	The in-band OS type. The possible values: Windows Linux Xen ESX
osAddress	String	Optional	The OS IP address or host name. It is used to specify the in-band OS information when the protocol is IPMI.
osUserName	String	Optional	The user name to access the OS. It is used to specify the in-band OS information when the protocol is IPMI.
osPassword	String	Optional	The user password to access the OS. It is used to specify the in-band OS information when the protocol is IPMI.
osSshPort	Integer	Optional	The in-band OS SSH port whose default value is 22. It is used to specify the in-band OS information when the protocol is IPMI.
description	String	Optional	Description for the device.
reportInletTemp	Boolean	Optional	Whether to report inlet temperature. Currently it is useful only for chatsworth PDU.
birthday	String	Optional	The birthday of the device. The format can be "yyyy-MM-dd" or "yyyy-MM-dd'T'HH:mm:ssZ". For examples, "2018-12-16" or "2018-12-16T00:00:00+08:00".
customConsoleURL	String	Optional	The specified device management console URL.
width	Integer	Optional	The horizontal grid count of a chassis or a blade. It is used to locate blades in a chassis.
height	Integer	Optional	The vertical grid count of a chassis or a blade. It is used to locate blades in a chassis.
startX	Integer	Optional	The horizontal location of a blade in a chassis.
startY	Integer	Optional	The vertical location of a blade in a chassis.
epr	String	Optional	The action of a device with emergency power reduction. Possible values: Minimize Power Consumption Shutdown

			No Action
weight	Double	Optional	Device weight. The unit is KG.
platformId	String	Optional	The id used to identify the device which is only for unmanaged devices when adding unmanaged devices. Unmanaged devices include: Unmanaged server Unmanaged network device Unmanaged storage device Unmanaged chassis Platform ID can be specified only when adding an unmanaged device and cannot be modified. It will be converted to upper case by DCM. When a managed device is being added, DCM will specify it internally.
onlyCurrentUserProvisioning	Boolean	Optional	Whether it is need to set the current user as provisioning user. It is useful only when the current user is an administrator.
customAttributes	Array	Optional	Custom attribute array. Each element contains a "customAttributeName" field and a "customAttributeValue" field.
customAttributeName	String	Optional	The name of the custom attribute. When the object exists in the "customAttributes" array, the field is required. The field can be specified not only for the custom attributes defined by the end-users, but also for some built-in attributes: Owner Business Unit Contact Warranty Expiration Vendor Email Except the built-in attributes, the custom attribute should exist before its value is specified. Refer to the chapter "Add a custom attribute" for details.
customAttributeValue	String	Optional	The value of the custom attribute. For the date type attribute, the format is "yyyy-MM-dd". For example, "2020-10-11".
vendor	String	Optional	The vendor name. If any specific vendor is specified, DCM will add the device only if the identified devices match the known vendor. If not specified or "ANY" specified, all known vendors are supported. "OTHERS" means all other vendors but excluding all the enumerated vendors. The possible values: ANY, ALTOS, AMD, APC, ARISTA, AVOCENT, BAYTECH, BROCADE, CHATSWORTH, CISCO,

			CONTOSO, DELL, EATON, EMC, EMERSON, ENLOGIC, EXTREMENETWORKS, F5, FUJITSU, H3C, HITACHI, HOFFMAN, HP, HUAWEI, IBM, INSPUR, INTEL, KONTRON, LENOVO, MELLANOX, NETAPP, NETTRIX, ORACLE, RARITAN, RSA, SERVERTECH, SIEMON, SUGON, SUPERMICRO, VERTIV, ZTE, OTHERS

8.50.5 Response example


```
{
  "parentId": 60,
  "id": 83,
  "protocol": "IPMI",
  "deviceType": "Server",
  "name": "rest-server-2",
  "address": "sh-mke",
  "deratedPower": 600,
  "size": 1,
  "model": "Node Manager 1.5 Intel - ProLiant DL380 G7",
  "customConsoleURL": "https://10.239.157.21:443",
  "epr": "Minimize Power Consumption",
  "links": [
    {
      "rel": "self",
      "href": "https://localhost:8643/DcmConsole/rest/devices/83"
    },
    {
      "rel": "power",
      "href":
"https://localhost:8643/DcmConsole/rest/devices/83/power{?latest,startDateTime,e
ndDateTime,page,itemsPerPage,granularity}"
    },
    {
      "rel": "thermal",
      "href":
"https://localhost:8643/DcmConsole/rest/devices/83/thermal{?latest,startDateTime
,endDateTime,page,itemsPerPage,granularity}"
```

```
      },
      {
        "rel": "health",
        "href": "https://localhost:8643/DcmConsole/rest/devices/83/health"
      },
      {
        "rel": "update",
        "href": "https://localhost:8643/DcmConsole/rest/devices/83"
      },
      {
        "rel": "delete",
        "href": "https://localhost:8643/DcmConsole/rest/devices/83"
      }
    ]
  }
```


8.50.6 Response field description

Field	Type	Description
id	Integer	The id of the device.
deviceType	String	Device type. Possible values: Server Blade Enclosure PDU UPS Network device Storage device Unmanaged server Unmanaged network device Unmanaged storage device Unmanaged chassis

Refer to the chapter "Get devices" for other field definitions.

8.50.7 Errors

Except the generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The parent id does not exist.
NAME_IS_REQUIRED	Name is required.
INVALID_NAME	Invalid name.
INVALID_PROTOCOL	Invalid protocol.
INVALID_DERATED_POWER	Invalid derated power.
INVALID_NAMEPLATE_POWER	Invalid nameplate power.
INVALID_HTTPS_PORT	Invalid HTTPS port.
INVALID_SSH_PORT	Invalid SSH port.
INVALID_SIZE	Invalid device size.
INVALID_LOCATION	Invalid location.
INVALID_TYPICAL_POWER	Invalid typical power.

INVALID_IDLE_POWER	Invalid idle power.
INVALID_PEAK_POWER	Invalid peak power.
INVALID_OS_TYPE	Invalid OS type.
INVALID_WIDTH	Invalid width.
INVALID_HEIGHT	Invalid height.
INVALID_START_X	Invalid start X.
INVALID_START_Y	Invalid start Y.
INVALID_EPR	Invalid EPR action.
DEVICE_IS_ALREADY_MANAGED	Device is already managed.
ADDRESS_IS_REQUIRED	Address is required.
INVALID_PARENT_ID	The parent id is not an id of a rack or chassis. A chassis cannot be added to a chassis, so this error will also be reported in this case.
TARGET_NOT_EXISTED	The parent id doesn't exist.
DEVICE_NUMBER_LIMIT	Number of the device exceeds the license limit.
DEVICE_COMMUNICATION_FAILED	Device communication failed.
PLATFORM_ID_CANNOT_BE_SPECIFIED	Platform ID cannot be specified for the device. It can be specified only for the unmanaged device.
CUSTOM_ATTR_NAME_IS_REQUIRED	Custom attribute name is required.
INVALID_CUSTOM_ATTR_VALUE	The custom attribute value is invalid. Typically, the time format of a custom value is not "yyyy-MM-dd" for date type custom attribute.
INVALID_PARAMETER	Invalid parameter.
INSUFFICIENT_DEVICE_RESOURCES	Insufficient device resources.

8.51 Get devices

8.51.1 Request URL

/rest/devices?{parentId=-1}

The optional parameter "parentId" is the ID of the parent rack or chassis.

8.51.2 HTTP request method

GET

8.51.3 Request example

https://localhost:8643/DcmConsole/rest/devices?parentId=-1&&name=server

8.51.4 Request field description

Field	Type	Attribute	Description
parentId	Integer	Optional	The id of the parent rack or chassis. The value "-1" means all devices. If the field is not specified, the value "-1" will be used by default.
name	String	Optional	Devices whose name contain the specified string will be returned.
model	String	Optional	Devices whose model contain the specified string will be returned.
description	String	Optional	Devices whose description contain the specified string will be returned.
serialNumber	String	Optional	Devices whose serial number contain the specified string will be returned.
powerStatus	String	Optional	Devices whose power status equal to the specified string will be returned. Possible values: All On Off Unknown
monitoringStatus	String	Optional	Monitoring status. Possible values: All BeingMonitored FailToMonitor Offline
healthStatus	String	Optional	Health status. Possible values: All OK NOT OK Unknown

8.51.5 Response example


```
{
  "links": [
    {
      "rel": "self",
      "href": "https://localhost:8643/DcmConsole/rest/devices/51"
    },
    {
      "rel": "power",
      "href":
"https://localhost:8643/DcmConsole/rest/devices/51/power{?latest,startDateTime,e
ndDateTime,page,itemsPerPage,granularity}"
    },
    {
      "rel": "thermal",
      "href":
"https://localhost:8643/DcmConsole/rest/devices/51/thermal{?latest,startDateTime
,endDateTime,page,itemsPerPage,granularity}"
    },
    {
      "rel": "health",
      "href": "https://localhost:8643/DcmConsole/rest/devices/51/health"
    }
  ],
  "content": [
```

```
        {
          "parentId": 60,
          "id": 83,
          "protocol": "IPMI",
          "deviceType": "Server",
          "name": "rest-server-2",
          "address": "sh-mke",
          "deratedPower": 600,
          "size": 1,
          "model": "Node Manager 1.5 Intel - ProLiant DL380 G7",
          "customConsoleURL": "https://10.239.157.21:443",
          "epr": "Minimize Power Consumption",
          "powerStatus": "ON",
          "serialNumber": "MX84012",
          "capabilities": [
            {
              "capability": "power monitoring"
            },
            {
              "capability": "temperature monitoring"
            },
            {
              "capability": "power control"
            },
            {
              "capability": "CPU telemetry"
            }
          ],
          "firmwareVersions": [
            {
              "firmwareType": "MGMT_MODULE",
              "firmwareVersion": "0.20"
            }
          ]
        }
      ],
      "customAttributes": [
        {
          "customAttributeName": "Vendor"
        },
        {
          "customAttributeValue": "Intel"
        }
      ],
      "parentId": 50,
      "id": 51,
      "protocol": "Unmanaged server",
      "deviceType": "Unmanaged server",
      "name": "bb",
      "description": "sdasdfasfsa",
      "deratedPower": 400,
      "size": 0.5,
      "model": "",
      "birthday": "2019-01-18T16:03:55+08:00"
    }
```


8.51.6 Response field description

Field	Type	Description
parentId	Integer	The id of the parent rack or chassis. For blade, it means the ID of the parent chassis.
protocol	String	The possible values: IPMI

		SSH HTTPS SNMPv1v2c SNMPv3 INBAND_PROTOCOL WS_MAN Unmanaged server It is used to manage the servers without interfaces to monitor power and thermal. Unmanaged network device Unmanaged storage device Unmanaged chassis
name	String	The name of the device.
address	String	The IP address or host name of the device. It is useless when the type is unmanaged device.
deratedPower	Integer	The derated power for none-chassis devices such as servers, PDU, UPS and so on. The unit is W.
namePlatePower	Integer	The nameplate power of the device.
snmpAuthenticationProtocol	String	SNMP authentication protocol. It is useful only when the protocol is SNMPv3. Possible values: AuthMD5 AuthSHA1 AuthHMAC128SHA224 AuthHMAC192SHA256 AuthHMAC256SHA384 AuthHMAC384SHA512
snmpEncryptionProtocol	String	SNMP encryption protocol. It is useful only when the type is SNMPv3. Possible values: PrivDES Priv3DES PrivAES128 PrivAES192 PrivAES256
httpsPort	Integer	The HTTPs port. The default value is 443.
sshPort	Integer	The SSH port. The default value is 22.
authenticateEntity	Boolean	Whether it is need to validate the device. For https, the device certificate needs to be imported into DCM key store if the field value is true.
size	Integer	The size of the device. For servers, the default value is 1. For chassis, the default value is 10. The unit is U.
location	Integer	The slot of the device.
typicalPower	Integer	The typical power of the device. It is useful only when the type is unmanaged device.
idlePower	Integer	The idle power of the device. It is useful only when the type is unmanaged device.
peakPower	Integer	The peak power of the device. It is useful only when the type is unmanaged device.
model	String	The device model of the device. It is useful only when the type is unmanaged device.
osType	String	The in-band OS type. The possible values: Windows Linux Xen

		ESX
osAddress	String	The OS IP address or host name which is used to specify the in-band OS information when the protocol is IPMI.
osSshPort	Integer	The in-band OS SSH port with the default value 22. It is used to specify the in-band OS information when the protocol is IPMI.
description	String	Description for the device.
reportInletTemp	Boolean	Whether to report inlet temperature. Currently it is useful only for chatsworth PDU.
birthday	String	The birthday of the device. The format is "yyyy-MM-dd'T'HH:mm:ssZ". For example, 2019-02-03T00:00:00+08:00
customConsoleURL	String	The specified device management console URL.
width	Integer	The horizontal grid count of a chassis or a blade. It is used to locate blades in a chassis.
height	Integer	The vertical grid count of a chassis or a blade. It is used to locate blades in a chassis.
startX	Integer	The horizontal location of a blade in a chassis.
startY	Integer	The vertical location of a blade in a chassis.
epr	String	The action of a device with emergency power reduction. Possible values: Minimize Power Consumption Shutdown No Action
powerStatus		Possible values: ON OFF UNKNOWN
capabilities	Array	Capabilities of the device.
capability	String	Capability. The possible values: power monitoring temperature monitoring CPU telemetry power control
serialNumber	String	The serial number of a device.
firmwareVersions	Array	Firmware versions.
firmwareType	String	Possible values: MGMT_MODULE Management module firmware. For servers, it is the BMC firmware version.
firmwareVersion	String	Firmware version.
platformId	String	For unmanaged devices, it can be specified by the user when adding devices.
customAttributes	Array	Refer to the chapter "Add a device".
customAttributeName	String	Refer to the chapter "Add a device".
customAttributeValue	String	Refer to the chapter "Add a device".
path	String	The path of the device. The hierarchy is linked with symbol " > "(e.g. "dc > room > row > rack").
offline	Boolean	Whether the device is offline or not.
provisioningUser	String	The provisioning user of the device.

8.51.7 Errors

Except the generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The parent id does not exist.
INVALID_PARENT_ID	The parent id is not an id of a rack or a chassis.
INVALID_POWER_STATUS	Invalid power status.
INVALID_MONITOR_STATUS	Invalid monitoring status.
INVALID_HEALTH_STATUS	Invalid health status.

8.52 Get a device

8.52.1 Request URL

```
/rest/devices/{deviceId}
```

The parameter "deviceId" is the ID of the device.

8.52.2 HTTP request method

GET

8.52.3 Request example

https://localhost:8643/DcmConsole/rest/devices/1

8.52.4 Request field description

N/A.

8.52.5 Response example


```
{
  "parentId": 60,
  "id": 83,
  "protocol": "IPMI",
  "deviceType": "Server",
  "name": "rest-server-2",
  "address": "sh-mke",
  "deratedPower": 600,
  "size": 1,
  "model": "Node Manager 1.5 Intel - ProLiant DL380 G7",
  "customConsoleURL": "https://10.239.157.21:443",
  "epr": "Minimize Power Consumption",
  "links": [
    {
      "rel": "self",
      "href": "https://localhost:8643/DcmConsole/rest/devices/83"
    },
    {
      "rel": "power",
      "href":
"https://localhost:8643/DcmConsole/rest/devices/83/power{?latest,startDateTime,e
ndDateTime,page,itemsPerPage,granularity}"
    },
    {
      "rel": "thermal",
      "href":
"https://localhost:8643/DcmConsole/rest/devices/83/thermal{?latest,startDateTime
,endDateTime,page,itemsPerPage,granularity}"
    },
```

```
      {
        "rel": "health",
        "href": "https://localhost:8643/DcmConsole/rest/devices/83/health"
      },
      {
        "rel": "update",
        "href": "https://localhost:8643/DcmConsole/rest/devices/83"
      },
      {
        "rel": "delete",
        "href": "https://localhost:8643/DcmConsole/rest/devices/83"
      }
    ]
  }
```


8.52.6 Response field description

Refer to the chapter "get devices".

8.52.7 Errors

Except the generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The device id does not exist.

8.53 Get power of a device

8.53.1 Request URL

```
/rest/devices/{deviceId}/power
```

The parameter "deviceId" is the ID of the device.

8.53.2 HTTP request method

GET

8.53.3 Request example

https://localhost:8643/DcmConsole/rest/devices/1/power?latest=true,
https://localhost:8643/DcmConsole/rest/devices/1/power?latest=false&&startDateTime=2019-01-17 10:00%2B08:00&&endDateTime=2019-01-17 10:00%2B08:00&&page=0&&itemsPerPage=10&&granularity=monitoringPeriod

8.53.4 Request field description

Refer to the chapter "Get power of a data center".

8.53.5 Response example

Refer to the chapter "Get power of a data center".

8.53.6 Response field description

Refer to the chapter "Get power of a data center".

8.53.7 Errors

Refer to the chapter "Get power of a data center".

8.54 Get thermal of a device

8.54.1 Request URL

```
/rest/devices/{deviceId}/thermal
```

The parameter "deviceId" is the ID of the device.

8.54.2 HTTP request method

GET

8.54.3 Request example

https://localhost:8643/DcmConsole/rest/devices/1/thermal?latest=true,
https://localhost:8643/DcmConsole/rest/devices/1/thermal?latest=false&&startDateTime=2019-01-17 10:00%2B08:00&&endDateTime=2019-01-17 10:00%2B08:00&&page=0&&itemsPerPage=10&&granularity=monitoringPeriod

8.54.4 Request field description

Refer to the chapter "Get thermal of a data center".

8.54.5 Response example

Refer to the chapter "Get thermal of a data center".

8.54.6 Response field description

Refer to the chapter "Get thermal of a data center".

8.54.7 Errors

Refer to the chapter "Get thermal of a data center".

8.55 Get health status of a device

8.55.1 Request URL

```
/rest/devices/{deviceId}/health
```

The parameter "deviceId" is the ID of the device.

8.55.2 HTTP request method

GET

8.55.3 Request example

https://localhost:8643/DcmConsole/rest/devices/1/health

8.55.4 Request field description

N/A.

8.55.5 Response example


```
{
  "componentHealth": [
    {
      "component": "SYSTEM",
      "status": "NORMAL",
      "description": "OK"
    },
    {
      "component": "STORAGE",
      "status": "NORMAL",
      "description": ""
    },
    {
      "component": "BATTERY",
      "status": "NORMAL",
      "description": ""
    },
    {
      "component": "FAN",
      "status": "NORMAL",
```

```
      "description": ""
    },
    {
      "component": "VOLTAGE",
      "status": "NORMAL",
      "description": ""
    },
    {
      "component": "MEMORY",
      "status": "NORMAL",
      "description": ""
    },
    {
      "component": "TEMPERATURE",
      "status": "NORMAL",
      "description": ""
    },
    {
      "component": "PROCESSOR",
      "status": "NORMAL",
      "description": ""
    },
    {
      "component": "FPGA",
      "status": "UNKNOWN",
      "description": ""
    },
    {
      "component": "OPTANE_DIMM",
      "status": "UNKNOWN",
      "description": ""
    },
    {
      "component": "POWER_SUPPLY",
      "status": "NORMAL",
      "description": ""
    }
  ]
}
```


8.55.6 Response field description

Field	Type	Description
componentHealth	Array	Device component health data.
component	String	Device component. Possible values include but not limited: SYSTEM STORAGE BATTERY VOLTAGE PROCESSOR FPGA IO_MODULE MANAGEMENT_MODULE POWER_SUPPLY FAN MEMORY TEMPERATURE OPTANE_DIMM BLADE

status	String	Status of the component. Possible values: FAULT It means the component is unhealthy. ERROR It is deprecated and replaced by "FAULT". WARNING It is deprecated and replaced by "FAULT". NORMAL UNKNOWN
description		Description of the component health status.

8.55.7 Errors

Except the generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The device id does not exist.

8.56 Get inventory of a device

8.56.1 Request URL

```
/rest/devices/{deviceId}/inventory
```

The parameter "deviceId" is the ID of the device.

8.56.2 HTTP request method

GET

8.56.3 Request example

https://localhost:8643/DcmConsole/rest/devices/1/inventory

8.56.4 Request field description

N/A.

8.56.5 Response example


```
{
  "totalDeviceCount": 3,
  "supportedDeviceCount": 1,
  "details": {
    "resourceCategories": [
      {
        "categoryType": "PROCESSOR",
        "categoryName": "Processor",
        "categoryProperties": [
          {
            "categoryPropertyName": "ProcessorCount",
            "categoryPropertyValue": "4"
          }
        ],
        "categoryResources": [
          {
            "resourceName": "Processor 1",
            "resourceType": "Processor",
            "resourceProperties": [
              {
                "resourcePropertyName": "Instruction Set",
                "resourcePropertyValue": "x86-64"
```

```
                }
              ]
            }
          ]
        },
        {
          "categoryType": "MEMORY",
          "categoryName": "Memory",
          "categoryProperties": [
            {
              "categoryPropertyName": "MemoryCount",
              "categoryPropertyValue": "2"
            }
          ],
          "categoryResources": [
            {
              "resourceName": "DIMM 1",
              "resourceType": "Memory",
              "resourceProperties": [
                {
                  "resourcePropertyName": "DimmSizeMiB",
                  "resourcePropertyValue": "8192"
                }
              ]
            }
          ]
        }
      ]
    }
  }
```


8.56.6 Response field description

Field	Type	Description
totalDeviceCount	Integer	The number of devices. For a server, the value is 1. For chassis, the value is the number of the blades of the chassis.
supportedDeviceCount	Integer	The number of devices with the capability to get inventory data.
Details	Object	The detailed inventory data.
resourceCategories	Array	The resource categories.
categoryType	String	Category type. Possible values include but not limited to: STORAGE PROCESSOR MEMORY NIC
categoryName		Category name. Possible values include but not limited to: Storage Processor Memory
categoryProperties	Array	The properties of this category.
categoryPropertyName	String	The category property name which doesn't exist for chassis.
categoryPropertyValue	String	The category property value.
categoryResources	Array	The resources in the category which doesn't exist for chassis.

resourceName	String	The name of the resource.
resourceType	String	The resource type. Possible values include but not limited to: Storage Processor Memory
resourceProperties	Array	The property array of the resource.
resourcePropertyName	String	The resource property name.
resourcePropertyValue	String	The resource property value.

8.56.7 Errors

Except the generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The id does not exist.

8.57 Get real time SNMP walk data of a device

8.57.1 Request URL

/rest/devices/{deviceId}/realTimeSNMPWalkData

The parameter "deviceId" is the ID of the device.

8.57.2 HTTP request method

GET

8.57.3 Request example

https://localhost:8643/DcmConsole/rest/devices/1/realTimeSNMPWalkData?oid=1.3.1.111

8.57.4 Request field description

Field	Type	Attribute	Description
oid	String	Required	The target OID to walk.

8.57.5 Response example


```
{
  "links": [
    {
      "rel": "self",
      "href":
"https://localhost:8643/DcmConsole/rest/devices/1/realTimeSNMPWalkData?oid=1.3.1
.111"
    }
  ],
  "content": [
    {
      "oid": "1.3.1.111.1",
      "value": "1000",
      "unit": "W"
    },
    {
      "oid": "1.3.1.111.1.2",
      "value": "1000",
      "unit": "W"
    }
```

```
    ]
}
```

8.57.6 Response field description

Field	Type	Description
oid	String	The OID.
value	String	The value of the OID.
unit	String	The unit of the OID value.

8.57.7 Errors

Except the generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The device does not exist.
DEVICE_COMMUNICATION_FAILED	Fail to communicate with the device.
DEVICE_DOES_NOT_SUPPORT_SNMP	Not a SNMP device.
INVALID_OID	Invalid OID.

8.58 Get real time sensor data of a device

8.58.1 Request URL

```
/rest/devices/{deviceId}/realTimeSensorData
```

The parameter "deviceId" is the ID of the device.

8.58.2 HTTP request method

GET

8.58.3 Request example

https://localhost:8643/DcmConsole/rest/devices/1/realTimeSensorData

8.58.4 Request field description

N/A.

8.58.5 Response example

```
{
  "links": [
    {
      "rel": "self",
      "href":
"https://localhost:8643/DcmConsole/rest/devices/1/realTimeSensorData"
    }
  ],
  "content": [
    {
      "sensorEntityId": "3",
      "sensorName": "CPU",
      "sensorReading": "30",
      "unit": "degree C",
      "stateUnavailable": "false",
      "sensorType": "01",
      "readingType": "111",
      "message": "ok",
      "isAnalog": "false"
```

```
      }
    ]
  }
```

8.58.6 Response field description

Field	Type	Description
sensorEntityId	Integer	The sensor entity id defined in IPMI spec.
sensorName	String	The sensor name.
sensorReading	Float	The sensor value.
unit	String	The value unit defined in IPMI spec.
stateUnavailable	Boolean	If the value is available.
sensorType	Integer	The sensor type defined in IPMI spec.
readingType	Integer	The reading type defined in IPMI spec.
message	String	The description of the sensor status.
isAnalog	Boolean	If the sensor is analog.

Please refer to IPMI spec for field definitions.

8.58.7 Errors

Except the generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The device does not exist.
DEVICE_COMMUNICATION_FAILED	Fail to communicate with the device.
DEVICE_DOES_NOT_SUPPORT_CAP	The device does not support this capability.

8.59 Get real time UPS data of a device

8.59.1 Request URL

```
/rest/devices/{deviceId}/realTimeUPSData
```

The parameter "deviceId" is the ID of the device.

8.59.2 HTTP request method

GET

8.59.3 Request example

https://localhost:8643/DcmConsole/rest/devices/1/realTimeUPSData

8.59.4 Request field description

N/A.

8.59.5 Response example

```
{
  "links": [
    {
      "rel": "self",
      "href": "https://localhost:8643/DcmConsole/rest/devices/1/realTimeUPSData"
    }
  ],
  "content": [
    {
      "upsEstimatedChargeRemaining": "60",
      "upsSecondsOnBattery": "3600",
```

```
        "upsEstimatedMinutesRemaining": "60",
        "upsOutputFrequency": "50",
        "upsBypassFrequency": "50",
        "upsBatterytemperature": "30",
        "upsMaxLoad": "1",
        "inputTable": [
          {
            "inputIndex": "1",
            "inputPower": "500",
            "inputVoltage": "220",
            "inputCurrent": "1000",
            "inputFrequency": "50"
          }
        ],
        "outputTable": [
          {
            "outputIndex": "1",
            "outputPower": "500",
            "outputVoltage": "220",
            "outputCurrent": "1000",
            "outputPercentLoad": "50"
          }
        ],
        "bypassTable": [
          {
            "bypassIndex": "1",
            "bypassPower": "500",
            "bypassVoltage": "220",
            "bypassCurrent": "1000"
          }
        ]
      }
    ]
  }
```


8.59.6 Response field description

Field	Type	Description
upsEstimatedChargeRemaining	Integer	Estimation of remaining battery capacity (Units: percent).
upsSecondsOnBattery	Integer	If the unit is on battery power, the elapsed time since the UPS last switched to battery power, or the time since the network management subsystem last restarted, whichever is less, and zero shall be returned if the unit is not on battery power (Units: seconds).
upsEstimatedMinutesRemaining	Integer	Estimation of the time it takes to battery depletion under the present load conditions if the utility power is off and remains off, or if it were to be lost and remain off (Units: minutes).
upsOutputFrequency	Integer	The present output frequency (Units: 0.1 Hertz).
upsBypassFrequency	Integer	The present bypass frequency (Units: 0.1 Hertz).
upsBatterytemperature	Integer	The temperature of battery.
upsMaxLoad	Integer	
inputIndex	Integer	The input line identifier.
inputPower	Integer	The magnitude of the present input true power (Units: Watts).
inputVoltage	Integer	The magnitude of the present input voltage (Units: RMS Volts).
inputCurrent	Integer	The magnitude of the present input current (Units: 0.1 RMS Amp).

inputFrequency	Integer	The present input frequency (Units: 0.1 Hertz).
outputIndex	Integer	The output line identifier.
outputPower	Integer	The present output true power (Units: Watts).
outputVoltage	Integer	The present output voltage (Units: RMS Volts).
outputCurrent	Integer	The present output current (Units: 0.1 RMS Amp).
outputPercentLoad	Integer	The percentage of the UPS power capacity being used on this output line, i.e., the greater of the percent load of true power capacity and the percent load of VA (Units: percent).
bypassIndex	Integer	The bypass line identifier.
bypassPower	Integer	The present bypass true power (Units: Watts).
bypassVoltage	Integer	The present bypass voltage (Units: RMS Volts).
bypassCurrent	Integer	The present bypass current (Units: 0.1 RMS Amp).

8.59.7 Errors

Except the generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The device does not exist.
DEVICE_COMMUNICATION_FAILED	Fail to communicate with the device.
DEVICE_DOES_NOT_SUPPORT_CAP	The device does not support this capability.

8.60 Get real time PDU data of a device

8.60.1 Request URL

/rest/devices/{deviceId}/realTimePDUData

The parameter "deviceId" is the ID of the device.

8.60.2 HTTP request method

GET

8.60.3 Request example

https://localhost:8643/DcmConsole/rest/devices/1/realTimePDUData

8.60.4 Request field description

N/A.

8.60.5 Response example


```
{
  "links": [
    {
      "rel": "self",
      "href": "https://localhost:8643/DcmConsole/rest/devices/1/realTimePDUData"
    }
  ],
  "content": [
    {
      "power": "6000",
      "outletNumber": "10",
      "powerUnitDivisor": "60",
      "phaseData": [
        {
          "phaseLoad": "50",
          "phaseVoltage": "220",
```

```
            "phaseLoadUnitDivisor": "10",
            "phaseVoltageUnitDivisor": "10"
          }
        ],
        "outletData": [
          {
            "outletIndex": "1",
            "outletLoad": "500",
            "outletPower": "220",
            "outletVoltage": "2080",
            "outletState": "ok",
            "outletPowerUnitDivisor": "10",
            "outletLoadUnitDivisor": "10",
            "outletVoltageUnitDivisor": "10"
          }
        ],
        "temperatureSensorReading": [
          {
            "sensorValue": "100",
            "sensorUnitDivisor": "10",
            "sensorName": "temp 1",
            "SensorSerialNumber": "ABC"
          }
        ],
        "humiditySensorReading": [
          {
            "sensorValue": "100",
            "sensorUnitDivisor": "10",
            "sensorName": "temp 1",
            "SensorSerialNumber": "ABC"
          }
        ]
      }
    ]
  }
```


8.60.6 Response field description

Field	Type	Description
power	Integer	Total power consumption.
outletNumber	Integer	Outlet count.
powerUnitDivisor	Integer	Power unit divider, which requires you to translate the value, e.g., power value is 650, powerUnitDivider = 10, the actual power is 65W.
phaseLoad	Integer	Phase level load.
phaseVoltage	Integer	Phase level voltage.
phaseLoadUnitDivisor	Integer	Load unit divider which requires you to translate the value, e.g., load value is 5, loadUnitDivider = 10, the actual load is 0.5 AMP.
phaseVoltageUnitDivisor	Integer	Voltage unit divider, you need to translate the value, e.g., voltage value is 2080, voltageUnitDivider = 10, the actual voltage is 208 V.
outletIndex	Integer	Outlet index.
outletLoad	Integer	Outlet load.
outletPower	Integer	Outlet power.
outletVoltage	Integer	Outlet voltage.
outletState	String	Outlet state.

outletPowerUnitDivisor	Integer	Load unit divider which requires you to translate the value, e.g., load value is 5, loadUnitDivider = 10, the actual load is 0.5 AMP.
outletLoadUnitDivisor	Integer	Power unit divider which requires you to translate the value, e.g., power value is 650, powerUnitDivider = 10, the actual power is 65 W.
outletVoltageUnitDivisor	Integer	Voltage unit divider which requires you to translate the value, e.g., voltage value is 2080, voltageUnitDivider = 10, the actual voltage is 208 V.
sensorValue	Integer	Sensor data value.
sensorUnitDivisor	Integer	Sensor data unit divider which requires you to translate the value, e.g., sensor value is 2080, sensorUnitDivider = 10, the actual voltage is 208.
sensorName	String	Sensor name.
SensorSerialNumber	Integer	Sensor serial number.

8.60.7 Errors

Except the generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The device does not exist.
DEVICE_COMMUNICATION_FAILED	Fail to communicate with the device.
DEVICE_DOES_NOT_SUPPORT_CAP	The device does not support this capability.

8.61 Get real time health data of a device

8.61.1 Request URL

```
/rest/devices/{deviceId}/realTimeHealthData
```

The parameter "deviceId" is the ID of the device.

8.61.2 HTTP request method

GET

8.61.3 Request example

https://localhost:8643/DcmConsole/rest/devices/1/realTimeHealthData

8.61.4 Request field description

N/A.

8.61.5 Response example


```
{
  "links": [
    {
      "rel": "self",
      "href":
"https://localhost:8643/DcmConsole/rest/devices/1/realTimeHealthData"
    }
  ],
  "content": [
    {
      "component": "SYSTEM",
      "status": "NORMAL",
      "description": "OK"
    },
    {
      "component": "STORAGE",
```

```
      "status": "NORMAL",
      "description": ""
    },
    {
      "component": "BATTERY",
      "status": "NORMAL",
      "description": ""
    },
    {
      "component": "FAN",
      "status": "NORMAL",
      "description": ""
    },
    {
      "component": "VOLTAGE",
      "status": "NORMAL",
      "description": ""
    },
    {
      "component": "MEMORY",
      "status": "NORMAL",
      "description": ""
    },
    {
      "component": "TEMPERATURE",
      "status": "NORMAL",
      "description": ""
    },
    {
      "component": "PROCESSOR",
      "status": "NORMAL",
      "description": ""
    },
    {
      "component": "FPGA",
      "status": "UNKNOWN",
      "description": ""
    },
    {
      "component": "OPTANE_DIMM",
      "status": "UNKNOWN",
      "description": ""
    },
    {
      "component": "POWER_SUPPLY",
      "status": "NORMAL",
      "description": ""
    }
  ]
}
```


8.61.6 Response field description

Refer to the chapter "Get health of a device".

8.61.7 Errors

Except the generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The device does not exist.
DEVICE_COMMUNICATION_FAILED	Fail to communicate with the device.

DEVICE_DOES_NOT_SUPPORT_CAP	The device does not support this capability.

***Note**: If the health data is pushed by external systems via calling the REST API "Push health data to a device", the API may return the health data from external systems.*

8.62 Get real time component faults of a device

8.62.1 Request URL

```
/rest/devices/{deviceId}/realTimeComponentFaults
```

The parameter "deviceId" is the ID of the device.

8.62.2 HTTP request method

GET

8.62.3 Request example

https://localhost:8643/DcmConsole/rest/devices/1/realTimeComponentFaults

8.62.4 Request field description

N/A.

8.62.5 Response example

```
{
  "links": [
    {
      "rel": "self",
      "href":
"https://localhost:8643/DcmConsole/rest/devices/1/realTimeComponentFaults"
    }
  ],
  "content": [
    {
      "component": "FAN",
      "description": "Fan is not ok"
    }
  ]
}
```

8.62.6 Response field description

Refer to the chapter "Get health of a device".

8.62.7 Errors

Except the generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The device does not exist.
DEVICE_COMMUNICATION_FAILED	Fail to communicate with the device.
DEVICE_DOES_NOT_SUPPORT_CAP	The device does not support this capability.

***Note**: If the health data is pushed by external systems via calling the REST API "Push health data to a device", the API may return the health data from external systems.*

8.63 Get SSD smart data of a device

8.63.1 Request URL

```
/rest/devices/{deviceId}/SSDSmartData
```

The parameter "deviceId" is the ID of the device.

8.63.2 HTTP request method

GET

8.63.3 Request example

https://localhost:8643/DcmConsole/rest/devices/1/SSDSmartData

8.63.4 Request field description

N/A.

8.63.5 Response example


```
{
  "links": [
    {
      "rel": "self",
      "href": "https://localhost:8643/DcmConsole/rest/devices/1/SSDSmartData"
    }
  ],
  "content": [
    {
      "model": "INTEL SSDMCEAC120A3",
      "serialNumber": "CVLI315201J8120B",
      "firmwareVersion": "LLGi",
      "userCapacity": "120,034,123,776 bytes [120 GB]",
      "attributes": [
        {
          "attributeId": 1,
          "attributeName": "Power_On_Hours",
          "attributeRawValue": "1000",
          "attributeValue": "100",
          "attributeThreshold": "100"
        },
        {
          "attributeId": 2,
          "attributeName": "Total_LBAs_Written",
          "attributeRawValue": "1000",
          "attributeValue": "100",
          "attributeThreshold": "100"
        }
      ]
    }
  ]
}
```


8.63.6 Response field description

Field	Type	Description
model	String	The model of the SSD.
serialNumber	String	The serial number of the SSD.
firmwareVersion	String	The firmware version of the SSD.
userCapacity	String	The user capacity of the SSD.
attributes	Array	The smart attributes.
attributeId	Integer	The smart attribute ID.
attributeName	String	The smart attribute name.
attributeValue	Integer	The smart attribute value.
attributeRawValue	Integer	The smart attribute raw value.

attributeThreshold	Integer	The smart attribute threshold.

8.63.7 Errors

Except the generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The device does not exist.
DEVICE_COMMUNICATION_FAILED	Fail to communicate with the device.
DEVICE_DOES_NOT_SUPPORT_SMART	The device does not support to get smart data from SSDs.

8.64 Get all outlets of a PDU

8.64.1 Request URL

/rest/devices/{deviceId}/outlets

"deviceId" is the id of the PDU.

8.64.2 HTTP request method

GET

8.64.3 Request example

https://localhost:8643/DcmConsole/rest/devices/1/outlets

8.64.4 Request field description

N/A.

8.64.5 Response example


```
{
  "links": [
    {
      "rel": "self",
      "href": "https://localhost:8643/DcmConsole/rest/devices/1/outlets"
    }
  ],
  "content": [
    {
      "outletId": "1",
      "associatedDeviceId": "2",
      "status": "ON",
      "links": [
        {
          "rel": "self",
          "href": "https://localhost:8643/DcmConsole/rest/devices/1/outlets/1"
        },
        {
          "rel": "update",
          "href": "https://localhost:8643/DcmConsole/rest/devices/1/outlets/1"
        }
      ]
    },
    {
      "outletId": "2",
      "status": "ON",
      "associatedDeviceId": "3",
      "links": [
        {
          "rel": "self",
          "href": "https://localhost:8643/DcmConsole/rest/devices/1/outlets/2"
```

```
        },
        {
          "rel": "update",
          "href": "https://localhost:8643/DcmConsole/rest/devices/1/outlets/2"
        }
      ]
    }
  ]
}
```

8.64.6 Response field description

Field	Type	Description
outletId	Integer	The id of the outlet.
associatedDeviceId	Integer	The id of the device associated to the outlet.
status	String	The on/off status of the outlet. Possible values: ON OFF UNKNOWN

8.64.7 Errors

Except the generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The device does not exist.
DEVICE_DOES_NOT_SUPPORT_CAP	The device does not have the capability. Typically, it is not a PDU.
DEVICE_COMMUNICATION_FAILED	Fail to communicate with the device.

8.65 Get an outlet of a PDU

8.65.1 Request URL

```
/rest/devices/{deviceId}/outlets/{outletId}
```

"deviceId" is the id of the PDU. "outletId" is the id of the outlet.

8.65.2 HTTP request method

GET

8.65.3 Request example

https://localhost:8643/DcmConsole/rest/devices/1/outlets/1

8.65.4 Request field description

N/A.

8.65.5 Response example

```
{
  "outletId": "1",
  "associatedDeviceId": "2",
  "status": "ON",
  "links": [
    {
      "rel": "self",
      "href": "https://localhost:8643/DcmConsole/rest/devices/1/outlets/1"
    },
    {
      "rel": "update",
      "href": "https://localhost:8643/DcmConsole/rest/devices/1/outlets/1"
```

```
        }
    ]
}
```

8.65.6 Response field description

Refer to the chapter "Get all outlets of a PDU".

8.65.7 Errors

Except the generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The device id or outlet id does not exist.
DEVICE_DOES_NOT_SUPPORT_CAP	The device does not have the capability. Typically, it is not a PDU.
DEVICE_COMMUNICATION_FAILED	Fail to communicate with the device.

8.66 Update an outlet of a PDU

8.66.1 Request URL

```
/rest/devices/{deviceId}/outlets/{outletId}
```

"deviceId" is the id of the PDU. "outletId" is the id of the outlet.

8.66.2 HTTP request method

PUT

8.66.3 Request example

```
{
  "associatedDeviceId": "3"
}
```

8.66.4 Request field description

The client can specify one or some of these fields to update.

Field	Type	Attribute	Description
associatedDeviceId	Integer	Optional	The id of the device to be associated. The value "-1" means to disassociate devices from the outlet. An outlet can only be associated with one device which means the old associated device id will be overwritten by the new one.
status	String	Optional	The on/off status of the outlet. Possible values: ON OFF Updating the status will power on/off the PDU outlet.

8.66.5 Response example

Refer to the chapter "Get an outlet of a PDU".

8.66.6 Response field description

Refer to the chapter "Get all outlets of a PDU".

8.66.7 Errors

Except the generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The PDU device id or outlet id does not exist.
DEVICE_DOES_NOT_SUPPORT_CAP	The device does not have the capability. Typically, it is not a PDU, or the outlet does not support power on/off operation.
INVALID_ASSOCIATED_DEVICE_ID	The associated device id does not exist.
DEVICE_COMMUNICATION_FAILED	Fail to communicate with the device.

8.67 Update a device

8.67.1 Request URL

/rest/devices/{id}

"id" is the id of the target device.

8.67.2 HTTP request method

PUT

8.67.3 Request example

```
{
  "parentId": 1,
  "name": "device1",
  "address": "192.168.1.1",
  "username": "user1",
  "password": "password1",
  "deratedPower": 400,
  "size": 1,
  "location": 1,
  "description": "device1",
  "birthday": "2019-01-17"
}
```

8.67.4 Request field description

The client can update one or more than one fields.

Field	Type	Attribute	Description
parentId	Integer	Optional	The id of the parent rack or chassis. For blade, it means the ID of the parent chassis. If the parent id is different from the current parent, the device will be moved to the new parent.
name	String	Optional	The name of the device.
address	String	Optional	The IP address or host name of the device. It is useless when the type is unmanaged device.
userName	String	Optional	The user name to connect to the device.
password	String	Optional	The password to connect to the device.
key	String	Optional	The BMC key which is useful only when the type is IPMI.
deratedPower	Integer	Optional	The derated power for none-chassis devices such as servers, PDU, UPS and so on. The unit is W.
namePlatePower	Integer	Optional	The nameplate power of the device.
snmpCommunityString	String	Optional	SNMP community string. It is useful only when the protocol is SNMPV1v2c.
snmpEncryptionPassword	String	Optional	SNMP encryption password. It is useful only when the protocol is SNMPv3.
snmpAuthenticationProtocol	String	Optional	SNMP authentication protocol. It is useful only when the protocol is SNMPv3. Possible values:

			AuthMD5 AuthSHA1 AuthHMAC128SHA224 AuthHMAC192SHA256 AuthHMAC256SHA384 AuthHMAC384SHA512
snmpEncryptionProtocol	String	Optional	SNMP encryption protocol. It is useful only when the type is SNMPv3. Possible values: PrivDES Priv3DES PrivAES128 PrivAES192 PrivAES256
httpsPort	Integer	Optional	The HTTPs port. The default value is 443.
sshPort	Integer	Optional	The SSH port. The default value is 22.
authenticateEntity	Boolean	Optional	Whether it is need to validate the device. For https, the device certificate needs to be imported into DCM key store if the field value is true.
size	Integer	Optional	The size of the device. For servers, the default value is 1. For chassis, the default value is 10. The unit is U.
location	Integer	Optional	The slot of the device.
typicalPower	Integer	Optional	The typical power of the device. It is useful only when the type is unmanaged device.
idlePower	Integer	Optional	The idle power of the device. It is useful only when the type is unmanaged device.
peakPower	Integer	Optional	The peak power of the device. It is useful only when the type is unmanaged device.
model	String	Optional	The device model of the device. It is useful only when the type is unmanaged device.
osType	String	Optional	The in-band OS type. The possible values: Windows Linux Xen ESX
osAddress	String	Optional	The OS IP address or host name. It is used to specify the in-band OS information when the protocol is IPMI.
osUserName	String	Optional	The user name to access the OS. It is used to specify the in-band OS information when the protocol is IPMI.
osPassword	String	Optional	The user password to access the OS. It is used to specify the in-band OS information when the protocol is IPMI.
osSshPort	Integer	Optional	The in-band OS SSH port with the default value 22. It is used to specify the in-band OS information when the protocol is IPMI.
description	String	Optional	Description for the device.
reportInletTemp	Boolean	Optional	Whether to report inlet temperature. Currently it is only useful for chatsworth PDU.
birthday	Date	Optional	The birthday of the device.
customConsoleURL	String	Optional	The specified device management console URL.
width	Integer	Optional	The horizontal grid count of a chassis or a blade. It is used to locate blades in a chassis.
height	Integer	Optional	The vertical grid count of a chassis or a blade. It is used to locate blades in a chassis.

startX	Integer	Optional	The horizontal position of a blade in a chassis.
startY	Integer	Optional	The vertical position of a blade in a chassis.
epr	String	Optional	The action of a device with emergency power reduction. Possible values: Minimize Power Consumption Shutdown No Action
customAttributes	Array	Optional	Refer to the chapter "Add a device".
customAttributeName	String	Optional	Refer to the chapter "Add a device".
customAttributeValue	String	Optional	Refer to the chapter "Add a device".
offline	Boolean	Optional	Whether the device is offline.
provisioningUser	String	Optional	The provisioning user of the device. The provisioning user must be an administrator. The value "All" means resetting provisioning user so that all administrators can do provisioning.

8.67.5 Response example


```
{
  "parentId": 60,
  "id": 83,
  "protocol": "IPMI",
  "deviceType": "Server",
  "name": "rest-server-2",
  "address": "sh-mke",
  "deratedPower": 600,
  "size": 1,
  "model": "Node Manager 1.5 Intel - ProLiant DL380 G7",
  "customConsoleURL": "https://10.239.157.21:443",
  "epr": "Minimize Power Consumption",
  "links": [
    {
      "rel": "self",
      "href": "https://localhost:8643/DcmConsole/rest/devices/83"
    },
    {
      "rel": "power",
      "href":
"https://localhost:8643/DcmConsole/rest/devices/83/power{?latest,startDateTime,e
ndDateTime,page,itemsPerPage,granularity}"
    },
    {
      "rel": "thermal",
      "href":
"https://localhost:8643/DcmConsole/rest/devices/83/thermal{?latest,startDateTime
,endDateTime,page,itemsPerPage,granularity}"
    },
    {
      "rel": "health",
      "href": "https://localhost:8643/DcmConsole/rest/devices/83/health"
    },
    {
      "rel": "update",
      "href": "https://localhost:8643/DcmConsole/rest/devices/83"
    },
    {
      "rel": "delete",
      "href": "https://localhost:8643/DcmConsole/rest/devices/83"
    }
```

```
    ]
}
```

8.67.6 Response field description

Refer to the chapter "Get devices".

8.67.7 Errors

Except the generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The parent id or the device id does not exist.
INVALID_PARENT_ID	The parent id is not an id of a rack or chassis. A chassis cannot be added to a chassis, so this error will also be reported in this case.
PROTOCOL_IS_NOT_UPDATE	Protocol cannot be updated.
INVALID_NAME	Invalid name.
INVALID_PROTOCOL	Invalid protocol.
INVALID_DERATED_POWER	Invalid derated power.
INVALID_NAMEPLATE_POWER	Invalid nameplate power.
INVALID_HTTPS_PORT	Invalid HTTPS port.
INVALID_SSH_PORT	Invalid SSH port.
INVALID_SIZE	Invalid device size.
INVALID_LOCATION	Invalid location.
INVALID_TYPICAL_POWER	Invalid typical power.
INVALID_IDLE_POWER	Invalid idle power.
INVALID_PEAK_POWER	Invalid peak power.
INVALID_OS_TYPE	Invalid OS type.
INVALID_WIDTH	Invalid width.
INVALID_HEIGHT	Invalid height.
INVALID_START_X	Invalid start X.
INVALID_START_Y	Invalid start Y.
INVALID_EPR	Invalid EPR action.
INVALID_PARAMETER	Invalid parameter.
PLATFORM_ID_CANNOT_BE_UPDATED	Platform ID cannot be updated. Platform ID is a write- once property and can only be specified when adding the device.
DEVICE_NUMBER_LIMIT	Number of the device exceeds the license limit.
CUSTOM_ATTR_NAME_IS_REQUIRED	Custom attribute name is required.
INVALID_CUSTOM_ATTR_VALUE	The custom attribute value is invalid. Typically, the time format of a custom value is not "yyyy-MM-dd" for date type custom attribute.
NO_PRIVILEDGE	No privilege to update provisioning user. Only build-in administrator can update it.

INVALID_PROVISIONING_USER_VALUE	Invalid provisioning user.

8.68 Control a device

8.68.1 Request URL

```
/rest/devices/{deviceId}/control
```

The parameter "deviceId" is the ID of the device.

8.68.2 HTTP request method

POST

8.68.3 Request example

```
{
  "controlAction": "POWERON"
}
```

8.68.4 Request field description

Field	Type	Attribute	Description
controlAction	String	Required	Possible values: POWERON Power on the device. POWEROFF Power off the device. If the device supports soft power-off, DCM will send soft power off command to it. Or else DCM will check if the device supports hard power-off. If yes, DCM will send hard power-off command to the device. SOFTPOWEROFF Perform soft power off. HARDPOWEROFF Perform hard power off. RECONNECT Reconnect to the device. ENABLE_POWER_MONITORING_CAP DISABLE_POWER_MONITORING_CAP Http status code 202 will be returned by this API if the action is OK.

8.68.5 Response example

Refer to the chapter "Add a device".

8.68.6 Response field description

Refer to the chapter "Add a device".

8.68.7 Errors

Except the generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The task Id does not exist.
CONTROL_ACTION_IS REQUIRED	Control action is required.

INVALID_CONTROL_ACTION	Invalid control action value.
ACTION_IS_NOT_SUPPORT	The device has no capability. For example, try to power on a device without "power on" capability or re-connect an unmanaged device in DCM.

***Note**:*

As some action results depend on the device (e.g., send a power-on command to device), http status code 202 will be returned for these kinds of actions. The client can query the device information after an interval to get the latest status.

8.69 Delete a device

8.69.1 Request URL

```
/rest/devices/{id}
```

"id" is the id of the target device.

8.69.2 HTTP request method

DELETE

8.69.3 Request example

https://localhost:8643/DcmConsole/rest/devices/1

8.69.4 Request field description

N/A.

8.69.5 Response example

N/A.

8.69.6 Response field description

N/A.

8.69.7 Errors

Except the generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The device id does not exist.

***Note**: All devices of the chassis will be deleted if the device is a chassis.*

8.70 Get blades of a chassis

8.70.1 Request URL

```
/rest/devices?parentId={chassisId}
```

The parameter "chassisId" is the ID of the chassis.

8.70.2 HTTP request method

GET

8.70.3 Request example

https://localhost:8643/DcmConsole/rest/devices?parentId=1

8.70.4 Request field description

N/A.

8.70.5 Response example

Refer to the chapter "Get devices".

8.70.6 Response field description

Refer to the chapter "Get devices".

8.70.7 Errors

Refer to the chapter "Get devices".

8.71 Push health data to a device

8.71.1 Request URL

```
/rest/devices/{deviceId}/health
```

The parameter "deviceId" is the ID of the device.

8.71.2 HTTP request method

POST

8.71.3 Request example

```
{
    "componentHealth": [
        {
            "component": "SYSTEM",
            "status": "NORMAL",
            "description": "OK"
        },
        {
            "component": "STORAGE",
            "status": "NORMAL",
            "description": ""
        }
],
"forceOverwrite": "false"
}
```

8.71.4 Request field description

Field	Type	Attribute	Description
componentHealth	Array	Mandatory	Device component health data.
Component	String	Mandatory	Device component. Possible values include but not limited: SYSTEM STORAGE BATTERY VOLTAGE PROCESSOR FPGA IO_MODULE MANAGEMENT_MODULE POWER_SUPPLY FAN MEMORY TEMPERATURE OPTANE_DIMM BLADE The component name is case insensitive.
status	String	Mandatory	Status of the component. Possible values:

			FAULT It means the component is unhealthy. NORMAL UNKNOWN
Description		Optional	Description information of the component health.
forceOverwrite	Boolean	Optional	The latest health data retrieved from other channels will be replaced by this data if the value is true for the same components. If DCM cannot get health data from other channels for some components, the health data will always be used in DCM with the default value false. If this value is true, DCM will always use this data even if DCM can get new health data of the same components from other channels. The external system should push data periodically. The external system should send request to set the value to be false if the user wants to disable this behavior.

8.71.5 Response example

N/A.

8.71.6 Response field description

N/A.

8.71.7 Errors

Except the generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The device id does not exist.
COMPONENT_NAME_IS_REQUIRED	Component name is required.
COMPONENT_STATUS_IS_REQUIRED	Component status is required.
COMPONENT_STATUS_IS_INVALID	Invalid component status.

8.72 Add a group

8.72.1 Request URL

```
/rest/groups
```

8.72.2 HTTP request method

POST

8.72.3 Request example

```
{
  "name": "group1",
  "description": "description"
}
```

8.72.4 Request field description

Field	Type	Attribute	Description
name	String	Required	The name of the group.
description	String	Optional	Description for the group.

8.72.5 Response example


```
{
  "id": 12,
  "name": "aa",
  "description": "",
  "links": [
    {
      "rel": "self",
      "href": "https://localhost:8643/DcmConsole/rest/groups/12"
    },
    {
      "rel": "power",
      "href":
"https://localhost:8643/DcmConsole/rest/groups/12/power{?latest,startDateTime,en
dDateTime,page,itemsPerPage,granularity}"
    },
    {
      "rel": "thermal",
      "href":
"https://localhost:8643/DcmConsole/rest/groups/12/thermal{?latest,startDateTime,
endDateTime,page,itemsPerPage,granularity}"
    },
    {
      "rel": "health",
      "href": "https://localhost:8643/DcmConsole/rest/groups/12/health"
    },
    {
      "rel": "devices",
      "href": "https://localhost:8643/DcmConsole/rest/groups/12/devices"
    },
    {
      "rel": "update",
      "href": "https://localhost:8643/DcmConsole/rest/groups/12"
    },
    {
      "rel": "delete",
      "href": "https://localhost:8643/DcmConsole/rest/groups/12"
    }
  ]
}
```


8.72.6 Response field description

Field	Type	Description
id	Integer	The id of the group.

Refer to the Request field description.

8.72.7 Errors

Except the generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
NAME_IS_REQUIRED	The name field is required, or the value is missing.
INVALID_NAME	The name is invalid.

8.73 Get groups

8.73.1 Request URL

```
/rest/groups
```

8.73.2 HTTP request method

GET

8.73.3 Request example

https://localhost:8643/DcmConsole/rest/groups

8.73.4 Request field description

N/A.

8.73.5 Response example


```
{
  "links": [
    {
      "rel": "self",
      "href": "https://localhost:8643/DcmConsole/rest/groups"
    },
    {
      "rel": "add",
      "href": "https://localhost:8643/DcmConsole/rest/groups"
    }
  ],
  "content": [
    {
      "id": 12,
      "name": "aa",
      "links": [
        {
          "rel": "self",
          "href": "https://localhost:8643/DcmConsole/rest/groups/12"
        },
        {
          "rel": "power",
          "href":
"https://localhost:8643/DcmConsole/rest/groups/12/power{?latest,startDateTime,en
dDateTime,page,itemsPerPage,granularity}"
        },
        {
          "rel": "thermal",
          "href":
"https://localhost:8643/DcmConsole/rest/groups/12/thermal{?latest,startDateTime,
endDateTime,page,itemsPerPage,granularity}"
        },
        {
          "rel": "health",
          "href": "https://localhost:8643/DcmConsole/rest/groups/12/health"
        },
        {
          "rel": "devices",
          "href": "https://localhost:8643/DcmConsole/rest/groups/12/devices"
        },
        {
          "rel": "update",
          "href": "https://localhost:8643/DcmConsole/rest/groups/12"
        },
        {
          "rel": "delete",
          "href": "https://localhost:8643/DcmConsole/rest/groups/12"
        }
      ]
    }
  ]
```

```
}
```

8.73.6 Response field description

Refer to the chaper "Add a group".

8.73.7 Errors

Generic errors defined in the chapter "Error handling".

8.74 Get a group

8.74.1 Request URL

```
/rest/groups/{groupId}
```

The parameter "groupId" is the ID of the group.

8.74.2 HTTP request method

GET

8.74.3 Request example

https://localhost:8643/DcmConsole/rest/groups/12

8.74.4 Request field description

N/A.

8.74.5 Response example

```
{
  "id": 12,
  "name": "aa",
  "description": "",
  "links": [
    {
      "rel": "self",
      "href": "https://localhost:8643/DcmConsole/rest/groups/12"
    },
    {
      "rel": "power",
      "href":
"https://localhost:8643/DcmConsole/rest/groups/12/power{?latest,startDateTime,en
dDateTime,page,itemsPerPage,granularity}"
    },
    {
      "rel": "thermal",
      "href":
"https://localhost:8643/DcmConsole/rest/groups/12/thermal{?latest,startDateTime,
endDateTime,page,itemsPerPage,granularity}"
    },
    {
      "rel": "health",
      "href": "https://localhost:8643/DcmConsole/rest/groups/12/health"
    },
    {
      "rel": "devices",
      "href": "https://localhost:8643/DcmConsole/rest/groups/12/devices"
    },
    {
      "rel": "update",
      "href": "https://localhost:8643/DcmConsole/rest/groups/12"
    },
    {
      "rel": "delete",
```

```
        "href": "https://localhost:8643/DcmConsole/rest/groups/12"
      }
    ]
  }
```


8.74.6 Response field description

Refer to the chapter "Get groups".

8.74.7 Errors

Except the generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The group does not exist.

8.75 Get power of a group

8.75.1 Request URL

/rest/groups/{groupId}/power

The parameter "groupId" is the ID of the group.

8.75.2 HTTP request method

GET

8.75.3 Request example

https://localhost:8643/DcmConsole/rest/groups/1/power?latest=true,
https://localhost:8643/DcmConsole/rest/groups/1/power?latest=false&&startDateTime=2019-01-17 10:00%2B08:00&&endDateTime=2019-01-17 10:00%2B08:00&&page=0&&itemsPerPage=10&&granularity=monitoringPeriod

8.75.4 Request field description

Refer to the chapter "Get power of data center".

8.75.5 Response example

Refer to the chapter "Get power of data center".

8.75.6 Response field description

Refer to the chapter "Get power of data center".

8.75.7 Errors

Refer to the chapter "Get power of data center".

8.76 Get thermal of a group

8.76.1 Request URL

/rest/groups/{groupId}/thermal

The parameter "groupId" is the ID of the group.

8.76.2 HTTP request method

GET

8.76.3 Request example

https://localhost:8643/DcmConsole/rest/groups/1/thermal?latest=true,

https://localhost:8643/DcmConsole/rest/groups/1/thermal?latest=false&&startDateTime=2019-01-17 10:00%2B08:00&&endDateTime=2019-01-17 10:00%2B08:00&&page=0&&itemsPerPage=10&&granularity=monitoringPeriod

8.76.4 Request field description

Refer to the chapter "Get thermal of data center".

8.76.5 Response example

Refer to the chapter "Get thermal of data center".

8.76.6 Response field description

Refer to the chapter "Get thermal of data center".

8.76.7 Errors

Refer to the chapter "Get thermal of data center".

8.77 Get health of a group

8.77.1 Request URL

/rest/groups/{groupId}/health

The parameter "groupId" is the ID of the group.

8.77.2 HTTP request method

GET

8.77.3 Request example

https://localhost:8643/DcmConsole/rest/groups/1/health

8.77.4 Request field description

N/A.

8.77.5 Response example

Refer to the chapter "Get health of data center".

8.77.6 Response field description

Refer to the chapter "Get health of data center".

8.77.7 Errors

Except the generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The group id does not exist.

8.78 Get inventory of a group

8.78.1 Request URL

/rest/groups/{groupId}/inventory

The parameter "groupId" is the ID of the group.

8.78.2 HTTP request method

GET

8.78.3 Request example

https://localhost:8643/DcmConsole/rest/groups/1/inventory

8.78.4 Request field description

N/A.

8.78.5 Response example

Refer to the chapter "Get inventory of a data center".

8.78.6 Response field description

Refer to the chapter "Get inventory of a data center".

8.78.7 Errors

Except the generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The id does not exist.

8.79 Get devices of a group

8.79.1 Request URL

```
/rest/groups/{groupId}/devices
```

The parameter "groupId" is the ID of the group.

8.79.2 HTTP request method

GET

8.79.3 Request example

https://localhost:8643/DcmConsole/rest/groups/1/devices

8.79.4 Request field description

N/A.

8.79.5 Response example


```
{
  "links": [
    {
      "rel": "self",
      "href": "https://localhost:8643/DcmConsole/rest/groups/12/devices"
    },
    {
      "rel": "add",
      "href": "https://localhost:8643/DcmConsole/rest/groups/12/devices"
    }
  ],
  "content": [
    {
      "parentId": 50,
      "id": 53,
      "protocol": "Unmanaged server",
      "deviceType": "Unmanaged server",
      "name": "aa",
      "description": "中文",
      "deratedPower": 400,
      "size": 5,
      "model": "",
      "birthday": "2019-01-18T14:32:35+08:00",
      "links": [
        {
          "rel": "self",
          "href": "https://localhost:8643/DcmConsole/rest/devices/53"
```

```
        },
        {
          "rel": "power",
          "href":
"https://localhost:8643/DcmConsole/rest/devices/53/power{?latest,startDateTime,e
ndDateTime,page,itemsPerPage,granularity}"
        },
        {
          "rel": "thermal",
          "href":
"https://localhost:8643/DcmConsole/rest/devices/53/thermal{?latest,startDateTime
,endDateTime,page,itemsPerPage,granularity}"
        },
        {
          "rel": "health",
          "href": "https://localhost:8643/DcmConsole/rest/devices/53/health"
        },
        {
          "rel": "update",
          "href": "https://localhost:8643/DcmConsole/rest/devices/53"
        },
        {
          "rel": "delete",
          "href": "https://localhost:8643/DcmConsole/rest/devices/53"
        }
      ]
    }
  ]
}
```


8.79.6 Response field description

Refer to the chapter "Get devices".

8.80 Update a group

8.80.1 Request URL

```
/rest/groups/{id}
```

"id" is the id of the target group.

8.80.2 HTTP request method

PUT

8.80.3 Request example


```
{
  "name": "group1",
  "description": "description"
}
```


8.80.4 Request field description

The client can update one or more than one fields.

Field	Type	Attribute	Description
name	String	Optional	The name of the group.
description	String	Optional	Description for the group.

8.80.5 Response example


```
{
  "id": 12,
  "name": " group1",
```

```
  "description": "description",
  "links": [
    {
      "rel": "self",
      "href": "https://localhost:8643/DcmConsole/rest/groups/12"
    },
    {
      "rel": "power",
      "href":
"https://localhost:8643/DcmConsole/rest/groups/12/power{?latest,startDateTime,en
dDateTime,page,itemsPerPage,granularity}"
    },
    {
      "rel": "thermal",
      "href":
"https://localhost:8643/DcmConsole/rest/groups/12/thermal{?latest,startDateTime,
endDateTime,page,itemsPerPage,granularity}"
    },
    {
      "rel": "health",
      "href": "https://localhost:8643/DcmConsole/rest/groups/12/health"
    },
    {
      "rel": "devices",
      "href": "https://localhost:8643/DcmConsole/rest/groups/12/devices"
    },
    {
      "rel": "update",
      "href": "https://localhost:8643/DcmConsole/rest/groups/12"
    },
    {
      "rel": "delete",
      "href": "https://localhost:8643/DcmConsole/rest/groups/12"
    }
  ]
}
```


8.80.6 Response field description

Refer to the chapter "Get a group".

8.80.7 Errors

Except the generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The group id does not exist.
INVALID_NAME	Name is invalid.

8.81 Add a device to a group

8.81.1 Request URL

```
/rest/groups/{groupId}/devices
```

The parameter "groupId" is the ID of the group.

8.81.2 HTTP request method

POST

8.81.3 Request example


```
{
```

```
  "id": 1
}
```

8.81.4 Request field description

Field	Type	Attribute	Description
id	Integer	required	The device id.

8.81.5 Response example

N/A.

8.81.6 Response field description

N/A.

8.81.7 Errors

Except the generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The group id or device id does not exist.

8.82 Remove a device from a group

8.82.1 Request URL

```
/groups/{groupIdId}/devices/{deviceId}
```

"groupIdId" is the id of the target group. "deviceId" is the device to be removed from the group.

8.82.2 HTTP request method

DELETE

8.82.3 Request example

N/A.

8.82.4 Request field description

Field	Type	Attribute	Description
Id	Integer	required	The device id.

8.82.5 Response example

N/A.

8.82.6 Response field description

N/A.

8.82.7 Errors

Except the generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The group id or device id does not exist.

8.83 Delete a group

8.83.1 Request URL

```
/rest/groups/{id}
```

"id" is the id of the target group.

8.83.2 HTTP request method

DELETE

8.83.3 Request example

https://localhost:8643/DcmConsole/rest/groups/1

8.83.4 Request field description

N/A.

8.83.5 Response example

N/A.

8.83.6 Response field description

N/A.

8.83.7 Errors

Except the generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The group id does not exist.

8.84 Get miscellaneous configurations

8.84.1 Request URL

S

8.84.2 HTTP request method

GET

8.84.3 Request example

https://localhost:8643/DcmConsole/rest/configuration/misc?name=maxHealthTemperature

8.84.4 Request field description

Field	Type	Attribute	Description
Name	String	optional	When this parameter is not specified, all miscellaneous configurations will be returned. Possible values: maxHealthTemperature Maximum healthy temperature. pue Power usage effectiveness. electricityCostPerKWh Electricity cost per kWh. electricityRateCurrency Electricity rate currency. temperatureUnit Temperature unit. Celsius or Fahrenheit. weightUnit Weight unit. receiveSNMPAlerts Whether receive SNMP alerts or not. carbonEmissionStartingMonth Carbon emission starting month sustainabilityPredictEnabled Whether carbon emission prediction enabled or not carbonEmissionFactor Carbon emission factors

			highCefThreshold High carbon emission factor threshold lowCefThreshold Low carbon emission factor threshold highRenewableEnergyRatio High renewable energy ratio mediumRenewableEnergyRatio Medium renewable energy ratio lowRenewableEnergyRatio Low renewable energy ratio

8.84.5 Response example


```
{
  "content": [
    {
      "name": "maxHealthTemperature",
      "value": "27"
    },
    {
      "name": "pue",
      "value": "2"
    },
    {
      "name": "electricityCostPerKWh",
      "value": ""
    },
    {
      "name": "electricityRateCurrency",
      "value": ""
    },
    {
      "name": "temperatureUnit",
      "value": "Celsius"
    },
    {
      "name": "weightUnit",
      "value": "Kilogram"
    },
    {
      "name": "receiveSNMPAlerts",
      "value": "false"
    },
    {
      "name": "carbonEmissionStartingMonth",
      "value": "1"
    },
    {
      "name": "sustainabilityPredictEnabled",
      "value": "true"
    },
    {
      "name": "carbonEmissionFactor",
      "cefConfig": {
        "unit": 0,
        "cef": [
          [
            1,
            1,
            1,
            1,
            1,
```

```
                1,
                1,
                1,
                1,
                1,
                1,
                1,
                1,
                1,
                1,
                1,
                1,
                1,
                1,
                1,
                1,
                1,
                1,
                1,
                1
              ],
              [
                1,
                1
              ]
            ]
          }
        },
        {
          "name": "highCefThreshold",
          "value": "10.0"
        },
        {
          "name": "lowCefThreshold",
          "value": "5.0"
        },
        {
          "name": "highRenewableEnergyRatio",
          "value": "0.0"
        },
        {
          "name": "mediumRenewableEnergyRatio",
          "value": "0.15"
        },
        {
          "name": "lowRenewableEnergyRatio",
          "value": "0.5"
        }
      ]
    }
```


8.84.6 Response field description

Field	Type	Description
maxHealthTemperature	String	Maximum healthy temperature. The unit is specified by the field "temperatureUnit".
pue	String	Power usage effectiveness.
electricityCostPerKWh	String	Electricity cost per kWh.
electricityRateCurrency	String	Electricity rate currency.
temperatureUnit	String	Temperature unit. Possible values: Celsius

		Fahrenheit
weightUnit	String	Weight unit. Possible values: Kilogram Pound
receiveSNMPAlerts	String	Whether to receive SNMP alerts or not. Possible values: True false
carbonEmissionStartingMonth	Integer	Carbon emission starting month. The value is an integer from 1 to 12.
sustainabilityPredictEnabled	String	Whether carbon emission prediction is enabled or not. Possible values: true false
carbonEmissionFactor	Struct	Carbon emission factor data is stored in a structure named cefConfig, where cef field is a two-dimensional array and unit field is the unit used by cef array. The unit value 0 is for kilograms, and 1 is for pounds. All carbon emission factors are stored in the cef array. The first dimension of cef is 12, which represents 12 months. The second dimension of cef is 24, which means 24 hours.
highCefThreshold	Double	High carbon emission factor threshold.
lowCefThreshold	Double	Low carbon emission factor threshold.
highRenewableEnergyRatio	Double	High renewable energy ratio.
mediumRenewableEnergyRatio	Double	Medium renewable energy ratio.
lowRenewableEnergyRatio	Double	Low renewable energy ratio.

8.84.7 Errors

Generic errors defined in chapter "Error handling".

8.85 Update miscellaneous configurations

8.85.1 Request URL

```
/rest/configuration/misc
```

8.85.2 HTTP request method

PUT

8.85.3 Request example

```
{
  "name": "maxHealthTemperature",
  "value": "27"
}
```

8.85.4 Request field description

Field	Type	Attribute	Description
name	String	Required	The configuration name. For the possible values, refer to chapter "Get miscellaneous configurations".
value	String	Required	The configuration value. For the possible values, refer to chapter "Get miscellaneous configurations".

For other details, refer to chapter "Get miscellaneous configurations".

8.85.5 Response example

```
{
```

```
    "name": "maxHealthTemperature",
    "value": "27"
  }
```

8.85.6 Response field description

Field	Type	Description
Name	String	The configuration name. For the possible values, refer to chapter "Get miscellaneous configurations".
Value	String	The configuration value. For the possible values, refer to chapter "Get miscellaneous configurations".

For other details, refer to chapter "Get miscellaneous configurations".

8.85.7 Errors

Except Generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
INVALID_CONFIGURATION_NAME	The configuration name is invalid.
INVALID_CONFIGURATION_VALUE	The configuration value is invalid.

8.86 Get predefined event configurations

8.86.1 Request URL

```
/rest/configuration/predefinedEvents
```

8.86.2 HTTP request method

GET

8.86.3 Request example

https://localhost:8643/DcmConsole/rest/configuration/predefinedEvents

8.86.4 Request field description

N/A

8.86.5 Response example

```
{
  "content": [
    {
      "name": "IPMI_PWR_UNIT",
      "severity": "CUSTOM",
      "register": "FALSE"
    },
    {
      "name": "SNMP_PWR_UNIT",
      "severity": "CUSTOM",
      "register": "FALSE"
    },
    {
      "name": "IPMI_PWR_SUPPLY",
      "severity": "CUSTOM",
      "register": "FALSE"
    },
    {
      "name": "IPMI_PROCESSOR_THERMAL_TRIP",
      "severity": "CUSTOM",
      "register": "FALSE"
    },
    {
      "name": "IPMI_FAN",
```

```
      "severity": "CUSTOM",
      "register": "FALSE"
    },
    {
      "name": "BMC_SEL_REACHING_FULL_CAPACITY",
      "severity": "WARNING",
      "register": "FALSE"
    },
    {
      "name": "BMC_SEL_AT_FULL_CAPACITY",
      "severity": "ERROR",
      "register": "FALSE"
    },
    {
      "name": "ENTITY_PROPERTY_CHANGED",
      "severity": "INFORMATIVE",
      "register": "FALSE"
    },
    {
      "name": "COMMUNICATION_WITH_NODE_FAILED",
      "severity": "ERROR",
      "register": "FALSE"
    },
    {
      "name": "COMMUNICATION_WITH_NODE_RESTORED",
      "severity": "INFORMATIVE",
      "register": "FALSE"
    },
    {
      "name": "INSUFFICIENT_NODE_PERMISSION",
      "severity": "ERROR",
      "register": "FALSE"
    },
    {
      "name": "CONTROL_POLICY_CANNOT_BE_MAINTAINED",
      "severity": "ERROR",
      "register": "FALSE"
    },
    {
      "name": "CANT_SET_NODE_EVENT",
      "severity": "ERROR",
      "register": "FALSE"
    },
    {
      "name": "CONTROL_POLICY_APPLIED",
      "severity": "INFORMATIVE",
      "register": "FALSE"
    },
    {
      "name": "CONTROL_POLICY_ENDED",
      "severity": "INFORMATIVE",
      "register": "FALSE"
    },
    {
      "name": "INTERNAL_ERROR",
      "severity": "CRITICAL ",
      "register": "TRUE"
    },
    {
      "name": "CONFIGURATION_CHANGED",
      "severity": "INFORMATIVE",
      "register": "FALSE"
    },
```

```
{
  "name": "HIERARCHY_CHANGED",
  "severity": "INFORMATIVE",
  "register": "FALSE"
},
{
  "name": "CONTROL_POLICY_CHANGED",
  "severity": "INFORMATIVE",
  "register": "FALSE"
},
{
  "name": "CUSTOM_EVENT_CHANGED",
  "severity": "INFORMATIVE",
  "register": "FALSE"
},
{
  "name": "EVENT_EVALUATION_FAILURE",
  "severity": "ERROR",
  "register": "FALSE"
},
{
  "name": "COLLECTION_STATE_CHANGED",
  "severity": "INFORMATIVE",
  "register": "FALSE"
},
{
  "name": "IMPORTHIERARCHY_COMPLETED",
  "severity": "INFORMATIVE",
  "register": "FALSE"
},
{
  "name": "DB_MAINTENANCE_STARTING",
  "severity": "INFORMATIVE",
  "register": "FALSE"
},
{
  "name": "DB_MAINTENANCE_ENDED",
  "severity": "INFORMATIVE",
  "register": "FALSE"
},
{
  "name": "DB_CONNECTION_FAILED",
  "severity": "CRITICAL",
  "register": "TRUE"
},
{
  "name": "DB_CONNECTION_RESTORED",
  "severity": "INFORMATIVE",
  "register": "FALSE"
},
{
  "name": "CONTROL_POLICY_PRIORITY_CONFLICT",
  "severity": "WARNING",
  "register": "FALSE"
},
{
  "name": "POWER_ON_MACHINE_FAILED",
  "severity": "WARNING",
  "register": "FALSE"
},
{
  "name": "POWER_OFF_MACHINE_FAILED",
  "severity": "WARNING",
```

```
    "register": "FALSE"
  },
  {
    "name": "CRITICAL_DATA_SYNCHRONIZATION_STARTING",
    "severity": "INFORMATIVE",
    "register": "FALSE"
  },
  {
    "name": "CRITICAL_DATA_SYNCHRONIZATION_ENDED",
    "severity": "INFORMATIVE",
    "register": "FALSE"
  },
  {
    "name": "NOTIFICATION_CHANGED",
    "severity": "INFORMATIVE",
    "register": "FALSE"
  },
  {
    "name": "NOTIFICATION",
    "severity": "INFORMATIVE",
    "register": "FALSE"
  },
  {
    "name": "SET_POLICY_TO_ENTITY_FAILED",
    "severity": "ERROR",
    "register": "FALSE"
  },
  {
    "name": "REMOVE_POLICY_FROM_ENTITY_FAILED",
    "severity": "ERROR",
    "register": "FALSE"
  },
  {
    "name": "COMMUNICATION_WITH_ENCLOSURE_FAILED",
    "severity": "ERROR",
    "register": "FALSE"
  },
  {
    "name": "COMMUNICATION_WITH_ENCLOSURE_RESTORED",
    "severity": "INFORMATIVE",
    "register": "FALSE"
  },
  {
    "name": "UPS_BAD_BATTERY",
    "severity": "ERROR",
    "register": "FALSE"
  },
  {
    "name": "UPS_LOW_BATTERY",
    "severity": "WARNING",
    "register": "FALSE"
  },
  {
    "name": "UPS_BAD_TEMPERATURE",
    "severity": "ERROR",
    "register": "FALSE"
  },
  {
    "name": "UPS_BAD_INPUT",
    "severity": "ERROR",
    "register": "FALSE"
  },
  {
```

```
    "name": "UPS_BAD_OUTPUT",
    "severity": "ERROR",
    "register": "FALSE"
  },
  {
    "name": "UPS_OVERLOAD",
    "severity": "WARNING",
    "register": "FALSE"
  },
  {
    "name": "UPS_ON_BYPASS",
    "severity": "INFORMATIVE",
    "register": "FALSE"
  },
  {
    "name": "UPS_BAD_BYPASS",
    "severity": "ERROR",
    "register": "FALSE"
  },
  {
    "name": "UPS_SHUTDOWN",
    "severity": "WARNING",
    "register": "FALSE"
  },
  {
    "name": "UPS_CHARGE_FAILURE",
    "severity": "ERROR",
    "register": "FALSE"
  },
  {
    "name": "UPS_FAN_FAILURE",
    "severity": "ERROR",
    "register": "FALSE"
  },
  {
    "name": "UPS_COMMUNICATION_LOST",
    "severity": "ERROR",
    "register": "FALSE"
  },
  {
    "name": "ENTITY_WITH_DUPLICATED_PLATFORMID",
    "severity": "CRITICAL",
    "register": "TRUE"
  },
  {
    "name": "PDU_LOW_LOAD",
    "severity": "CUSTOM",
    "register": "FALSE"
  },
  {
    "name": "PDU_HIGH_LOAD",
    "severity": "CUSTOM",
    "register": "FALSE"
  },
  {
    "name": "PDU_OVERLOAD",
    "severity": "CUSTOM",
    "register": "FALSE"
  },
  {
    "name": "PDU_OUTLET_ON",
    "severity": "CUSTOM",
    "register": "FALSE"
```

```
    },
    {
      "name": "PDU_OUTLET_OFF",
      "severity": "CUSTOM",
      "register": "FALSE"
    },
    {
      "name": "ENTITY_CAPABILITIES_CHANGED",
      "severity": "INFORMATIVE",
      "register": "FALSE"
    },
    {
      "name": "PDU_OUTLET_LOW_LOAD",
      "severity": "CUSTOM",
      "register": "FALSE"
    },
    {
      "name": "PDU_OUTLET_HIGH_LOAD",
      "severity": "CUSTOM",
      "register": "FALSE"
    },
    {
      "name": "PDU_OUTLET_OVERLOAD",
      "severity": "CUSTOM",
      "register": "FALSE"
    },
    {
      "name": "PLATFORM_OPERATION_FAILED",
      "severity": "ERROR",
      "register": "FALSE"
    },
    {
      "name": "IPMI_TEST_EVENT",
      "severity": "INFORMATIVE",
      "register": "FALSE"
    },
    {
      "name": "DEVICES_WITHOUT_PWR_MONITORING",
      "severity": "WARNING",
      "register": "TRUE"
    },
    {
      "name": "SERVER_COMPONENT_HIGH_TEMPERATURE",
      "severity": "CUSTOM",
      "register": "FALSE"
    },
    {
      "name": "COOLING_ANOMALY",
      "severity": "CRITICAL",
      "register": "TRUE"
    },
    {
      "name": "DEVICE_COMPONENT_FAULT",
      "severity": "WARNING",
      "register": "TRUE"
    },
    {
      "name": "PREDICTIVE_FAN_FAILURE",
      "severity": "CUSTOM",
      "register": "FALSE"
    }
  ]
}
```

8.86.6 Response field description

Field	Type	Description
name	String	Predefined event name. Refer to the following table for the predefined event definitions.
severity	String	The severity of the event.
register	String	"True" means this kind of event will be received or filtered.

Predefined Event Name	Severity	Description
IPMI_PWR_UNIT	CUSTOM	IPMI originated event related to power unit.
SNMP_PWR_UNIT	CUSTOM	SNMP originated event related to power unit.
IPMI_PWR_SUPPLY	CUSTOM	IPMI originated event related to the power supply.
IPMI_PROCESSOR_THERMAL_TRIP	CUSTOM	IPMI originated event related to the processor thermal trip.
IPMI_FAN	CUSTOM	IPMI originated event related to the fan.
BMC_SEL_REACHING_FULL_CAPACITY	WARNING	BMC's system event log is nearing full capacity (equal or larger than75% capacity).
BMC_SEL_AT_FULL_CAPACITY	ERROR	BMC's system event log is full (equal or larger than 97% capacity).
ENTITY_PROPERTY_CHANGED	INFORMATIVE	Entity properties have changed. A property of an entity changed.
COMMUNICATION_WITH_NODE_FAILED	ERROR	DCM lost communication with a node.
COMMUNICATION_WITH_NODE_RESTORED	INFORMATIVE	DCM restored communication with a node.
INSUFFICIENT_NODE_PERMISSION	ERROR	Node privilege level is insufficient.
CONTROL_POLICY_CANNOT_BE_MAINTAINED	ERROR	DCM cannot maintain power policy.
CANT_SET_NODE_EVENT	ERROR	DCM cannot set a node event.
CONTROL_POLICY_APPLIED	INFORMATIVE	DCM started to apply a control policy because of a policy schedule or an API call.
CONTROL_POLICY_ENDED	INFORMATIVE	DCM ended a control policy because of a policy schedule or an API call.
INTERNAL_ERROR	CRITICAL	DCM internal error.
CONFIGURATION_CHANGED	INFORMATIVE	A configuration change occurred. This error also provides the change details.
HIERARCHY_CHANGED	INFORMATIVE	A change in the datacenter hierarchy occurred.
CONTROL_POLICY_CHANGED	INFORMATIVE	A change in the definition of a control policy occurred.
CUSTOM_EVENT_CHANGED	INFORMATIVE	A change in the definition of a custom event occurred.
EVENT_EVALUATION_FAILURE	ERROR	DCM could not complete a custom event measurement because the data is incomplete for the evalPeriod specified.
COLLECTION_STATE_CHANGED	INFORMATIVE	collection state operation setting is completed.
IMPORTHIERARCHY_COMPLETED	INFORMATIVE	Importing hierarchy operation is completed.

DB_MAINTENANCE_STARTING	INFORMATIVE	Database maintenance is about to start.
DB_MAINTENANCE_ENDED	INFORMATIVE	Database maintenance is ended.
DB_CONNECTION_FAILED	CRITICAL	The connection to the database failed.
DB_CONNECTION_RESTORED	INFORMATIVE	The connection to the database was restored.
CONTROL_POLICY_PRIORITY_CONFLICT	WARNING	Priority conflict in a single policy has been detected and internally resolved by DCM.
POWER_ON_MACHINE_FAILED	WARNING	Fail to power-on the machine specified by user.
POWER_OFF_MACHINE_FAILED	WARNING	Fail to power-off the machine specified by user.
CRITICAL_DATA_SYNCHRONIZATION_STARTING	INFORMATIVE	Inconsistent critical data are detected, and synchronization operation is about to start.
CRITICAL_DATA_SYNCHRONIZATION_ENDED	INFORMATIVE	Critical data synchronization is ended.
NOTIFICATION_CHANGED	INFORMATIVE	A change in the definition of a notification occurred.
NOTIFICATION	INFORMATIVE	Define a notification on an entity.
SET_POLICY_TO_ENTITY_FAILED	ERROR	Fail to set a policy on an entity.
REMOVE_POLICY_FROM_ENTITY_FAILED	ERROR	Fail to remove a policy from an entity.
COMMUNICATION_WITH_ENCLOSURE_FAILED	ERROR	DCM lost communication with an enclosure.
COMMUNICATION_WITH_ENCLOSURE_RESTORED	INFORMATIVE	DCM restored communication with an enclosure.
UPS_BAD_BATTERY	ERROR	Event of battery failure in UPS.
UPS_LOW_BATTERY	WARNING	Event of remaining battery capacity under a predefined threshold in UPS.
UPS_BAD_TEMPERATURE	ERROR	Reserved for events related to temperature limits & thresholds exceeding in UPS.
UPS_BAD_INPUT	ERROR	Event of power input failure in UPS.
UPS_BAD_OUTPUT	ERROR	Reserved for events related to power output failure in UPS.
UPS_OVERLOAD	WARNING	Reserved for events related to output power load limits & thresholds exceeding in UPS.
UPS_ON_BYPASS	INFORMATIVE	Event of switching to bypass state in UPS.
UPS_BAD_BYPASS	ERROR	Reserved for events related to bypass failure in UPS.
UPS_SHUTDOWN	WARNING	Event of UPS shutdown.
UPS_CHARGE_FAILURE	ERROR	Reserved for events related to charge failure in UPS.
UPS_FAN_FAILURE	ERROR	Reserved for events related to power fan failure in UPS.
UPS_COMMUNICATION_LOST	ERROR	Reserved for events related to communication lost in UPS.
ENTITY_WITH_DUPLICATED_PLATFORMID	CRITICAL	Event on duplicated platform ID.
PDU_LOW_LOAD	CUSTOM	Event indicating that PDU is under low load.
PDU_HIGH_LOAD	CUSTOM	Event indicating PDU is near overload.
PDU_OVERLOAD	CUSTOM	Event indicating PDU is overloaded.

PDU_OUTLET_ON	CUSTOM	Event indicating that one of the PDU outlets is powered on.
PDU_OUTLET_OFF	CUSTOM	Event indicating that one of the PDU outlets is powered off.
ENTITY_CAPABILITIES_CHANGED	INFORMATIVE	Entity capabilities have changed.
PDU_OUTLET_LOW_LOAD	CUSTOM	Event indicating that one of the PDU outlets is under low load.
PDU_OUTLET_HIGH_LOAD	CUSTOM	Event indicating that one of the PDU outlets is above high load.
PDU_OUTLET_OVERLOAD	CUSTOM	Event indicating that one of the PDU outlets is overloaded.
PLATFORM_OPERATION_FAILED	ERROR	DCM could communicate with a device which returns operation failure to DCM.
IPMI_TEST_EVENT	INFORMATIVE	DCM could route the IPMI test event to external event listener.
DEVICES_WITHOUT_PWR_MONITORING	WARNING	Device(s) do(es) not have power monitoring capability.
SERVER_COMPONENT_HIGH_TEMPERATURE	CUSTOM	Server component temperature reaches the threshold.
COOLING_ANOMALY	CRITICAL	Event indicating a cooling anomaly.
DEVICE_COMPONENT_FAULT	WARNING	Event indicating a device component fault.
PREDICTIVE_FAN_FAILURE	CUSTOM	Event indicating a predictive fan failure.

8.86.7 Errors

Generic errors defined in chapter "Error handling".

8.87 Update predefined event configurations

8.87.1 Request URL

/rest/configuration/predefinedEvents

8.87.2 HTTP request method

PUT

8.87.3 Request example

```
{
  "name": "IPMI_PWR_UNIT",
  "register": "TRUE"
}
```

8.87.4 Request field description

Field	Type	Attribute	Description
name	String	Required	The predefined event name. For the possible values, refer to chapter "Get predefined event configuration".
register	String	Required	Possible values: True The event will be received. False The event will be filtered.

8.87.5 Response example

```
{
  "name": "IPMI_PWR_UNIT",
  "register": "TRUE"
}
```

8.87.6 Response field description

Refer to Request field description.

8.87.7 Errors

Except Generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
INVALID_EVENT_NAME	The event name field is required, or the value is invalid.
INVALID_REGISTER_VALUE	The register field is missing, or the value is invalid.

8.88 Export CPU telemetry data

8.88.1 Request URL

```
/rest/configuration/cpuTelemetry
```

8.88.2 HTTP request method

GET

8.88.3 Request example

https://localhost:8643/DcmConsole/rest/configuration/cpuTelemetry?startDateTime=2019-08-14T22:00:00%2B08:00&endDateTime=2019-08-14T23:00:00%2B08:00&hasCapability=false

8.88.4 Request field description

Field	Type	Attribute	Description
startDateTime	String	Required	The start time. The format is "yyyy-MM-dd'T'HH:mm:ssXXX". For an example, 2018-12-12T01:01:01+08:00.
endDateTime	String	Optional	If it is not presented, the current time will be the default time.
hasCapability	Boolean	Optional	The default value is true.

8.88.5 Response example

N/A.

8.88.6 Response field description

N/A.

8.88.7 Errors

Except Generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
INVALID_START_TIME	
START_TIME_BEFORE_END_TIME	The start time should be before the end time.

8.89 Get SiAAS configurations

8.89.1 Request URL

```
/rest/configuration/SiAASConfig
```

8.89.2 HTTP request method

GET

8.89.3 Request example

https://localhost:8643/DcmConsole/rest/configuration/SiAASConfig

8.89.4 Request field description

N/A.

8.89.5 Response example

```
{
  "soldTo": "xyza",
  "shipTo": "xyzb",
  "customIdPrefix": "abcde",
  "enablePCC": "false"
}
```

8.89.6 Response field description

Field	Type	Description
soldTo	String	Sold-To customer ID.
shipTo	String	Ship-To customer ID.
customIdPrefix	String	Custom ID prefix.
enablePCC	String	Whether PCC is enabled or not. Possible values: true false

8.89.7 Errors

Generic errors defined in chapter "Error handling".

8.90 Update SiAAS configurations

8.90.1 Request URL

```
/rest/configuration/SiAASConfig
```

8.90.2 HTTP request method

PUT

8.90.3 Request example

```
{
  "soldTo": "xyzd"
}
```

8.90.4 Request field description

N/A.

8.90.5 Response example

```
{
  "soldTo": "xyzd",
  "shipTo": "xyzb",
  "customIdPrefix": "abcde",
  "enablePCC": "false"
}
```

8.90.6 Response field description

Refer to the chapter "Get SiAAS configurations".

8.90.7 Errors

Except Generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
INVALID_SOLDTO	Invalid soldTo.
INVALID_SHIPTO	Invalid shipTo.
INVALID_CUSTOMIDPREFIX	Invalid customIdPrefix. The maximum length is 5.
INVALID_ENABLEPCC_VALUE	Invalid enablePCC value.

8.91 Export SiAAS telemetry data

8.91.1 Request URL

```
/rest/configuration/SiAASTelemetry
```

8.91.2 HTTP request method

GET

8.91.3 Request example

https://localhost:8643/DcmConsole/rest/configuration/SiAASTelemetry

8.91.4 Request field description

N/A.

8.91.5 Response example

```
{
  "stateCertificates": [
    {
      "timestamp": "2021-07-27T02:27:08.0257187Z",
      "customerMetadata": null,
      "customerId": {
        "soldTo": "xyza",
        "shipTo": "xyzb"
      },
      "intelLicenseData": null,
      "intelTelemetryData": []
    }
  ]
}
```

8.91.6 Response field description

N/A.

8.91.7 Errors

Except Generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
SOLDTO_IS_REQUIRED	The "soldTo" of SiAASConfig is required.
SHIPTO_IS_REQUIRED	The "shipTo" of SiAASConfig is required.
CUSTOMIDPREFIX_IS_REQUIRED	The "customIdPrefix" of SiAASConfig is required.

***Note**: Refer to the chapter "Update SiAAS configurations" to set SiAASConfig if some of its values are missing.*

8.92 Get emergency power reduction rooms

8.92.1 Request URL

/rest/configuration/EPRRooms

8.92.2 HTTP request method

GET

8.92.3 Request example

https://localhost:8643/DcmConsole/rest/configuration/EPRRooms

8.92.4 Request field description

N/A.

8.92.5 Response example


```
{
  "links": [
    {
      "rel": "self",
      "href": "https://localhost:8643/DcmConsole/rest/configuration/EPRRooms"
    },
    {
      "rel": "add",
      "href": "https://localhost:8643/DcmConsole/rest/configuration/EPRRooms"
    }
  ],
  "content": [
    {
      "parentId": 3,
      "id": 5,
      "name": "test-room-1",
      "pue": 0,
      "links": [
        {
          "rel": "self",
          "href":
"https://localhost:8643/DcmConsole/rest/configuration/EPRRooms/5"
        },
        {
          "rel": "delete",
          "href":
"https://localhost:8643/DcmConsole/rest/configuration/EPRRooms/5"
        }
      ]
    }
  ]
}
```

8.92.6 Response field description

Refer to chapter "Get rooms".

8.92.7 Errors

Generic errors defined in chapter "Error handling".

8.93 Add an emergency power reduction room

8.93.1 Request URL

```
/rest/configuration/EPRRooms
```

8.93.2 HTTP request method

POST

8.93.3 Request example

https://localhost:8643/DcmConsole/rest/configuration/EPRRooms

8.93.4 Request field description

```
{
  "id": "1"
}
```

8.93.5 Response example

```
{
  "parentId": 3,
  "id": 1,
  "name": "test-room-1",
  "pue": 0,
  "links": [
    {
      "rel": "self",
      "href": "https://localhost:8643/DcmConsole/rest/configuration/EPRRooms/1"
    },
    {
      "rel": "delete",
      "href": "https://localhost:8643/DcmConsole/rest/configuration/EPRRooms/1"
    }
  ]
}
```

8.93.6 Response field description

Refer to chapter "Get rooms".

8.93.7 Errors

Except Generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The room does not exist.

8.94 Delete an emergency power reduction room

8.94.1 Request URL

```
/rest/configuration/EPRRooms/{roomId}
```

"roomId" is the id of the room.

8.94.2 HTTP request method

DELETE

8.94.3 Request example

https://localhost:8643/DcmConsole/rest/configuration/EPRRooms/1

8.94.4 Request field description

N/A.

8.94.5 Response example

N/A.

8.94.6 Response field description

N/A.

8.94.7 Errors

Except Generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The room does not exist.

8.95 Add a policy

8.95.1 Request URL

/rest/policies

8.95.2 HTTP request method

POST

8.95.3 Request example

```
{
  "name": "policy-1",
  "targetId": "1",
  "policyType": "Custom Power Policy",
  "threshold": "1000",
  "reserveBudget": "200",
  "enabled": "true",
  "startTime": "2019-08-02'T'08:00:00+08:00",
  "endTime": "2019-08-10'T'08:00:00+08:00",
  "startHour": "08:00",
  "endHour": "16:30",
  "days": [
    "MONDAY",
    "TUESDAY"
  ],
  "priorities": [
    {
      "childId": "10",
      "priority": "High"
    },
    {
      "childId": "11",
      "priority": "Low"
    }
  ]
}
```

8.95.4 Request field description

Field	Type	Attribute	Description
name	String	Required	The name of the policy.
targetId	Integer	Required	The ID of the target entity. For an example, if a policy is applied to a datacenter, the "targetId" will be the ID of the datacenter.
policyType	String	Required	Policy type. Possible values: Custom Power Policy Minimum Power Policy
threshold	Integer	Optional	The threshold to control. If the policy type is "Custom Power Policy", the field should be provided with an integer value greater than 0. If the policy type is "Minimum Power Polciy", the field will be ignored.
reserveBudget	Integer	Optional	The reserved threshold. If the policy type is "Custom Power Policy", the field should be provided with a non-negative integer value. If the policy type is "Minimum Power Polciy", the field will be ignored.
enabled	Boolean	Required	Value "true" means the policy will be enabled.
startTime	String	Optional	The start time to apply the policy. The format is "yyyy-MM-dd'T'HH:mm:ssXXX". For example, 2019-08-02T11:00:00+08:00.
endTime	String	Optional	The end time to apply the policy.
startHour	String	Optional	The start hour with the format "HH:mm". It is used to specify the start time to apply the policy in the days defined in the parameter "days".
endHour	String	Optional	The end hour with the format "HH:mm". It is used to specify the end time to apply the policy in the days defined in the parameter "days".
days	String Array	Optional	Specify the days to apply the policy. Possible values: SUNDAY MONDAY TUESDAY WEDNESDAY THURSDAY FRIDAY SATURDAY
priorities	Array	Optional	When adding a policy to a rack or an enclosure without power control capability, we use this field to specify the priorities of the direct children with power control capability.
childId	Integer	Optional	The id of a device under a rack or an enclosure without power control capability. The device should have power control capability.
priority	String	Optional	Possible values: Low Medium High Critical

***Note**:*

- *To add a permanent policy, do not present the field "startTime", "endTime", "startHour" and "endHour", or set them with null value.*
- *To schedule a policy at a specific time period, only the field "startTime" and "endTime" need to be set. Do not set the field "startHour" and "endHour" or set them with null value.*

- *To schedule a recurrent policy, use the field "startTime" and "endTime" to specify the time period, use field "startHour" and "endHour" to specify the time period of each day, and use "days" field to specify the working days.*

8.95.5 Response example


```
{
  "policyId": "1",
  "active": "true",
  "name": "policy-1",
  "targetId": "1",
  "policyType": "Custom Power Policy",
  "threshold": "1000",
  "reserveBudget": "200",
  "Enabled": "true",
  "startTime": "2019-08-02'T'08:00:00+08:00",
  "endTime": "2019-08-10'T'08:00:00+08:00",
  "startHour": "8",
  "endHour": "16",
  "days": [
    "MONDAY",
    "TUESDAY"
  ],
  "priorities": [
    {
      "childId": "10",
      "priority": "High"
    },
    {
      "childId": "11",
      "priority": "Low"
    }
  ],
  "links": [
    {
      "rel": "self",
      "href": "https://localhost:8643/DcmConsole/rest/policies/1"
    },
    {
      "rel": "update",
      "href": "https://localhost:8643/DcmConsole/rest/policies/1"
    },
    {
      "rel": "delete",
      "href": "https://localhost:8643/DcmConsole/rest/policies/1"
    }
  ]
}
```


8.95.6 Response field description

Field	Type	Description
policyId	Integer	The id of the policy.
active	Boolean	If the policy is active.

Refer to chapter "Add a policy" for other field definition.

8.95.7 Errors

Except Generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The target id or the id in "priorities" does not exist.

POLICY_TYPE_IS_REQUIRED	Policy type is required.
INVALID_POLICY_TYPE	Invalid policy type.
INVALID_THRESHOLD	Invalid threshold.
INVALID_RESERVE_BUDGET	Invalid reserve budget.
INVALID_START_HOUR	Invalid start hour.
INVALID_END_HOUR	Invalid end hour.
INVALID_DAYS	Invalid days.
INVALID_PRIORITY	Invalid priority.
INVALID_PARAMETER	Invalid parameter.
END_TIME_IS_NOT_AFTER_NOW	End time should be after now.
START_HOUR_BEFORE_END_HOUR	Start hour should be before end hour.
START_HOUR_END_HOUR_SHOULD_BE_TOGETHER	Start hour and end hour should be specified at the same time.

8.96 Get policies

8.96.1 Request URL

/rest/policies

8.96.2 HTTP request method

GET

8.96.3 Request example

https://localhost:8643/DcmConsole/rest/policies

8.96.4 Request field description

N/A.

8.96.5 Response example


```
{
  "links": [
    {
      "rel": "self",
      "href": "https://localhost:8643/DcmConsole/rest/policies"
    },
    {
      "rel": "add",
      "href": "https://localhost:8643/DcmConsole/rest/policies"
    }
  ],
  "content": [
    {
      "policyId": "1",
      "active": "true",
      "name": "policy-1",
      "targetId": "1",
      "policyType": "Custom Power Policy",
      "threshold": "1000",
      "reserveBudget": "200",
      "Enabled": "true",
```

```
        "startTime": "2019-08-02'T'08:00:00+08:00",
        "endTime": "2019-08-10'T'08:00:00+08:00",
        "startHour": "8",
        "endHour": "16",
        "days": [
          "MONDAY",
          "TUESDAY"
        ],
        "priorities": [
          {
            "childId": "10",
            "priority": "High"
          },
          {
            "childId": "11",
            "priority": "Low"
          }
        ],
        "links": [
          {
            "rel": "self",
            "href": "https://localhost:8643/DcmConsole/rest/policies/1"
          },
          {
            "rel": "update",
            "href": "https://localhost:8643/DcmConsole/rest/policies/1"
          },
          {
            "rel": "delete",
            "href": "https://localhost:8643/DcmConsole/rest/policies/1"
          }
        ]
      }
    ]
  }
```


8.96.6 Response field description

Refer to the chapter "Add a policy".

8.96.7 Errors

Generic errors defined in chapter "Error handling".

8.97 Get a policy

8.97.1 Request URL

```
/rest/policies/{policyId}
```

The parameter "policyId" is the ID of the policy.

8.97.2 HTTP request method

GET

8.97.3 Request example

https://localhost:8643/DcmConsole/rest/policies/1

8.97.4 Request field description

N/A.

8.97.5 Response example

```
{
  "policyId": "1",
  "active": "true",
  "name": "policy-1",
  "targetId": "1",
  "policyType": "Custom Power Policy",
  "threshold": "1000",
  "reserveBudget": "200",
  "Enabled": "true",
  "startTime": "2019-08-02'T'08:00:00+08:00",
  "endTime": "2019-08-10'T'08:00:00+08:00",
  "startHour": "8",
  "endHour": "16",
  "days": [
    "MONDAY",
    "TUESDAY"
  ],
  "priorities": [
    {
      "childId": "10",
      "priority": "High"
    },
    {
      "childId": "11",
      "priority": "Low"
    }
  ],
  "links": [
    {
      "rel": "self",
      "href": "https://localhost:8643/DcmConsole/rest/policies/1"
    },
    {
      "rel": "update",
      "href": "https://localhost:8643/DcmConsole/rest/policies/1"
    },
    {
      "rel": "delete",
      "href": "https://localhost:8643/DcmConsole/rest/policies/1"
    }
  ]
}
```

8.97.6 Response field description

Refer to the chapter "Add a policy".

8.97.7 Errors

Except Generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The target does not exist.

8.98 Update a policy

8.98.1 Request URL

```
/rest/policies/{policyId}
```

The parameter "policyId" is the ID of the policy.

8.98.2 HTTP request method

PUT

8.98.3 Request example

```
{
  "policyId": "1",
  "name": "policy-1",
  "threshold": "1000",
  "reserveBudget": "200",
  "Enabled": "true",
  "startTime": "2019-08-02'T'08:00:00+08:00",
  "endTime": "2019-08-10'T'08:00:00+08:00",
  "startHour": "8",
  "endHour": "16",
  "days": [
    "MONDAY",
    "TUESDAY"
  ],
  "priorities": [
    {
      "childId": "10",
      "priority": "High"
    },
    {
      "childId": "11",
      "priority": "Low"
    }
  ]
}
```

8.98.4 Request field description

Except the field "policyId, ""policyType" and "targetId", other fields can be updated. Refer to the chapter "Add a policy" for field definition.

8.98.5 Response example

```
{
  "policyId": "1",
  "active": "true",
  "name": "policy-1",
  "targetId": "1",
  "policyType": "Custom Power Policy",
  "threshold": "1000",
  "reserveBudget": "200",
  "Enabled": "true",
  "startTime": "2019-08-02'T'08:00:00+08:00",
  "endTime": "2019-08-10'T'08:00:00+08:00",
  "startHour": "8",
  "endHour": "16",
  "days": [
    "MONDAY",
    "TUESDAY"
  ],
  "priorities": [
    {
      "childId": "10",
      "priority": "High"
    },
    {
```

```
        "childId": "11",
        "priority": "Low"
      }
    ],
    "links": [
      {
        "rel": "self",
        "href": "https://localhost:8643/DcmConsole/rest/policies/1"
      },
      {
        "rel": "update",
        "href": "https://localhost:8643/DcmConsole/rest/policies/1"
      },
      {
        "rel": "delete",
        "href": "https://localhost:8643/DcmConsole/rest/policies/1"
      }
    ]
  }
```


8.98.6 Response field description

Refer to the chapter "Add a policy".

8.98.7 Errors

Except Generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The policy id or the id in "priorities" field does not exist.
INVALID_POLICY_TYPE	Invalid policy type.
INVALID_THRESHOLD	Invalid threshold.
INVALID_RESERVE_BUDGET	Invalid reserve budget.
INVALID_START_HOUR	Invalid start hour.
INVALID_END_HOUR	Invalid end hour.
INVALID_DAYS	Invalid days.
INVALID_PRIORITY	Invalid priority.
INVALID_PARAMETER	Invalid parameter.
POLICY_TYPE_IS_NOT_UPDATEABLE	Policy type cannot be updated.
END_TIME_IS_NOT_AFTER_NOW	End time should be after now.
START_HOUR_BEFORE_END_HOUR	Start hour should be before end hour.
START_HOUR_END_HOUR_SHOULD_BE_TOGETHER	Start hour and end hour should be specified at the same time.
TARGET_ID_IS_NOT_UPDATEABLE	Target id cannot be updated.

8.99 Delete a policy

8.99.1 Request URL

/rest/policies/{policyId}

The parameter "policyId" is the ID of the policy.

8.99.2 HTTP request method

DELETE

8.99.3 Request example

https://localhost:8643/DcmConsole/rest/policies/1

8.99.4 Request field description

N/A.

8.99.5 Response example

N/A.

8.99.6 Response field description

N/A.

8.99.7 Errors

Generic errors defined in chapter "Error handling".

8.100 Add a discovery task

8.100.1 Request URL

/rest/discoveryTasks

8.100.2 HTTP request method

POST

8.100.3 Request example

```
{
  "protocolType": "IPMI",
  "startIP": "192.168.1.1",
  "endIP": "192.168.1.255",
  "subnetMask": "255.255.255.0",
  "description": "",
  "vendor": "ANY",
  "protocolInfo": [
    {
      "name": "BMC_USER",
      "value": "user1"
    },
    {
      "name": "BMC_PASSWORD",
      "value": "mypassword"
    },
    {
      "name": "BMC_KEY",
      "value": ""
    }
  ]
}
```

8.100.4 Request field description

Field	Type	Attribute	Description
protocolType	String	Required	The protocol type. Possible values: IPMI SNMPv1v2c SNMPv3

			WS_MAN HTTPS SSH INBAND_PROTOCOL
startIP	String	Required	The start IP address.
endIP	String	Required	The end IP address.
subnetMask	String	Optional	The default value is "255.255.255.0".
description	String	Optional	Notes for the task.
authenticateEntity	Boolean	Optional	Whether it is need to validate the device. For HTTPS protocol, the device certificate needs to be imported into DCM key store if the field value is true. By default, the value is true for most of protocols.
protocolInfo	Array	Required	An Array to specify the necessary protocol fields. Each element is an object with a name property and a value property which are used to specify the required field. The fields depend on the protocol type: IPMI BMC_USER The BMC user name. Empty string will be used if not present. BMC_PASSWORD The BMC user password. Empty string will be used if not present. BMC_KEY The BMC Key. Empty string will be used if not present. SNMPv1v2c SNMP_COMMUNITY_STRING The SNMP community string. Empty string will be used if not present. SNMPv3 SNMP_USER The SNMP user name. Empty string will be used if not present. SNMP_AUTHENTICATION_PASSWORD The SNMP authentication password. Empty string will be used if not present. SNMP_ENCRYPTION_PASSWORD The SNMP encryption password. Empty string will be used if not present. SNMP_AUTHENTICATION_PROTOCOL One of the values "MD5", "SHA1", "HMAC128-SHA224", "HMAC192-SHA256","HMAC256-SHA384" and "HMAC384-SHA512". "HMAC192-SHA256" will be used if not present. SNMP_ENCRYPTION_PROTOCOL One of the values "DES", "3DES", "AES128", "AES192" and "AES256". "AES128" will be used if not present. WS_MAN HTTPS_PORT HTTPS port. "443" will be used if not present. HTTP_BASIC_USER HTTPS user name. Empty string will be used if not present. HTTP_BASIC_PASSWORD Empty string will be used if not present. HTTPS HTTPS_PORT

			"443" will be used if not present. HTTP_BASIC_USER Empty string will be used if not present. HTTP_BASIC_PASSWORD Empty string will be used if not present. SSH SSH_USER SSH user name. Empty string will be used if not present. SSH_PASSWORD SSH user password. Empty string will be used if not present. SSH_PORT SSH port. "22" will be used if not present. INBAND_PROTOCOL INBAND_USERNAME The in-band OS user name. Empty string will be used if not present. INBAND_PASSWORD The in-band OS user password. Empty string will be used if not present. INBAND_SSH_PORT The in-band SSH port Which is useful only when the in-band OS is Linux, XEN or ESXi. If this field is not specified, the default value "22" will be used.
vendor	String	Optional	The vendor name. If any vendor is specified, DCM will add the device only if the identified devices match the known vendor. If not specified or "ANY" specified, all known vendors are supported. "OTHERS" means all other vendors but the enumerated vendors. The possible values: ANY, ALTOS, AMD, APC, ARISTA, AVOCENT, BAYTECH, BROCADE, CHATSWORTH, CISCO, CONTOSO, DELL, EATON, EMC, EMERSON, ENLOGIC, EXTREMENETWORKS, F5, FUJITSU, H3C, HITACHI, HOFFMAN, HP, HUAWEI, IBM, INSPUR, INTEL,

			KONTRON, LENOVO, MELLANOX, NETAPP, NETTRIX, ORACLE, RARITAN, RSA, SERVERTECH, SIEMON, SUGON, SUPERMICRO, VERTIV, ZTE, OTHERS

8.100.5 Response example


```
{
  "taskId": "1",
  "createDate": "2019-01-17T10:00:00+08:00",
  "protocol": "IPMI",
  "startIP": "192.168.1.1",
  "endIP": "192.168.1.255",
  "subnetMask": "255.255.255.0",
  "description": "",
  "discoveredCount": "1",
  "addedDevices": [
    "10"
  ],
  "addedDeviceDetails": [
    {
      "addedDeviceId": "10",
      "addedDeviceTime": " 2019-01-17T11:00:00+08:00"
    }
  ],
  "progress": "0",
  "timeElapsed": "0",
  "links": [
    {
      "rel": "self",
      "href": "https://localhost:8643/DcmConsole/rest/discoveryTasks/1"
    },
    {
      "rel": "delete",
      "href": "https://localhost:8643/DcmConsole/rest/discoveryTasks/1"
    },
    {
      "rel": "control",
      "href": "https://localhost:8643/DcmConsole/rest/discoveryTasks/1"
    }
  ]
}
```


8.100.6 Response field description

Field	Type	Description
taskId	Integer	The id of the discovery task.
createDate	String	The creation time of the task. The format is "yyyy-MM-dd'T'HH:mm:ssXXX". For example, 2019-08-02T11:00:00+08:00.

discoveredCount	Integer	The total detected devices. For new devices, DCM will add them automatically. For existing devices, DCM will update their properties.
addedDevices	Array	The added device Id array. This field is replaced by the field "addedDeviceDetails" and will be removed.
addedDeviceDetails	Array	The array contains the added device details.
addedDeviceId	Integer	The id of the added device.
addedDeviceTime		The time to add the device. The format is "yyyy-MM-dd'T'HH:mm:ssXXX". For example, 2019-08-02T11:00:00+08:00.
progress	Integer	The value is from 0 to 100. The value "100" means the task is finished.
timeElapsed	Integer	The time spent since the task runs. The unit is second.
endDate	String	The end time of the task. The format is "yyyy-MM-dd'T'HH:mm:ssXXX". For example, 2019-08-02T11:00:00+08:00. This field will not be presented if the task is not finished.

Refer to the chapter "Request field description" for other field definition.

8.100.7 Errors

Except Generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
PROTOCOL_IS_REQUIRED	The protocol field is not present.
INVALID_PROTOCOL	Protocol is invalid.
START_IP_IS_REQUIRED	Start IP address is required.
INVALID_START_IP	Invalid start IP address.
END_IP_IS_REQUIRED	Start IP address is required.
INVALID_END_IP	Invalid end IP address.
INVALID_SUBNET_MASK	Invalid subnet mask.
INVALID_SNMP_AUTHENTICATION_PROTOCOL	Invalid SNMP_AUTHENTICATION_PROTOCOL value.
INVALID_SNMP_ENCRYPTION_PROTOCOL	Invalid SNMP_ENCRYPTION_PROTOCOL value.
INVALID_ HTTPS_PORT	Invalid HTTPS_PORT.
INVALID_ SSH_PORT	Invalid SSH_PORT.
INVALID_ INBAND_SSH_PORT	Invalid INBAND_SSH_PORT.

8.101 Get discovery tasks

8.101.1 Request URL

/rest/discoveryTasks

8.101.2 HTTP request method

GET

8.101.3 Request example

https://localhost:8643/DcmConsole/rest/discoveryTasks

8.101.4 Request field description

N/A.

8.101.5 Response example


```
{
  "links": [
    {
      "rel": "self",
      "href": "https://localhost:8643/DcmConsole/rest/discoveryTasks"
    },
    {
      "rel": "add",
      "href": "https://localhost:8643/DcmConsole/rest/discoveryTasks"
    }
  ],
  "content": [
    {
      "taskId": "1",
      "createDate": "2019-01-17T10:00:00+08:00",
      "protocol": "IPMI",
      "startIP": "192.168.1.1",
      "endIP": "192.168.1.255",
      "subnetMask": "255.255.255.0",
      "description": "",
      "discoveredCount": "1",
      "addedDevices": [
        "10"
      ],
      "addedDeviceDetails": [
        {
          "addedDeviceId": "10",
          "addedDeviceTime": "    2019-01-17T11:00:00+08:00"
        }
      ],
      "progress": "0",
      "timeElapsed": "0",
      "links": [
        {
          "rel": "self",
          "href": "https://localhost:8643/DcmConsole/rest/discoveryTasks/1"
        },
        {
          "rel": "delete",
          "href": "https://localhost:8643/DcmConsole/rest/discoveryTasks/1"
        },
        {
          "rel": "control",
          "href":
"https://localhost:8643/DcmConsole/rest/discoveryTasks/1/control"
        }
      ]
    }
  ]
}
```


8.101.6 Response field description

Refer to the chapter "Add a discovery task".

8.101.7 Errors

Generic errors defined in chapter "Error handling".

8.102 Get a discovery task

8.102.1 Request URL

/rest/discoveryTasks/{taskId}

The parameter "taskId" is the ID of the discovery task.

8.102.2 HTTP request method

GET

8.102.3 Request example

https://localhost:8643/DcmConsole/rest/discoveryTasks/1

8.102.4 Request field description

N/A.

8.102.5 Response example

Refer to the chapter "Add a discovery task".

8.102.6 Response field description

Refer to the chapter "Add a discovery task".

8.102.7 Errors

Except Generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The target does not exist.

8.103 Control a discovery task

8.103.1 Request URL

/rest/discoveryTasks/{taskId}/control

The parameter "taskId" is the ID of the discovery task.

8.103.2 HTTP request method

POST

8.103.3 Request example

```
{
  "controlAction": "STOP"
}
```

8.103.4 Request field description

Field	Type	Attribute	Description
controlAction	String	Required	Possible values: RERUN Re-run a stopped discovery task. STOP Stop a running discovery task.

8.103.5 Response example

```
{
  "taskId": "1",
  "createDate": "2019-01-17T10:00:00+08:00",
  "protocol": "IPMI",
```

```
  "startIP": "192.168.1.1",
  "endIP": "192.168.1.255",
  "subnetMask": "255.255.255.0",
  "description": "",
  "discoveredCount": "0",
  "progress": "0",
  "timeElapsed": "0",
  "links": [
    {
      "rel": "self",
      "href": "https://localhost:8643/DcmConsole/rest/discoveryTasks/1"
    },
    {
      "rel": "delete",
      "href": "https://localhost:8643/DcmConsole/rest/discoveryTasks/1"
    },
    {
      "rel": "control",
      "href": "https://localhost:8643/DcmConsole/rest/discoveryTasks/1/control"
    }
  ]
}
```


8.103.6 Response field description

Refer to the chapter "Add a discovery task" for field definitions.

8.103.7 Errors

Except the generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The task Id does not exist.
CONTROL_ACTION_IS REQUIRED	Control action is required.
INVALID_CONTROL_ACTION	Invalid control action value.
ACTION_IS_NOT_SUPPORT	Possible reasons: To re-run a running task is not supported. To stop a stopped task will always return OK.

8.104 Delete a discovery task

8.104.1 Request URL

```
/rest/discoveryTasks/{taskId}
```

The parameter "taskId" is the ID of the discovery task.

8.104.2 HTTP request method

DELETE

8.104.3 Request example

https://localhost:8643/DcmConsole/rest/discoveryTasks/1

8.104.4 Request field description

N/A.

8.104.5 Response example

N/A.

8.104.6 Response field description

N/A.

8.104.7 Errors

Except Generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The discovery task id does not exist.
TASK_IS_RUNNING	A running task cannot be deleted.

8.105 Add a batch add task

8.105.1 Request URL

```
/rest/batchAddTasks
```

8.105.2 HTTP request method

POST

8.105.3 Request example

```
{
  "devices": [
    {
      "name": "device1",
      "address": "192.168.1.1",
      "userName": "user1",
      "password": "password1",
      "deratedPower": 400,
      "protocol": "IPMI",
      "size": 1,
      "location": 1,
      "description": "device1",
      "birthday": "2019-01-17",
      "weight": 100,
      "rack": "rack1",
      "row": "row1",
      "room": "room1",
      "dc": "dc1",
      "vendor": "ANY"
    }
  ]
}
```

8.105.4 Request field description

Field	Type	Attribute	Description
devices	Array	Required	The devices with hierarchy information to be added.
protocol	String	Optional	The possible values: IPMI SSH HTTPS SNMPv1v2c SNMPv3 INBAND_PROTOCOL WS_MAN Unmanaged server It is used to manage the servers without interfaces to monitor power and thermal. Unmanaged network device

			Unmanaged storage device Unmanaged chassis
Name	String	Optional	The name of the device.
address	String	Optional	The IP address or host name of the device. It is useless when the type is unmanaged device.
userName	String	Optional	The user name to connect to the device.
password	String	Optional	The password to connect to the device.
key	String	Optional	The BMC key which is useful only when the type is IPMI.
deratedPower	Integer	Optional	The derated power for none-chassis devices such as servers, PDU, UPS and so on. The unit is W.
namePlatePower	Integer	Optional	The nameplate power of the device.
snmpCommunityString	String	Optional	SNMP community string which is useful only when the protocol is SNMPV1v2c.
snmpEncryptionPassword	String	Optional	SNMP encryption password which is useful only when the protocol is SNMPv3.
snmpAuthenticationProtocol	String	Optional	SNMP authentication protocol which is useful only when the protocol is SNMPv3. Possible values: AuthMD5 AuthSHA1 AuthHMAC128SHA224 AuthHMAC192SHA256 AuthHMAC256SHA384 AuthHMAC384SHA512
snmpEncryptionProtocol	String	Optional	SNMP encryption protocol which is useful only when the type is SNMPv3. Possible values: PrivDES Priv3DES PrivAES128 PrivAES192 PrivAES256
httpsPort	Integer	Optional	The HTTPs port. The default value is 443.
sshPort	Integer	Optional	The SSH port. The default value is 22.
authenticateEntity	Boolean	Optional	Whether it is need to validate the device. For https, the device certificate needs to be imported into DCM key store if the field value is true. By default, the value is true for most protocols.
size	Integer	Optional	The size of the device. For servers, the default value is 1. For chassis, the default value is 10. The unit is U.
location	Integer	Optional	The slot of the device.
typicalPower	Integer	Optional	The typical power of the device which is useful only when the type is unmanaged device.
idlePower	Integer	Optional	The idle power of the device which is useful only when the type is unmanaged device.
peakPower	Integer	Optional	The peak power of the device which is useful only when the type is unmanaged device.
model	String	Optional	The device model of the device which is useful only when the type is unmanaged device.
osType	String	Optional	The in-band OS type. The possible values: Windows Linux Xen

			ESX
osAddress	String	Optional	The OS IP address or host name which is used to specify the in-band OS information when the protocol is IPMI.
osUserName	String	Optional	The user name to access the OS which is used to specify the in-band OS information when the protocol is IPMI.
osPassword	String	Optional	The user password to access the OS which is used to specify the in-band OS information when the protocol is IPMI.
osSshPort	Integer	Optional	The in-band OS SSH port with the default value 22. It is used to specify the in-band OS information when the protocol is IPMI.
description	String	Optional	Description for the device.
reportInletTemp	Boolean	Optional	Whether to report inlet temperature. Currently it is useful only for chatsworth PDU.
birthday	String	Optional	The birthday of the device. The format can be "yyyy-MM-dd" or "yyyy-MM-dd'T'HH:mm:ssZ". For examples, "2018-12-16" or "2018-12-16T00:00:00+08:00".
customConsoleURL	String	Optional	The specified device management console URL.
width	Integer	Optional	The horizontal grid count of a chassis or a blade which is used to locate blades in a chassis.
height	Integer	Optional	The vertical grid count of a chassis or a blade. which is used to locate blades in a chassis.
startX	Integer	Optional	The horizontal location of a blade in a chassis.
startY	Integer	Optional	The vertical location of a blade in a chassis.
epr	String	Optional	The action of a device with emergency power reduction. Possible values: Minimize Power Consumption Shutdown No Action
weight	Double	Optional	Device weight. The unit is KG.
platformId	String	Optional	The id which is used to identify the device. It is only used for unmanaged devices when adding unmanaged devices. Unmanaged devices include: Unmanaged server Unmanaged network device Unmanaged storage device Unmanaged chassis Platform ID can only be specified when adding an unmanaged device and cannot be modified. It will be converted to upper case by DCM. When adding a managed device DCM will specify it internally.
provisioningUser	String	Optional	The provisioning user of the device which must be an administrator.
customAttributes	Array	Optional	Custom attribute array. Each element contains a "customAttributeName" field and a "customAttributeValue" field.
customAttributeName	String	Optional	The name of the custom attribute. When the object exists in the "customAttributes" array, the field is required. The field can be specified not only for the custom attributes defined by the end-users, but also for some built-in attributes:

			Owner Business Unit Contact Warranty Expiration Vendor Email Except the built-in attributes, the custom attribute should exist before its value is specified. Refer to the chapter "Add a custom attribute" for details.
customAttributeValue	String	Optional	The value of the custom attribute. For the date type attribute, the format is "yyyy-MM-dd". For example, "2020-10-11".
rack	String	Optional	The name of the rack to add the device.
row	String	Optional	The name of the row to add the device.
room	String	Optional	The name of the room to add the device.
dc	String	Optional	The name of the datacenter to add the device.
enclosure	String	Optional	The name of the enclosure to add the blade.
rackCapacity	Integer	Optional	The rack space capacity.
rackPowerCapacity	Integer	Optional	The rack power capacity.
dcPowerCapacity	Integer	Optional	The datacenter power capacity.
dcElectricityRate	Double	Optional	The datacenter electricity rate.
roomPowerCapacity	Integer	Optional	The room power capacity.
rowPowerCapacity	Integer	Optional	The row power capacity.
gridX	Integer	Optional	The grid x coordinate of the rack.
gridY	Integer	Optional	he grid y coordinate of the rack.
vendor	String	Optional	The vendor name. If any vendor is specified, DCM will add the device only if the identified devices match the known vendor. If not specified or "ANY" specified, all known vendors are supported. "OTHERS" means all other vendors but the enumerated vendors. The possible values: ANY, ALTOS, AMD, APC, ARISTA, AVOCENT, BAYTECH, BROCADE, CHATSWORTH, CISCO, CONTOSO, DELL, EATON, EMC, EMERSON, ENLOGIC, EXTREMENETWORKS, F5, FUJITSU,

			H3C, HITACHI, HOFFMAN, HP, HUAWEI, IBM, INSPUR, INTEL, KONTRON, LENOVO, MELLANOX, NETAPP, NETTRIX, ORACLE, RARITAN, RSA, SERVERTECH, SIEMON, SUGON, SUPERMICRO, VERTIV, ZTE, OTHERS

***Note**:*

- *When a device exists, it will be updated.*
- *If the hierarchy (i.e., dc/room/row/rack) does not exist, it will be created. Or else it will be updated.*
- *If the field "name" and "protocol" of a device are not specified, only hierarchy will be created or updated. Specifying only one of the fields is invalid.*

8.105.5 Response example


```
{
    "taskId": "1",
    "createDate": "2019-01-17T10:00:00+08:00",
    "results": [
        {
            "deviceId": 10,
            "name": "device1",
            "address": "192.1681.1",
            "result": "OK",
            "failedMsg": ""
        }
    ],
    "progress": "0",
    "timeElapsed": "0",
    "links": [
        {
            "rel": "self",
            "href": "https://localhost:8643/DcmConsole/rest/batchAddTasks/1"
        },
        {
            "rel": "delete",
            "href": "https://localhost:8643/DcmConsole/rest/batchAddTasks/1"
        },
        {
            "rel": "control",
            "href": "https://localhost:8643/DcmConsole/rest/batchAddTasks/1"
        }
```

```
    ]
}
```

8.105.6 Response field description

Field	Type	Description
taskId	Integer	The id of the task.
createDate	String	The creation time of the task. The format is "yyyy-MM-dd'T'HH:mm:ssXXX". For example, 2019-08-02T11:00:00+08:00.
results	Array	The array contains the result of all devices to add.
deviceId	Integer	The id of the added device.
result	String	If the task is not finished, the field will be empty. Possible values: OK Failed
failedMsg	String	Sometimes it will give the failed message if the result failed.
progress	Integer	The value is from 0 to 100. The value "100" means the task is finished.
timeElapsed	Integer	The time spent since the task runs. The unit is second.
endDate	String	The end time of the task. The format is "yyyy-MM-dd'T'HH:mm:ssXXX". For example, 2019-08-02T11:00:00+08:00. This field will not be presented if the task is not finished.

Refer to the chapter "Request field description" for other field definition.

8.105.7 Errors

Except Generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
INVALID_PARAMETER	Invalid parameter. The array of the devices is empty.

8.106 Get batch add tasks

8.106.1 Request URL

```
/rest/batchAddTasks
```

8.106.2 HTTP request method

GET

8.106.3 Request example

https://localhost:8643/DcmConsole/rest/batchAddTasks

8.106.4 Request field description

N/A.

8.106.5 Response example

```
{
  "links": [
    {
      "rel": "self",
      "href": "https://localhost:8643/DcmConsole/rest/batchAddTasks"
    },
    {
```

```
            "rel": "add",
            "href": "https://localhost:8643/DcmConsole/rest/batchAddTasks"
          }
        ],
        "content": [
          {
            "taskId": "1",
            "createDate": "2019-01-17T10:00:00+08:00",
            "results": [
              {
                "deviceId": 10,
                "name": "device1",
                "address": "192.1681.1",
                "result": "OK",
                "failedMsg": ""
              }
            ],
            "progress": "0",
            "timeElapsed": "0",
            "links": [
              {
                "rel": "self",
                "href": "https://localhost:8643/DcmConsole/rest/batchAddTasks/1"
              },
              {
                "rel": "delete",
                "href": "https://localhost:8643/DcmConsole/rest/batchAddTasks/1"
              },
              {
                "rel": "control",
                "href": "https://localhost:8643/DcmConsole/rest/batchAddTasks/1"
              }
            ]
          }
        ]
      }
```


8.106.6 Response field description

Refer to the chapter "Add a batch add task".

8.106.7 Errors

Generic errors defined in chapter "Error handling".

8.107 Get a batch add task

8.107.1 Request URL

/rest/batchAddTasks/{taskId}

The parameter "taskId" is the ID of the task.

8.107.2 HTTP request method

GET

8.107.3 Request example

https://localhost:8643/DcmConsole/rest/batchAddTask/1

8.107.4 Request field description

N/A.

8.107.5 Response example

Refer to the chapter "Add a batch add task".

8.107.6 Response field description

Refer to the chapter "Add a batch task".

8.107.7 Errors

Except Generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The target does not exist.

8.108 Control a batch add task

8.108.1 Request URL

```
/rest/batchAddTasks/{taskId}/control
```

The parameter "taskId" is the ID of the task.

8.108.2 HTTP request method

POST

8.108.3 Request example

```
{
  "controlAction": "STOP"
}
```

8.108.4 Request field description

Field	Type	Attribute	Description
controlAction	String	Required	Possible values: STOP Stop a running task.

8.108.5 Response example

Refer to the chapter "Add a batch add task".

8.108.6 Response field description

Refer to the chapter "Add a batch add task" for field definitions.

8.108.7 Errors

Except the generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The task Id does not exist.
CONTROL_ACTION_IS REQUIRED	Control action is required.
INVALID_CONTROL_ACTION	Invalid control action value.

8.109 Delete a batch add task

8.109.1 Request URL

```
/rest/batchAddTasks/{taskId}
```

The parameter "taskId" is the ID of the task.

8.109.2 HTTP request method

DELETE

8.109.3 Request example

https://localhost:8643/DcmConsole/rest/batchAddTasks/1

8.109.4 Request field description

N/A.

8.109.5 Response example

N/A.

8.109.6 Response field description

N/A.

8.109.7 Errors

Except Generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The task id does not exist.
TASK_IS_RUNNING	A running task cannot be deleted.

8.110 Update/delete entities

8.110.1 Request URL

```
/rest/entities
```

8.110.2 HTTP request method

PUT

8.110.3 Request example

```
{
  "action": "UPDATE",
  "ids": [
    1,
    2
  ],
  "parentId": 1
}
```

8.110.4 Request field description

Field	Type	Attribute	Description
Action	String	Required	Possible values: UPDATE Update the information such as the parent id of the entities. DELETE Delete all entities specified in the field "ids".
Ids	Array	Required	The id array of entities to input a batch id of datacenters, rooms, rows, or devices. The entity type should be same.
parentId	Integer	Optional	The id of the parent entity. To move a batch of devices to a rack, specify the id of the rack as the value of this field.

			If this field is not present, nothing will be done at the moment.

8.110.5 Response example

N/A.

8.110.6 Response field description

N/A.

8.110.7 Errors

Except the generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The id array is missing, or the id it contains did not exist.
ENTITY_TYPES_ARE_NOT_SAME	The entity types should be same.
INVALID_PARENT_ID	The parent id is invalid. For example, specifying the id of a row as parent id is invalid.
ACTION_IS REQUIRED	Action is required.
INVALID_ACTION	Invalid action value.

8.111 Get unhealthy SSDs

8.111.1 Request URL

/rest/analysis/unhealthySSDs

8.111.2 HTTP request method

GET

8.111.3 Request example

https://localhost:8643/DcmConsole/rest/analysis/unhealthySSDs

8.111.4 Request field description

N/A.

8.111.5 Response example


```
{
  "links": [
    {
      "rel": "self",
      "href": "https://localhost:8643/DcmConsole/rest/analysis/unhealthySSDs"
    }
  ],
  "content": [
    {
      "SSDSerialNumber": "CVLI315201J8120B",
      "SSDModel": "INTEL SSDMCEAC120A3",
      "SSDFirmwareVersion": "LLGi",
      "SSDUserCapacity": "120,034,123,776 bytes [120 GB]",
      "deviceId": "1",
      "unhealhyAttributes": [
        {
          "attributeName": "Reported_Uncorrect",
          "attributeValue": "40",
          "attributeRawValue": "100",
          "attributeThreshold": "50"
        }
```

```
        ]
      }
    ]
}
```

8.111.6 Response field description

Field	Type	Description
SSDSerialNumber	String	The serial number of the SSD.
SSDModel	String	The model of the SSD.
SSDFirmwareVersion	String	The firmware version of the SSD.
deviceId	Integer	The server id.
unhealhyAttributes	Array	The unhealthy attribute list.
attributeName	String	The smart attribute name.
attributeValue	Integer	The smart attribute value.
attributeRawValue	Integer	The smart attribute raw value.
attributeThreshold	Integer	The smart attribute threshold.

8.111.7 Errors

Generic errors defined in chapter "Error handling".

8.112 Get unhealthy devices

8.112.1 Request URL

```
/rest/analysis/unhealthyDevices?ancestorId={id}
```

The parameter "ancestorId" is the ID of the target group. It's optional.

8.112.2 HTTP request method

GET

8.112.3 Request example

https://localhost:8643/DcmConsole/rest/analysis/unhealthyDevices?ancestorId=8

8.112.4 Request field description

N/A.

8.112.5 Response example

```
{
  "links": [
    {
      "rel": "self",
      "href":
"https://localhost:8643/DcmConsole/rest/analysis/unhealthyDevices?ancestorId=8"
    }
  ],
  "content": [
    {
      "deviceId": "5",
      "name": "server-1",
      "entityPath": [
```

```
            "4",
            "3",
            "2",
            "1"
          ],
          "entityPathStr": "dc > room > row > rack",
          "deviceType": "Server",
          "serialNumber": "CNG83210D5",
          "description": "server 1",
          "componentHealth": [
            {
              "component": "SYSTEM",
              "status": "FAULT",
              "description": "Not OK"
            },
            {
              "component": "STORAGE",
              "status": "FAULT",
              "description": "Not OK"
            }
          ]
        },
        {
          "deviceId": "6",
          "name": "server-2",
          "entityPath": [
            "4",
            "3",
            "2",
            "1"
          ],
          "entityPathStr": "dc > room > row > rack",
          "deviceType": "Server",
          "serialNumber": "CNG83210D6",
          "description": "server 2",
          "componentHealth": [
            {
              "component": "SYSTEM",
              "status": "FAULT",
              "description": "Not OK"
            },
            {
              "component": "STORAGE",
              "status": "FAULT",
              "description": "Not OK"
            }
          ]
        }
      ]
    }
```


8.112.6 Response field description

Field	Type	Description
deviceId	Integer	The id of the device.
Name	String	The name of the device.
entityPath	Array	The parent id array which will be in sequence of rack id, row id, room id and datacenter id.
entityPathStr	Integer	The path in the hierarchy which will be like "dc > room > row > rack".
deviceType	String	Possible values:

		Server Chassis PDU UPS Storage device Network device
serialNumber	String	The serial number read from device.
description	String	The description of the device.
componentHealth	Array	The component health information.

Refer to the chapter "Get health status of a device" for other field definitions.

8.112.7 Errors

Generic errors defined in chapter "Error handling".

8.113 Add a notification

8.113.1 Request URL

```
/rest/event/notifications
```

8.113.2 HTTP request method

POST

8.113.3 Request example

```
{
  "targetId": "1",
  "eventType": "IT_EQPMNT_PWR",
  "condition": "GREATER_THAN",
  "threshold": "100",
  "description": ""
}
```

8.113.4 Request field description

Field	Type	Attribute	Description
targetId	Integer	Required	The id of a datacenter, a room, a row, a device, or a group. If eventType is CARBON_EMISSION_ANNUAL or CARBON_EMISSION_MONTH, it should be the id of a datacenter, all datacenters, or a group. -4 means all datacenters.
eventType	String	Required	The event type. Possible values: MAX_INLET_TEMP Maximum inlet temperature. The unit is Celsius. AVG_INLET_TEMP Average inlet temperature. The unit is Celsius. MIN_INLET_TEMP Minimum inlet temperature. The unit is Celsius. IT_EQPMNT_PWR IT equipment power. HUMIDITY Only applicable for rooms and devices with humidity monitoring capability. CARBON_EMISSION_ANNUAL Annual carbon emission CARBON_EMISSION_MONTH Monthly carbon emission

condition	String	Required	The condition to trigger the notification. Possible value: GREATER_THAN LESS_THAN
threshold	Integer	Required	The threshold.
description	String	Optional	Description.

***Note**: Only the notification with the same "targetId", "eventType" and "condition" can be added. The old one will be updated after receiving a new request.*

8.113.5 Response example


```
{
  "id": "1",
  "targetId": "1",
  "eventType": "IT_EQPMNT_PWR",
  "condition": "GREATER_THAN",
  "threshold": "100",
  "description": "",
  "links": [
    {
      "rel": "self",
      "href": "https://localhost:8643/DcmConsole/rest/event/notifications/1"
    },
    {
      "rel": "update",
      "href": "https://localhost:8643/DcmConsole/rest/event/notifications/1"
    },
    {
      "rel": "delete",
      "href": "https://localhost:8643/DcmConsole/rest/event/notifications/1"
    }
  ]
}
```


8.113.6 Response field description

Field	Type	Description
id	Integer	The id of the notification.

Refer to the chapter "Request field description" for other field definition.

8.113.7 Errors

Except the generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The target id does not exist or is not supported.
NOTIFICATION_NOT_SUPPORT	The target does not support to add notification.
EVENT_TYPE_IS_REQUIRED	Event type is required.
INVALID_EVENT_TYPE	Invalid event type.
CONDITION_IS_REQUIRED	Condition is required.
INVALID_CONDITION	Invalid condition.
THRESHOLD_IS_REQUIRED	Threshold is required.
INVALID_THRESHOLD	Invalid threshold value.

8.114 Update a notification

8.114.1 Request URL

/rest/event/notifications/{notificationId}

Note that only "threshold" and "description" of a notification can be updated. Other fields will be ignored.

8.114.2 HTTP request method

PUT

8.114.3 Request example

```
{
  "notificationId": "1",
  "threshold": "100",
  "description": ""
}
```

8.114.4 Request field description

Refer to chapter "Add a notification".

8.114.5 Response example

```
{
    "notificationId": "1",
    "targetId": "1",
    "eventType": "IT_EQPMNT_PWR",
    "condition": "GREATER_THAN",
    "threshold": "100",
    "description": "",
    "links": [
        {
            "rel": "self",
            "href":
"https://localhost:8643/DcmConsole/rest/event/notifications/1"
        },
        {
            "rel": "update",
            "href":
"https://localhost:8643/DcmConsole/rest/event/notifications/1"
        },
        {
            "rel": "delete",
            "href":
"https://localhost:8643/DcmConsole/rest/event/notifications/1"
        }
    ]
}
```

8.114.6 Response field description

Refer to the chapter "Request field description" for other field definition.

8.114.7 Errors

Except the generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The notification Id does not exist.
INVALID_THRESHOLD	Invalid threshold value.

8.115 Get notifications

8.115.1 Request URL

/rest/event/notifications

8.115.2 HTTP request method

GET

8.115.3 Request example

https://localhost:8643/DcmConsole/rest/event/notifications?targetId=-1

8.115.4 Request field description

Field	Type	Attribute	Description
targetId	Integer	Required	The id of a datacenter, a room, a row, a device, or a group. If this parameter is not present or the values is -1, all notifications will be returned.

8.115.5 Response example


```
{
  "links": [
    {
      "rel": "self",
      "href":
"https://localhost:8643/DcmConsole/rest/event/notifications?targetId=-1"
    },
    {
      "rel": "add",
      "href": "https://localhost:8643/DcmConsole/rest/event/notifications"
    }
  ],
  "content": [
    {
      "notificationId": "1",
      "targetId": "1",
      "eventType": "IT_EQPMNT_PWR",
      "condition": "GREATER_THAN",
      "threshold": "100",
      "description": "",
      "links": [
        {
          "rel": "self",
          "href": "https://localhost:8643/DcmConsole/rest/event/notifications/1"
        },
        {
          "rel": "update",
          "href": "https://localhost:8643/DcmConsole/rest/event/notifications/1"
        },
        {
          "rel": "delete",
          "href": "https://localhost:8643/DcmConsole/rest/event/notifications/1"
        }
      ]
    }
  ]
}
```


8.115.6 Response field description

Refer to the chapter "Add a notification".

8.115.7 Errors

Generic errors defined in chapter "Error handling".

8.116 Get a notification

8.116.1 Request URL

/rest/event/notifications/{notificationId}

The parameter "notificationId" is the ID of the notification.

8.116.2 HTTP request method

GET

8.116.3 Request example

https://localhost:8643/DcmConsole/rest/event/notifications/1

8.116.4 Request field description

N/A.

8.116.5 Response example

Refer to the chapter "Add a notification".

8.116.6 Response field description

Refer to the chapter "Add a notification".

8.116.7 Errors

Except Generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The notification id does not exist.

8.117 Delete a notification

8.117.1 Request URL

/rest/event/notifications/{notificationId}

The parameter "notificationId" is the ID of the notification.

8.117.2 HTTP request method

DELETE

8.117.3 Request example

https://localhost:8643/DcmConsole/rest/event/notifications/1

8.117.4 Request field description

N/A.

8.117.5 Response example

N/A.

8.117.6 Response field description

N/A.

8.117.7 Errors

Except Generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description

TARGET_NOT_EXISTED	The notification id does not exist.

8.118 Add an event handler

8.118.1 Request URL

/rest/event/eventHandlers

8.118.2 HTTP request method

POST

8.118.3 Request example

```
{
  "handlerURL": "tcp://127.0.0.1:6553"
}
```

8.118.4 Request field description

Field	Type	Attribute	Description
handlerURL	String	Required	A universal resource locator (URL) that identifies the event handler endpoint which could be a network address or a logical address. The URL protocol must be http:// or tcp://. To receive events from DCM, you need to add an event handler to accept connections and receive messages by using the handler URL.

***Note**: DCM supports both short and long connection mode to send events. The mode can be changed by a dedicated configuration "EVENT_USING_TCP_LONG_CONNECTION" in file "conf/user.config.xml" under the installation folder. The default value is "False".*

8.118.5 Response example

```
{
  "id": "1",
  "handlerURL": "tcp://127.0.0.1:6553",
  "links": [
    {
      "rel": "self",
      "href": "https://localhost:8643/DcmConsole/rest/event/eventHandlers/1"
    },
    {
      "rel": "delete",
      "href": "https://localhost:8643/DcmConsole/rest/event/eventHandlers/1"
    }
  ]
}
```

8.118.6 Response field description

Field	Type	Description
id	Integer	The id of the event handler.

Refer to the chapter "Request field description" for other field definition.

8.118.7 Errors

Except the generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
HANDLER_URL_IS_REQUIRED	Handler URL is required.

INVALID_URL	URL is invalid.

8.119 Get event handlers

8.119.1 Request URL

/rest/event/eventHandlers

8.119.2 HTTP request method

GET

8.119.3 Request example

https://localhost:8643/DcmConsole/rest/event/eventHandlers

8.119.4 Request field description

N/A.

8.119.5 Response example

```
{
  "links": [
    {
      "rel": "self",
      "href": "https://localhost:8643/DcmConsole/rest/event/eventHandlers"
    },
    {
      "rel": "add",
      "href": "https://localhost:8643/DcmConsole/rest/event/eventHandlers"
    }
  ],
  "content": [
    {
      "id": "1",
      "handlerURL": "tcp://127.0.0.1:6553",
      "links": [
        {
          "rel": "self",
          "href": "https://localhost:8643/DcmConsole/rest/event/eventHandlers/1"
        },
        {
          "rel": "delete",
          "href": "https://localhost:8643/DcmConsole/rest/event/eventHandlers/1"
        }
      ]
    }
  ]
}
```

8.119.6 Response field description

Refer to the chapter "Add an event handler".

8.119.7 Errors

Generic errors defined in chapter "Error handling".

8.120 Get an event handler

8.120.1 Request URL

/rest/event/eventHandlers/{eventHandlerId}

The parameter "eventHandlerId" is the ID of the event handler.

8.120.2 HTTP request method

GET

8.120.3 Request example

https://localhost:8643/DcmConsole/rest/event/eventHandlers/1

8.120.4 Request field description

N/A.

8.120.5 Response example

Refer to the chapter "Add an event handler".

8.120.6 Response field description

Refer to the chapter "Add an event handler".

8.120.7 Errors

Except Generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The event handler id does not exist.

8.121 Delete an event handler

8.121.1 Request URL

/rest/event/eventHandlers/{eventHandlerId}

The parameter "eventHandlerId" is the ID of the event handler.

8.121.2 HTTP request method

DELETE

8.121.3 Request example

https://localhost:8643/DcmConsole/rest/event/eventHandlers/1

8.121.4 Request field description

N/A.

8.121.5 Response example

N/A.

8.121.6 Response field description

N/A.

8.121.7 Errors

Except Generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The event handler id does not exist.

8.122 Get historical events

8.122.1 Request URL

/rest/event/historicalEvents

8.122.2 HTTP request method

GET

8.122.3 Request example

https://localhost:8643/DcmConsole/rest/event/historicalEvents?entityId=80&&severity=CRITICAL|ERROR&&startDateTime=2019-01-17T10:00%2B08:00&&endDateTime=2019-01-17T20:00%2B08:00&&page=0&&itemsPerPage=2&&sortField=timestamp&&sortOrder=desc

8.122.4 Request field description

Field	Type	Attribute	Description
entityId	Integer	Optional	The events related entity id. When the entity is a group, all events related to the group and the sub entities under the group will be returned. If the entity is a server, all events related to this device will be returned. The default value is -1 which means the events are related to all entities.
severity	String	Optional	The severity strings. The possible values of the severity: ALL CRITICAL ERROR WARNING INFORMATIVE CUSTOM Event categories can be specified as a prefix before the severity. The format is "eventCategory:severity". For example, "PREDEFINED_EVENT:CRITICAL, ERROR". Multiple severity strings can be linked by character "|". For example, "PREDEFINED_EVENT:CRITICAL,ERROR|NOTIFICATION:ALL". The possible event categories: PREDEFINED_EVENT NOTIFICATION If this field is not present, all events will be returned.
startDateTime	String	required	The start date time. The format is "yyyy-MM-dd'T'HH:mm:ssXXX".
endDateTime	String	Optional	The end date time. NULL value means the current time. If the parameter is not present, current time will be used.
page	Integer	Optional	The result supports pagination. This parameter means the page number starting from 0. The default value is 0.
itemsPerPage	Integer	Optional	The number of items in one page. The default value is 10.
sortField	String	Optional	The possible values: severity entityName assetTag eventType description timestamp The default value is "timestamp".
sortOrder		Optional	The possible values: asc desc

			The default value is "desc".

8.122.5 Response example

```
{
  "totalItems": 2,
  "itemsPerPage": 2,
  "page": 0,
  "dataList": [
    {
      "eventId": "1",
      "entityId": "10",
      "entityName": "server-1",
      "assetTag": "MX84012",
      "description": "Status: Abnormal; Name: SYSTEM; Detail: ERROR",
      "severity": "WARNING",
      "eventType": "DEVICE_COMPONENT_FAULT",
      "eventCategory": "PREDEFINED_EVENT",
      "timestamp": "2019-01-17T10:00:00+08:00"
    },
    {
      "eventId": "2",
      "entityId": "10",
      "entityName": "server-1",
      "assetTag": "MX84012",
      "description": "The current IT Equipment Power is 1200 W which is greater
than the threshold of 10 W.",
      "severity": "CRITICAL",
      "eventType": "IT_EQPMNT_PWR",
      "eventCategory": "NOTIFICATION",
      "timestamp": "2019-01-17T10:00:00+08:00"
    }
  ]
}
```

8.122.6 Response field description

Field	Type	Description
totalItems	Integer	The total items when pagination parameters are specified in the request.
itemsPerPage	Integer	The item count per page.
page	Integer	The current page.
dataList	Array	The item list.
eventId	Integer	The id of the event.
entityId	Integer	The event related entity.
entityName	String	The name of the entity.
assetTag	String	The serial number read from the device.
description	String	The event description.
severity	String	The event severity. Refer to the request field description.
eventType	String	The event type. The description field gives the detailed information.
eventCategory	String	The event category. Refer to the request field description.
timestamp	String	The time that the event is reported.

8.122.7 Errors

Except the generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The entity id does not exist.
INVALID_SEVERITY	The severity string is invalid.
START_TIME_IS_REQUIRED	Start time is required.
START_TIME_BEFORE_END_TIME	Start time is not before the end time.
INVALID_PAGE	The page value is invalid.
INVALID_ITEMS_PER_PAGE	Invalid item number per page.
INVALID_PARAMETER	Invalid parameter. Typically, the time formation is incorrect.
INVALID_SORT_FIELD	Invalid sort field.
INVALID_SORT_ORDER	Invalid sort order.

8.123 Get measurement data of an entity

8.123.1 Request URL

/rest/data/measurementData

8.123.2 HTTP request method

GET

8.123.3 Request example

https://localhost:8643/DcmConsole/rest/data/measurementData?entityId=1&&measurementType=MAX_INLET_TEMP&&startDateTime=2019-01-17T10:00%2B08:00&&endDateTime=2019-01-17T10:00%2B08:00&&granularity=monitoringPeriod&&page=0&&itemsPerPage=10

8.123.4 Request field description

Field	Type	Attribute	Description
entityId	Integer	Required	The id of the entity. When measurementType is CARBON_EMISSION, -4 means all datacenters.
measurementType	String	Required	Possible values: MAX_PWR The maximum power consumed by any single node/enclosure within the specified entity. AVG_PWR The average power consumption across all nodes/enclosures within the specified entity. MIN_PWR The minimum power consumed by any single node/enclosure within the specified entity. MAX_CPU_PWR The average CPU power consumption of a node within the specified period. MIN_CPU_PWR The minimum CPU power consumption of a node within the specified period. MAX_MEM_PWR The maximum memory power consumption of a node within the specified period. AVG_MEM_PWR

			The average memory power consumption of a node within the specified period. MIN_MEM_PWR The minimum memory power consumption of a node within the specified period. MAX_AVG_PWR The maximum group sampling (in a monitoring cycle) power in specified aggregation period for the sum of average power measurement in a group of nodes/enclosures within the specified entity. MIC_PWR The average power consumption of the Many Integrated Core (MIC) subsystem of the node within the specified period. TOTAL_MAX_PWR The maximum group sampling (in a monitoring cycle) power in specified aggregation period for sum of maximum power measurement in a group of nodes/enclosures within the specified entity. TOTAL_AVG_PWR The average (in specified aggregation period) group power for sum of average power measurement in a group of nodes/enclosures within the specified entity. MAX_AVG_PWR_CAP The maximum group sampling (in a monitoring cycle) power in specified aggregation period for the sum of average power measurement in a group of nodes with power capping capability. TOTAL_MAX_PWR_CAP The maximum group sampling (in a monitoring cycle) power in specified aggregation period for sum of maximum power measurement in a group of nodes with power capping capability. TOTAL_AVG_PWR_CAP The average (in specified aggregation period) group power for sum of average power measurement in a group of nodes with power capping capability. TOTAL_MIN_PWR The minimal group sampling (in a monitoring cycle) power in specified aggregation period for sum of minimum power measurement in a group of nodes/enclosures within the specified entity. MIN_AVG_PWR The minimal group sampling (in a monitoring cycle) power in specified aggregation period for sum of average power measurement in a group of nodes/enclosures within the specified entity. MAX_INLET_TEMP The maximum temperature for any single node within the specified entity. AVG_INLET_TEMP The average temperature for any single node within the specified entity. MIN_INLET_TEMP The minimum temperature for any single node within the specified entity. INS_PWR The instantaneous power. PDU_PWR The aggregated instantaneous power for PDU or PDUs in the group. OBSV_MAX_PDU_PWR

			The maximum value of PDU_PWR of the entity observed by DCM within each of the aggregation periods. CPU_USED The utilization of the physical CPU cores. MEMORY_USED The utilization of the physical memory bandwidth. IO_USED The utilization of the physical I/O bandwidth. AVG_OUTLET_TEMP The weighted average outlet temperature with respect to airflow for any single node within the specified entity. AIRFLOW The average airflow going through the node or all nodes in the group. CPU_UTIL The CPU Utilization data for nodes scaled with %. DISK_IO The Disk IO data for nodes in the unit of Bytes/Second. HUMIDITY The humidity data for PDUs with the corresponding humidity sensors MNGED_NODES_ENERGY The total energy consumed by all managed nodes in the specified entity, in Wh. MNGED_NODES_ENERGY_BILL The total power bill for all energy consumed by all managed nodes in the specified entity. The cost per kWh is defined by COST_PER_KW_HR. IT_EQPMNT_ENERGY The total energy consumed by IT equipment, in Wh. IT_EQPMNT_ENERGY_BILL The calculated power bill for IT equipment. The cost per kWh is defined by COST_PER_KW_HR. CALC_COOLING_ENERGY The energy needed to cool the selected entity, in Wh. CALC_COOLING_ENERGY_BILL The calculated power bill for the energy needed to cool the selected entity. MNGED_NODES_PWR The total average power consumption by the managed nodes in the selected entity, in watts. IT_EQPMNT_PWR An estimation of power consumption by IT equipments. CALC_COOLING_PWR The average cooling power based on the IT_EQPMNT_PWR multiplie by COOLING_MULT in watts. CARBON_EMISSION The carbon emission data of the related entity.
startDateTime	String	Required	The start date time.
endDateTime	String	Optional	The end date time. NULL value means the current time. The default value is the current time.
page	Integer	Optional	The result supports pagination. This parameter is the page number starting from 0. The default value is 0.
itemsPerPage	Integer	Optional	The number of items in one page. The default value is 10.
granularity	String	Optional	The possible values: monitoringPeriod The current monitoring interval. For example, if the current monitoring period is 3 minutes, the granularity will be 3 minutes.

			1hour 1 hour. 6hours 6 hours. 24hours 24 hours. 7 days 7 days. 14days 14 days. The default value is "monitoringPeriod". By default, if the time range exceeds 1 week, the minimal granularity should be 1 hour.

8.123.5 Response example

```
{
  "totalItems": 121,
  "itemsPerPage": 2,
  "page": 0,
  "dataList": [
    {
      "value": 2816,
      "dateTime": "2019-01-17T10:00:00+08:00"
    },
    {
      "value": 2820,
      "dateTime": "2019-01-17T10:03:00+08:00"
    }
  ]
}
```

8.123.6 Response field description

Field	Type	Description
totalItems	Integer	The total items when pagination parameters are specified in the request.
itemsPerPage	Integer	The item count per page.
page	Integer	The current page.
dataList	Array	The item list.
dateTime	String	The timestamp of the data. The format is "yyyy-MM-dd'T'HH:mm:ssXXX".
value	Integer	The value of the data.

8.123.7 Errors

Except the generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The entity id does not exist.
QUERY_TYPE_IS_REQUIRED	Measurement type is required.
INVALID_QUERY_TYPE	Invalid measurement type.
INVALID_PAGE	The page value is invalid.
INVALID_ITEMS_PER_PAGE	Invalid item number per page.

INVALID_GRANULARITY	Invalid granularity.
START_TIME_IS_REQUIRED	Start time is missing or the value is invalid.
START_TIME_BEFORE_END_TIME	Start time is not before the end time.
INVALID_PARAMETER	Invalid parameter. Typically, the time format is incorrect.

8.124 Get latest measurement data of entities

8.124.1 Request URL

/rest/data/latestMeasurementData

8.124.2 HTTP request method

POST.
Note that POST instead of GET is used here to avoid URL length limitation, and HTTP status code 200 will be returned.

8.124.3 Request example

```
{
  "ids": [
    "1",
    "2"
  ],
  "measurementTypes": [
    "AVG_PWR",
    "IT_EQPMNT_PWR"
  ]
}
```

8.124.4 Request field description

Field	Type	Attribute	Description
ids	Array	Required	The id array of the entities. Negative value of the entity id has special usages: "-1" means all datacenters "-2" means all rooms "-3" means all rows "-4" means all racks "-5" means all devices "-6" means all logical groups The union of the specified entity id will be used. The negative values can be specified with normal entity ids.
measurementTypes	Array	Required	Possible values in the array: MAX_PWR The maximum power consumed by any single node/enclosure within the specified entity. AVG_PWR The average power consumption across all nodes/enclosures within the specified entity. MIN_PWR The minimum power consumed by any single node/enclosure within the specified entity. AVG_CPU_PWR The average CPU power consumption of a node within the specified period. MAX_CPU_PWR The maximum CPU power consumption of a node within the specified period.

			MIN_CPU_PWR The minimum CPU power consumption of a node within the specified period. MAX_MEM_PWR The maximum memory power consumption of a node within the specified period. AVG_MEM_PWR The average memory power consumption of a node within the specified period. MIN_MEM_PWR The minimum memory power consumption of a node within the specified period. MIC_PWR The average power consumption of the Many Integrated Core (MIC) subsystem of the node within the specified period. TOTAL_MAX_PWR The maximum group sampling (in a monitoring cycle) power in specified aggregation period for sum of maximum power measurement in a group of nodes/enclosures within the specified entity. TOTAL_AVG_PWR The average (in specified aggregation period) group power for sum of average power measurement in a group of nodes/enclosures within the specified entity. TOTAL_MIN_PWR The minimal group sampling (in a monitoring cycle) power in specified aggregation period for sum of minimum power measurement in a group of nodes/enclosures within the specified entity. MAX_INLET_TEMP The maximum temperature for any single node within the specified entity. AVG_INLET_TEMP The average temperature for any single node within the specified entity. MIN_INLET_TEMP The minimum temperature for any single node within the specified entity. INS_PWR The instantaneous power. CPU_USED The utilization of the physical CPU cores. MEMORY_USED The utilization of the physical memory bandwidth. IO_USED The utilization of the physical I/O bandwidth. AVG_OUTLET_TEMP The weighted average outlet temperature with respect to airflow for any single node within the specified entity. AIRFLOW The average airflow going through the node or all nodes in the group. CPU_UTIL The CPU Utilization data for nodes scaled with %. DISK_IO The Disk IO data for nodes in the unit of Bytes/Second. HUMIDITY The humidity data for PDUs with the corresponding humidity sensors.

			ESTIMATED_PWR The estimated average power for nodes without power monitoring capability or the estimated power aggregated on groups. IT_EQPMNT_PWR An estimation of power consumption by IT equipments. MAX_AVG_IT_EQPMNT_PWR The maximum average IT equipment power.

8.124.5 Response example

```
{
  "dataList": [
    {
      "entityId": "1",
      "measurementType": "IT_EQPMNT_PWR",
      "value": "2816",
      "dateTime": "2019-01-17T10:00:00+08:00",
      "period": 180
    }
  ]
}
```

8.124.6 Response field description

Field	Type	Description
dataList	Array	The item list.
entityId	Integer	The id of the entity.
measurementType	String	Measurement type. Refer to the request field definitions.
dateTime	String	The timestamp of the data. The format is "yyyy-MM-dd'T'HH:mm:ssXXX".
value	Integer	The value of the data.
period	Integer	The monitoring period. The unit is second.

8.124.7 Errors

Except the generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The entity id does not exist.
QUERY_TYPE_IS_REQUIRED	Measurement type is required.
INVALID_QUERY_TYPE	Invalid measurement type.

8.125 Get power capacity of an entity

8.125.1 Request URL

```
/rest/data/powerCapacity
```

8.125.2 HTTP request method

GET

8.125.3 Request example

https://localhost:8643/DcmConsole/rest/data/powerCapacity?entityId=-1

8.125.4 Request field description

Field	Type	Attribute	Description
entityId	Integer	Optional	The id of the datacenter, room, row, or rack. The value "-1" means all datacenters. If the parameter is not provided, "-1" will be used.

8.125.5 Response example

```
{
 "powerCapacity": "100000",
 "powerUsed": "10000"
}
```

8.125.6 Response field description

Field	Type	Description
powerCapacity	Integer	The power capacity of the specified entity. The unit is watt.
powerUsed	Integer	The used power of the specified entity. The unit is watt.

8.125.7 Errors

Except the generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The entity id is not an id of a datacenter, a room, a row, or a rack.

8.126 Get space capacity of an entity

8.126.1 Request URL

```
/rest/data/spaceCapacity
```

8.126.2 HTTP request method

GET

8.126.3 Request example

https://localhost:8643/DcmConsole/rest/data/spaceCapacity?entityId=-1

8.126.4 Request field description

Field	Type	Attribute	Description
entityId	Integer	Optional	The id of the datacenter, room, row, or rack. The value "-1" means all datacenters. If the parameter is not provided, "-1" will be used.

8.126.5 Response example

```
{
 "spaceCapacity": "1000",
 "spaceUsed": "100"
}
```

8.126.6 Response field description

Field	Type	Description
spaceCapacity	Integer	The space capacity of the specified entity. The unit is U.
spaceUsed	Double	The used space of the specified entity. The unit is U.

8.126.7 Errors

Except the generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The entity id is not an id of a datacenter, a room, a row, or a rack.

8.127 Get weight capacity of an entity

8.127.1 Request URL

```
/rest/data/weightCapacity
```

8.127.2 HTTP request method

GET

8.127.3 Request example

https://localhost:8643/DcmConsole/rest/data/weightCapacity?entityId=-1

8.127.4 Request field description

Field	Type	Attribute	Description
entityId	Integer	Optional	The id of the datacenter, room, row, or rack. The value "-1" means all datacenters. If the parameter is not provided, "-1" will be used.

8.127.5 Response example

```
{
 "weightCapacity": "1000",
 "weightUsed": "100",
 "weightUnused": "890"
}
```

8.127.6 Response field description

Field	Type	Description
weightCapacity	Double	The weight capacity of the specified entity. The unit is KG.
weightUsed	Double	The used weight of the specified entity. The unit is KG.
weightUnused	Double	The unused weight of the specified entity. The unit is KG.

8.127.7 Errors

Except the generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The entity id is not an id of a datacenter, a room, a row, or a rack.

8.128 Add a user

8.128.1 Request URL

```
/rest/configuration/users
```

8.128.2 HTTP request method

POST

8.128.3 Request example

```
{
```

```
  "name": "aa",
  "password": "user@123",
  "role": "Administrator",
  "description": "admin 1"
}
```

8.128.4 Request field description

Field	Type	Attribute	Description
name	String	Required	The name of the user to be created.
password	String	Required	The rules for user password: Passwords must be a minimum of eight characters long. Passwords must not be a repeat or reverse of the associated user ID. Passwords must be no more than two same consecutive characters. Passwords must contain at least three of the following combinations: Alpha characters - at least one lower case alpha character. Alpha characters - at least one upper case alpha character. Numeric characters - at least one numeric character. Special characters - at least one special character. The permitted characters include '!@#$%^*()_+|.:?=-'
role	String	Required	The user role. Possible values: Administrator Guest Power User
description	String	Optional	The description for the user to be created.

8.128.5 Response example

```
{
  "id": "2",
  "name": "aa",
  "role": "Administrator",
  "description": "admin 1",
  "links": [
    {
      "rel": "self",
      "href": "https://localhost:8643/DcmConsole/rest/configuration/users/2"
    },
    {
      "rel": "update",
      "href": "https://localhost:8643/DcmConsole/rest/configuration/users/2"
    },
    {
      "rel": "delete",
      "href": "https://localhost:8643/DcmConsole/rest/configuration/users/2"
    }
  ]
}
```

8.128.6 Response field description

Field	Type	Description
id	Integer	The id of the user.

Refer to the chapter "Add a user" for other field definition.

8.128.7 Errors

Except Generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
USER_NAME_EXISTS	User name already exists.
NAME_IS_REQUIRED	Name is required.
INVALID_NAME	Invalid name.
PASSWORD_IS_REQUIRED	Password is required.
INVALID_PASSWORD	Invalid password.
ROLE_IS_REQUIRED	Role is required.
INVALID_ROLE	Invalid role.
NO_PRIVILEDGE	No privilege. Only administrators can create users.

8.129 Get users

8.129.1 Request URL

/rest/configuration/users

8.129.2 HTTP request method

GET

8.129.3 Request example

https://localhost:8643/DcmConsole/rest/configuration/users?currentUser=false

8.129.4 Request field description

Field	Type	Attribute	Description
currentUser	Boolean	Optional	If the value is true, only the current user information will be returned otherwise all user information will be returned. The default value is false.

8.129.5 Response example


```
{
  "links": [
    {
      "rel": "self",
      "href": "https://localhost:8643/DcmConsole/rest/configuration/users"
    },
    {
      "rel": "add",
      "href": "https://localhost:8643/DcmConsole/rest/configuration/users"
    }
  ],
  "content": [
    {
      "id": "2",
      "name": "aa",
      "role": "Administrator",
      "description": "admin 1",
      "links": [
```

```
            {
              "rel": "self",
              "href": "https://localhost:8643/DcmConsole/rest/configuration/users/2"
            },
            {
              "rel": "update",
              "href": "https://localhost:8643/DcmConsole/rest/configuration/users/2"
            },
            {
              "rel": "delete",
              "href": "https://localhost:8643/DcmConsole/rest/configuration/users/2"
            }
          ]
        }
      ]
    }
```


8.129.6 Response field description

Refer to the chapter "Add a user".

8.129.7 Errors

Except Generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
NO_PRIVILEDGE	No privilege. Only administrators can list users.

8.130 Get a user

8.130.1 Request URL

```
/rest/configuration/users/{userId}
```

The parameter "userId" is the ID of the user.

8.130.2 HTTP request method

GET

8.130.3 Request example

https://localhost:8643/DcmConsole/rest/configuration/users/2

8.130.4 Request field description

N/A.

8.130.5 Response example

Refer to the chapter "Add a user".

8.130.6 Response field description

Refer to the chapter "Add a user".

8.130.7 Errors

Except Generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
NO_PRIVILEDGE	No privilege. Users can only get user information of their own. Only administrators can get information of other users.
TARGET_NOT_EXISTED	The user does not exist.

8.131 Update a user

8.131.1 Request URL

```
/rest/configuration/users/{userId}
```

The parameter "userId" is the ID of the user.

8.131.2 HTTP request method

PUT

8.131.3 Request example

```
{
  "name": "aa",
  "password": "user@123",
  "role": "Administrator",
  "description": "admin 1"
}
```

8.131.4 Request field description

Field	Type	Attribute	Description
name	String	Optional	New user name.
password	String	Optional	New password.
role	String	Optional	New user role.
description	String	Optional	New description.

8.131.5 Response example

Refer to the chapter "Add a user" for field definition.

8.131.6 Response field description

Refer to the chapter "Add a user".

8.131.7 Errors

Except Generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The user does not exist.
USER_NAME_EXISTS	User name already exists.
INVALID_NAME	Invalid name.
INVALID_PASSWORD	Invalid password.
INVALID_ROLE	Invalid role.
NOT_SUPPORT_THIRD_PASSWORD_CHANGE	Changing AD or LDAP user password is not supported.
NO_PRIVILEDGE	No privilege. The rules: The built-in administrator can only change its own password. Administrator users can update all fields of all of none built-in users. Non-administrator users can only update their own passwords.

8.132 Delete a user

8.132.1 Request URL

```
/rest/configuration/users/{userId}
```

The parameter "userId" is the ID of the user.

8.132.2 HTTP request method

DELETE

8.132.3 Request example

https://localhost:8643/DcmConsole/rest/configuration/users/2

8.132.4 Request field description

N/A.

8.132.5 Response example

N/A.

8.132.6 Response field description

N/A.

8.132.7 Errors

Except Generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
NO_PRIVILEDGE	No privilege. Only administrators can delete users. The built-in administrator cannot be deleted.
TARGET_NOT_EXISTED	The user does not exist.

8.133 Add a custom attribute

8.133.1 Request URL

```
/rest/configuration/customAttributes
```

8.133.2 HTTP request method

POST

8.133.3 Request example

```
{
  "name": "aa",
  "type": "Text"
}
```

8.133.4 Request field description

Field	Type	Attribute	Description
name	String	Required	The name of the attribute.
type	String	Required	Possible values: Text Number Date

8.133.5 Response example

```
{
  "id": "1",
  "name": "aa",
```

```
  "type": "Text",
  "links": [
    {
      "rel": "self",
      "href":
"https://localhost:8643/DcmConsole/rest/configuration/customAttributes/1"
    },
    {
      "rel": "update",
      "href":
"https://localhost:8643/DcmConsole/rest/configuration/customAttributes/1"
    },
    {
      "rel": "delete",
      "href":
"https://localhost:8643/DcmConsole/rest/configuration/customAttributes/1"
    }
  ]
}
```

8.133.6 Response field description

Field	Type	Description
id	Integer	The id of the attribute.

Refer to the chapter "Add a custom attribute" for other field definitions.

8.133.7 Errors

Except Generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
CUSTOM_ATTR_NAME_IS_REQUIRED	Custom attribute Name is required.
CUSTOM_ATTRIBUTE_EXISTS	The custom attribute already exists.
BUILT_IN_ATTRIBUTE_EXISTS	Built-in attribute already exists.
TYPE_IS_REQUIRED	Type is required.
INVALID_TYPE	Invalid type.

8.134 Get custom attributes

8.134.1 Request URL

```
/rest/configuration/customAttributes
```

8.134.2 HTTP request method

GET

8.134.3 Request example

https://localhost:8643/DcmConsole/rest/configuration/customAttributes

8.134.4 Request field description

N/A.

8.134.5 Response example

```
{
  "id": "1",
  "name": "aa",
  "type": "Text",
```

```
  "links": [
    {
      "rel": "self",
      "href":
"https://localhost:8643/DcmConsole/rest/configuration/customAttributes/1"
    },
    {
      "rel": "update",
      "href":
"https://localhost:8643/DcmConsole/rest/configuration/customAttributes/1"
    },
    {
      "rel": "delete",
      "href":
"https://localhost:8643/DcmConsole/rest/configuration/customAttributes/1"
    }
  ]
}
```


8.134.6 Response field description

Refer to the chapter "Add a custom attribute".

8.134.7 Errors

Refer to the generic errors defined in chapter "Error handling".

8.135 Get a custom attribute

8.135.1 Request URL

/rest/configuration/customAttributes/{attributeId}

The parameter "attributeId" is the ID of the attribute.

8.135.2 HTTP request method

GET

8.135.3 Request example

https://localhost:8643/DcmConsole/rest/configuration/customAttributes/1

8.135.4 Request field description

N/A.

8.135.5 Response example

Refer to the chapter "Add a custom attribute".

8.135.6 Response field description

Refer to the chapter "Add a custom attribute".

8.135.7 Errors

Except Generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The custom attribute does not exist.

8.136 Update a custom attribute

8.136.1 Request URL

/rest/configuration/customAttributes/{attributeId}

The parameter "attributeId" is the ID of the attribute.

8.136.2 HTTP request method

PUT

8.136.3 Request example

```
{
 "name": "bb"
}
```

8.136.4 Request field description

Field	Type	Attribute	Description
name	String	Required	New attribute name.

Note that only attribute name can be updated.

8.136.5 Response example

Refer to the chapter "Add a custom attribute" for field definition.

8.136.6 Response field description

Refer to the chapter "Add a custom attribute".

8.136.7 Errors

Except Generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The custom attribute does not exist.
CUSTOM_ATTR_NAME_IS_REQUIRED	Custom attribute name is required.
CUSTOM_ATTRIBUTE_EXISTS	The custom attribute already exists.
BUILT_IN_ATTRIBUTE_EXISTS	Built-in attribute already exists.
TYPE_IS_NOT_UPDATED	Type cannot be updated.

8.137 Delete a custom attribute

8.137.1 Request URL

```
/rest/configuration/customAttributes/{attributeId}
```

The parameter "attributeId" is the ID of the attribute.

8.137.2 HTTP request method

DELETE

8.137.3 Request example

https://localhost:8643/DcmConsole/rest/configuration/customAttributes/2

8.137.4 Request field description

N/A.

8.137.5 Response example

N/A.

8.137.6 Response field description

N/A.

8.137.7 Errors

Except Generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The attribute does not exist.

8.138 Add a provisioning task

8.138.1 Request URL

```
/rest/provisioningTasks
```

8.138.2 HTTP request method

POST

8.138.3 Request example

```
{
  "beginDate": "2022-04-12T10:00:00+08:00",
  "taskType": "FirmwareUpdate",
  "deviceIds": [
    2,
    3
  ],
  "memo": "An example note.",
  "autoResetServer": "true"
}
```

8.138.4 Request field description

Field	Type	Attribute	Description
beginDate	String	Optional	The begin time of this provisioning task. The format is "yyyy-MM-dd'T'HH:mm:ssXXX". If it is not specified, this provisioning task should be executed immediately.
taskType	String	Required	The provisioning task type. The supported values: FirmwareUpdate Firmware update for BIOS/BMC/ME/SDR component CustomFirmwareUpdate Firmware update for Custom component MountISO Virtual media mount UnmountISO Virtual media unmount SystemConfigUpdate System config update
deviceIds	Array	Required	The device ids which need to perform provisioning task.
memo	String	Optional	The memo for this provisioning task.
autoResetServer	Boolean	Optional	whether it is need to automatically reset server for the task.

8.138.5 Response example

```
{
  "taskId": 11,
```

```
  "beginDate": "2022-04-12T10:00:00+08:00",
  "taskType": "FirmwareUpdate",
  "deviceIds": [
    2,
    3
  ],
  "autoResetServer": true,
  "links": [
    {
      "rel": "self",
      "href": "https://localhost:8643/DcmConsole/rest/provisioningTasks/11"
    }
  ]
}
```


8.138.6 Response field description

Field	Type	Description
taskId	Integer	The provisioning task Id
beginDate	String	The begin time stamp of the specified user. The format is "yyyy-MM-dd'T'HH:mm:ssXXX". If this timestamp is not specified in the request, there will be no such field in the response.
taskType	String	The provisioning task type. The possible values: FirmwareUpdate Firmware update for BIOS/BMC/ME/SDR component CustomFirmwareUpdate Firmware update for Custom component MountISO Virtual media mount UnmountISO Virtual media unmount SystemConfigUpdate System config update
deviceIds	Array	The device ids to perform provisioning task.
autoResetServer	Boolean	Whether it is need to automatically reset server for the task.

8.138.7 Errors

Generic errors defined in chapter "Error handling".

8.139 Get provisioning tasks

8.139.1 Request URL

/rest/provisioningTasks

8.139.2 HTTP request method

GET

8.139.3 Request example

https://localhost:8643/DcmConsole/rest/provisioningTasks

8.139.4 Request field description

N/A.

8.139.5 Response example


```
{
  "links": [
    {
      "rel": "self",
      "href": "https://localhost:8643/DcmConsole/rest/provisioningTasks"
    },
    {
      "rel": "add",
      "href": "https://localhost:8643/DcmConsole/rest/provisioningTasks"
    }
  ],
  "content": [
    {
      "taskId": 2,
      "createDate": "2022-04-08T15:46:17+08:00",
      "beginDate": "2022-04-10T20:00:00+08:00",
      "taskType": "FirmwareUpdate",
      "status": "Stopped",
      "timeElapsed": 0,
      "deviceIds": [
        2,
        3
      ],
      "result": [
        {
          "deviceId": 2,
          "status": "Fail",
          "message": ""
        },
        {
          "deviceId": 3,
          "status": "Fail",
          "message": ""
        }
      ],
      "autoResetServer": true,
      "links": [
        {
          "rel": "self",
          "href": "https://localhost:8643/DcmConsole/rest/provisioningTasks/2"
        },
        {
          "rel": "delete",
          "href": "https://localhost:8643/DcmConsole/rest/provisioningTasks/2"
        },
        {
          "rel": "control",
          "href":
"https://localhost:8643/DcmConsole/rest/provisioningTasks/2/control"
        }
      ]
    },
    {
      "taskId": 3,
      "createDate": "2022-04-11T09:20:19+08:00",
      "beginDate": "2022-04-11T09:20:19+08:00",
      "endDate": "2022-04-11T09:20:19+08:00",
      "taskType": "SystemConfigUpdate",
      "status": "Stopped",
      "timeElapsed": 0,
      "memo": "",
```

```
        "deviceIds": [
          9
        ],
        "result": [
          {
            "deviceId": 9,
            "status": "Fail",
            "msg": ""
          }
        ],
        "autoResetServer": true,
        "links": [
          {
            "rel": "self",
            "href": "https://localhost:8643/DcmConsole/rest/provisioningTasks/3"
          },
          {
            "rel": "delete",
            "href": "https://localhost:8643/DcmConsole/rest/provisioningTasks/3"
          },
          {
            "rel": "control",
            "href":
"https://localhost:8643/DcmConsole/rest/provisioningTasks/3/control"
          }
        ]
      },
      {
        "taskId": 4,
        "createDate": "2022-04-11T09:20:44+08:00",
        "beginDate": "2022-04-11T09:20:44+08:00",
        "endDate": "2022-04-11T09:20:44+08:00",
        "taskType": "SystemConfigUpdate",
        "status": "Stopped",
        "timeElapsed": 0,
        "memo": "",
        "deviceIds": [
          2
        ],
        "result": [
          {
            "entityId": 2,
            "status": "Fail",
            "msg": ""
          }
        ],
        "autoResetServer": true,
        "links": [
          {
            "rel": "self",
            "href": "https://localhost:8643/DcmConsole/rest/provisioningTasks/4"
          },
          {
            "rel": "delete",
            "href": "https://localhost:8643/DcmConsole/rest/provisioningTasks/4"
          },
          {
            "rel": "control",
            "href":
"https://localhost:8643/DcmConsole/rest/provisioningTasks/4/control"
          }
        ]
      }
```

```
    ]
}
```

8.139.6 Response field description

Field	Type	Description
taskId	Integer	The provisioning task Id
createDate	String	The creation time stamp of the provisioning task. The format is "yyyy-MM-dd'T'HH:mm:ssXXX".
beginDate	String	The begin time stamp of the provisioning task. The format is "yyyy-MM-dd'T'HH:mm:ssXXX".
endDate	String	The end time stamp of the provisioning task. The format is "yyyy-MM-dd'T'HH:mm:ssXXX".
taskType	String	The provisioning task type. The possible values: FirmwareUpdate CustomFirmwareUpdate MountISO UnmountISO SystemConfigUpdate SingleSystemConfigUpdate BootOverride LoadBIOSDefault BatchUpdate RedfishFirmwareUpdate RedfishMountISO RedfishUnmountISO GetSystemConfig
status	String	The provisioning task status. The possible values: Waiting Running Stopped
timeElapsed	Integer	The elapsed time of this provisioning task
deviceIds	Array	The device ids
result	Array	The result list
autoResetServer	Boolean	Whether it is need to automatically reset server for the task.
deviceId	Integer	The device Id
status	String	The result for the target device. The possible values: Success Fail Waiting Unsupported Duplicated BootOverideUnsupported Processing
message	String	The detailed result message

8.139.7 Errors

Generic errors defined in chapter "Error handling".

8.140 Get a provisioning task

8.140.1 Request URL

/rest/provisioningTasks/{taskId}

The parameter is the ID of the provisioning task.

8.140.2 HTTP request method

GET

8.140.3 Request example

https://localhost:8643/DcmConsole/rest/provisioningTasks/2

8.140.4 Request field description

N/A.

8.140.5 Response example


```
{
  "taskId": 2,
  "createDate": "2022-04-08T15:46:17+08:00",
  "beginDate": "2022-04-10T10:00:00+08:00",
  "taskType": "FirmwareUpdate",
  "status": "Stopped",
  "timeElapsed": 0,
  "result": [
    {
      "deviceId": 3,
      "status": "Fail",
      "message": "message example"
    },
    {
      "deviceId ": 2,
      "status": "Success",
      "message": "message example"
    }
  ],
  "autoResetServer": true,
  "links": [
    {
      "rel": "self",
      "href": "https://localhost:8643/DcmConsole/rest/provisioningTasks/2"
    },
    {
      "rel": "delete",
      "href": "https://localhost:8643/DcmConsole/rest/provisioningTasks/2"
    },
    {
      "rel": "control",
      "href":
"https://localhost:8643/DcmConsole/rest/provisioningTasks/2/control"
    }
  ]
}
```


8.140.6 Response field description

Field	Type	Description
taskId	Integer	The provisioning task Id

createDate	String	The creation time stamp of the provisioning task. The format is "yyyy-MM-dd'T'HH:mm:ssXXX".
beginDate	String	The begin time stamp of the provisioning task. The format is "yyyy-MM-dd'T'HH:mm:ssXXX".
endDate	String	The end time stamp of the provisioning task. The format is "yyyy-MM-dd'T'HH:mm:ssXXX".
taskType	String	The provisioning task type. The possible values: FirmwareUpdate CustomFirmwareUpdate MountISO UnmountISO SystemConfigUpdate SingleSystemConfigUpdate BootOverride LoadBIOSDefault BatchUpdate RedfishFirmwareUpdate RedfishMountISO RedfishUnmountISO GetSystemConfig
status	String	The provisioning task status. The possible values: Waiting Running Stopped
timeElapsed	Integer	The elapsed time of this provisioning task
deviceIds	Array	The device ids
result	Array	The results list
autoResetServer	Boolean	Whether it is need to automatically reset server for the task.
deviceId	Integer	The device Id
status	String	The result for the target device. The possible values: Success Fail Waiting Unsupported Duplicated BootOverideUnsupported Processing
message	String	The detailed result message

8.140.7 Errors

Except Generic errors defined in chapter "Error handling", the following errors can be returned:

Error Code	Description
TARGET_NOT_EXISTED	The target does not exist.

8.141 Control a provisioning task

8.141.1 Request URL

/rest/provisioningTasks/{taskId}/control

The parameter taskId is the ID of the provisioning task.

8.141.2 HTTP request method

POST

8.141.3 Request example

```
{
  "controlAction": "STOP"
}
```

8.141.4 Request field description

Field	Type	Attribute	Description
controlAction	String	Required	Possible values: RERUN Re-run a stopped provisioning task. STOP Stop a provisioning task.

8.141.5 Response example

```
{
  "taskId": 2,
  "createDate": "2022-04-11T20:29:33+08:00",
  "beginDate": "2022-04-11T20:35:39+08:00",
  "endDate": "2022-04-11T20:35:39+08:00",
  "taskType": "FirmwareUpdate",
  "status": "Stopped",
  " deviceIds ": [
    13
  ],
  "autoResetServer": true,
  "links": [
    {
      "rel": "self",
      "href": "https://localhost:8643/DcmConsole/rest/provisioningTasks/2"
    },
    {
      "rel": "delete",
      "href": "https://localhost:8643/DcmConsole/rest/provisioningTasks/2"
    },
    {
      "rel": "control",
      "href":
"https://localhost:8643/DcmConsole/rest/provisioningTasks/2/control"
    }
  ]
}
```

8.141.6 Response field description

Field	Type	Description
taskId	Integer	The provisioning task Id
createDate	String	The creation time stamp of the provisioning task. The format is "yyyy-MM-dd'T'HH:mm:ssXXX".
beginDate	String	The begin time stamp of the provisioning task. The format is "yyyy-MM-dd'T'HH:mm:ssXXX".

endDate	String	The end time stamp of the provisioning task. The format is "yyyy-MM-dd'T'HH:mm:ssXXX".
taskType	String	The provisioning task type. The possible values: FirmwareUpdate Firmware update for BIOS/BMC/ME/SDR component CustomFirmwareUpdate Firmware update for Custom component MountISO Virtual media mount UnmountISO Virtual media unmount SystemConfigUpdate System config update
status	String	The provisioning task status. The possible values: Waiting Running Stopped
autoResetServer	Boolean	Whether it is need to automatically reset serverfor the task.
deviceIds	Array	The device ids to perform provisioning task.

8.141.7 Errors

The following errors may be returned.

Error Code	Description
TARGET_NOT_EXISTED	The task Id does not exist.
CONTROL_ACTION_IS_REQUIRED	Control action is required.
INVALID_CONTROL_ACTION	Invalid control action value.
ACTION_IS_NOT_SUPPORT	Possible reasons: To re-run a running task is not supported. To stop a waiting task is not supported. To stop a stopped task will always return OK.

8.142 Delete a provisioning task

8.142.1 Request URL

/rest/provisioningTasks/{taskId}

The parameter taskId is the ID of the provisioning task.

8.142.2 HTTP request method

DELETE

8.142.3 Request example

https://localhost:8643/DcmConsole/rest/provisioningTasks/21

8.142.4 Request field description

N/A.

8.142.5 Response example

N/A.

8.142.6 Response field description

N/A.

8.142.7 Errors

The following errors may be returned.

Error Code	Description
TARGET_NOT_EXISTED	The provisioning task id does not exist.
TASK_IS_RUNNING	A running task cannot be deleted.

8.143 Get power status of entities

8.143.1 Request URL

/rest/data/powerStatusHistory

8.143.2 HTTP request method

GET

8.143.3 Request example

https://localhost:8643/DcmConsole/rest/data/powerStatusHistory?startDateTime=2022-01-17T10:00%2B08:00&&endDateTime=2022-04-28T20:00%2B08:00

8.143.4 Request field description

Field	Type	Attribute	Description
startDateTime	String	Optional	The start date time. If it is not specified, the API will return all historical power status data since 1970-01-01 0:00.
endDateTime	String	Optional	The end date time. The default value is the current time.

8.143.5 Response example


```
[
  {
    "entityId": 8,
    "address": "10.40.122.25",
    "serialNumber": "08ac1f2d-a6e1-45bd-ae7f-9c41f6b45b3b",
    "uuid": "fe057a57-1d35-4366-a384-87e895653cef",
    "sku": "5e0bee4a-9d0b-4260-9877-1eeec6f0542c",
    "powerStatusHistory": [
      {
        "dateTime": "2022-01-28T10:43:48+08:00",
        "eventType": "ADDED"
      },
      {
        "dateTime": "2022-02-28T10:43:48+08:00",
        "eventType": "ON"
      },
      {
        "dateTime": "2022-03-28T10:43:48+08:00",
        "eventType": "OFF"
      },
      {
        "dateTime": "2022-04-28T10:43:48+08:00",
        "eventType": "REMOVED"
      }
    ]
  },
  {
    "entityId": 9,
    "address": "10.40.125.21",
    "serialNumber": "e18af9d2-fcc6-41b5-b28c-c57436cbe1ac",
    "uuid": "2d8c9438-8aa3-44ae-97d0-caea8d9f4d30",
```

```
    "sku": "e3b66b1f-fd37-4109-bdce-fc19bf231ae8",
    "powerStatusHistory": [
      {
        "dateTime": "2022-01-28T10:43:48+08:00",
        "eventType": "ADDED"
      },
      {
        "dateTime": "2022-02-28T10:43:48+08:00",
        "eventType": " ON"
      },
      {
        "dateTime": "2022-03-28T10:43:48+08:00",
        "eventType": " OFF"
      },
      {
        "dateTime": "2022-04-28T10:43:48+08:00",
        "eventType": "REMOVED"
      }
    ]
  }
]
```


8.143.6 Response field description

Field	Type	Description
entityId	Integer	The entity Id.
address	String	The entity address.
serialNumber	String	Serial Number of the system.
uuid	String	UUID of the system.
sku	String	The subsystem SKU.
powerStatusHistory	Array	The system power status history array.
dateTime	String	Power status event time.
eventType	String	Power status event type. The possible values: ADDED The device is added into DCM. ON The device power status is changed to ON. Or the device is detected with ON for the first time. OFF The device power status is changed to OFF. Or the device is detected with OFF for the first time. REMOVED The device is removed from DCM. UNKNOWN The device power status has not been detected until the timestamp which is specified by endDateTime.

8.143.7 Errors

Generic errors defined in chapter "Error handling".

8.144 Get low utilization servers

8.144.1 Request URL

```
/rest/data/lowUtilizationServers
```

8.144.2 HTTP request method

GET

8.144.3 Request example

https://localhost:8643/DcmConsole/rest/data/lowUtilizationServers

8.144.4 Request field description

N/A

8.144.5 Response example


```
{
  "analysisTimestamp": "2022-07-12T16:09:55+08:00",
  "lowUtilizationServers": [
    {
      "name": "Server - 10.239.172.109",
      "fullPath": "dc > room > row > rack > Server - 10.239.172.109",
      "entityId": 7,
      "entityPath": [
        5,
        4,
        3,
        2
      ],
      "estimatedAverageUtilization": "0.00%",
      "estimatedDistributionUtilization": "0.00%",
      "potentialEnergySave": "41"
    },
    {
      "name": "Server - 10.239.172.143",
      "fullPath": "dc > room > row > rack > Server - 10.239.172.143",
      "entityId": 8,
      "entityPath": [
        5,
        4,
        3,
        2
      ],
      "estimatedAverageUtilization": "8.75%",
      "estimatedDistributionUtilization": "13.24%",
      "potentialEnergySave": "23",
      "minCarbonEmission": "123",
      "maxCarbonEmission": "456"
    }
  ]
}
```


8.144.6 Response field description

Field	Type	Description
analysisTimestamp	Date	The analysis completion time.
lowUtilizationServers	Array	Array of low utilization servers.
name	String	The name of the device.
fullPath	String	The path in the hierarchy which will be like "dc > room > row > rack".
entityId	Integer	The id of the device.
entityPath	Array	The parent id array which will be in sequence of rack id, row id, room id and datacenter id.
estimatedAverageUtilization	String	The estimated average utilization
estimatedDistributionUtilization	String	The value below which the estimated utilization falls for 99% of the time.

potentialEnergySave	String	Estimated weekly energy Savings (kWh) after consolidation.
minCarbonEmission	Integer	Estimated monthly min carbon Emission savings after consolidation. The unit is grams.
maxCarbonEmission	Integer	Estimated monthly max carbon emission savings after consolidation. The unit is grams.

8.144.7 Errors

The following errors may be returned.

Error Code	Description
ERR_NO_LU_DATA	No analysis results yet.
ERR_LUA_IN_PROGRESS	Analysis is in progress.

8.145 Analyze low utilization servers

8.145.1 Request URL

/rest/data/lowUtilizationServers/analyze

8.145.2 HTTP request method

POST

8.145.3 Request example

https://localhost:8643/DcmConsole/rest/data/lowUtilizationServers/analyze

8.145.4 Request field description

N/A

8.145.5 Response example

N/A

8.145.6 Response field description

N/A

8.145.7 Errors

Generic errors defined in chapter "Error handling".

8.146 Get carbon emission of a data center

8.146.1 Request URL

/rest/datacenters/{dcId}/carbonEmission

The parameter "dcId" is the ID of the datacenter. -4 means all datacenters.

8.146.2 HTTP request method

GET

8.146.3 Request example

https://localhost:8643/DcmConsole/rest/datacenters/2/carbonEmission

8.146.4 Request field description

N/A

8.146.5 Response example


```
{
  "timestamp": "2022-08-18T13:49:06+08:00",
  "stats": {
    "entityId": 2,
    "entityName": "dc1",
    "startingDate": "2022-07-01",
    "monthEmission": [
      0,
      271512
    ],
    "annualEmission": [
      0,
      271512
    ],
    "totalAnnualEmission": 271512,
    "cue": 0.72
  }
}
```


8.146.6 Response field description

Field	Type	Description
timestamp	Date	The timestamp of the data
stats	Object	The container of carbon emission data
entityId	Integer	The entity id of the datacenter. -4 means all datacenters
entityName	String	The entity name of the datacenter
startingDate	String	The carbon emission starting date
monthEmission	Array	Monthly data on carbon emission
annualEmission	Array	Annual data on carbon emission
totalAnnualEmission	Integer	Total annual data on carbon emission
cue	Double	Carbon usage effectiveness. -1.0 means no valid data.

8.146.7 Errors

Except the generic errors defined in chapter "Error handling".

8.147 Get carbon emission of a room

8.147.1 Request URL

/rest/rooms/{roomId}/carbonEmission

The parameter "roomId" is the ID of the room.

8.147.2 HTTP request method

GET

8.147.3 Request example

https://localhost:8643/DcmConsole/rest/rooms/3/carbonEmission

8.147.4 Request field description

N/A

8.147.5 Response example


```
{
  "timestamp": "2022-08-18T13:49:06+08:00",
  "stats": {
    "entityId": 3,
    "entityName": "room1",
    "startingDate": "2022-07-01",
    "monthEmission": [
      0,
      271512
    ],
    "annualEmission": [
      0,
      271512
    ],
    "totalAnnualEmission": 271512,
    "cue": 0.72
  }
}
```


8.147.6 Response field description

Field	Type	Description
timestamp	Date	The timestamp of the data
stats	Object	The container of carbon emission data
entityId	Integer	The entity id of the room
entityName	String	The entity name of the room
startingDate	String	The carbon emission starting date
monthEmission	Array	Monthly data on carbon emission
annualEmission	Array	Annual data on carbon emission
totalAnnualEmission	Integer	Total annual data on carbon emission
cue	Double	Carbon usage effectiveness. -1.0 means no valid data.

8.147.7 Errors

Except the generic errors defined in chapter "Error handling".

8.148 Get carbon emission of a row

8.148.1 Request URL

/rest/rows/{rowId}/carbonEmission

The parameter "rowId" is the ID of the row.

8.148.2 HTTP request method

GET

8.148.3 Request example

https://localhost:8643/DcmConsole/rest/rows/4/carbonEmission

8.148.4 Request field description

N/A

8.148.5 Response example

```
{
  "timestamp": "2022-08-18T13:49:06+08:00",
  "stats": {
    "entityId": 4,
    "entityName": "row1",
    "startingDate": "2022-07-01",
    "monthEmission": [
      0,
      271512
    ],
    "annualEmission": [
      0,
      271512
    ],
    "totalAnnualEmission": 271512,
    "cue": -1
  }
}
```

8.148.6 Response field description

Field	Type	Description
timestamp	Date	The timestamp of the data
stats	Object	The container of carbon emission data
entityId	Integer	The entity id of the row
entityName	String	The entity name of the row
startingDate	String	The carbon emission starting date
monthEmission	Array	Monthly data on carbon emission
annualEmission	Array	Annual data on carbon emission
totalAnnualEmission	Integer	Total annual data on carbon emission
cue	Double	Carbon usage effectiveness. -1.0 means no valid data.

8.148.7 Errors

Except the generic errors defined in chapter "Error handling".

8.149 Get carbon emission of a rack

8.149.1 Request URL

```
/rest/racks/{rackId}/carbonEmission
```

The parameter "rackId" is the ID of the rack.

8.149.2 HTTP request method

GET

8.149.3 Request example

https://localhost:8643/DcmConsole/rest/racks/36/carbonEmission

8.149.4 Request field description

N/A

8.149.5 Response example


```
{
  "timestamp": "2022-08-18T14:12:30+08:00",
  "stats": {
    "entityId": 36,
    "entityName": "rack",
    "startingDate": "2022-07-01",
    "monthEmission": [
      0,
      38165
    ],
    "annualEmission": [
      0,
      38165
    ],
    "totalAnnualEmission": 38165,
    "cue": -1
  }
}
```


8.149.6 Response field description

Field	Type	Description
timestamp	Date	The timestamp of the data
stats	Object	The container of carbon emission data
entityId	Integer	The entity id of the rack
entityName	String	The entity name of the rack
startingDate	String	The carbon emission starting date
monthEmission	Array	Monthly data on carbon emission
annualEmission	Array	Annual data on carbon emission
totalAnnualEmission	Integer	Total annual data on carbon emission
cue	Double	Carbon usage effectiveness. -1.0 means no valid data.

8.149.7 Errors

Except the generic errors defined in chapter "Error handling".

8.150 Get carbon emission of a group

8.150.1 Request URL

/rest/groups/{groupId}/carbonEmission

The parameter "groupId" is the ID of the row.

8.150.2 HTTP request method

GET

8.150.3 Request example

https://localhost:8643/DcmConsole/rest/groups/94/carbonEmission

8.150.4 Request field description

N/A

8.150.5 Response example


```
{
  "timestamp": "2022-08-18T14:21:26+08:00",
  "stats": {
    "entityId": 94,
    "entityName": "g1",
    "startingDate": "2022-07-01",
    "monthEmission": [
      0,
      19390
    ],
    "annualEmission": [
      0,
      19390
    ],
    "totalAnnualEmission": 19390,
    "cue": 0.72
  }
}
```


8.150.6 Response field description

Field	Type	Description
timestamp	Date	The timestamp of the data
stats	Object	The container of carbon emission data
entityId	Integer	The entity id of the group
entityName	String	The entity name of the group
startingDate	String	The carbon emission starting date
monthEmission	Array	Monthly data on carbon emission
annualEmission	Array	Annual data on carbon emission
totalAnnualEmission	Integer	Total annual data on carbon emission
cue	Double	Carbon usage effectiveness. -1.0 means no valid data.

8.150.7 Errors

Except the generic errors defined in chapter "Error handling".

8.151 Get carbon emission of a device

8.151.1 Request URL

/rest/devices/{deviceId}/carbonEmission

The parameter "deviceId" is the ID of the row.

8.151.2 HTTP request method

GET

8.151.3 Request example

https://localhost:8643/DcmConsole/rest/devices/2/carbonEmission

8.151.4 Request field description

N/A

8.151.5 Response example


```
{
  "timestamp": "2022-08-25T16:20:55+08:00",
  "stats": {
    "entityId": 2,
    "entityName": "Server - 10.239.172.9",
    "startingDate": "2022-07-01",
    "monthEmission": [
      0,
      19390
    ],
    "annualEmission": [
      0,
      19390
    ],
    "totalAnnualEmission": 19390,
    "cue": -1
  }
}
```


8.151.6 Response field description

Field	Type	Description
timestamp	Date	The timestamp of the data
stats	Object	The container of carbon emission data
entityId	Integer	The entity id of the device
entityName	String	The entity name of the device
startingDate	String	The carbon emission starting date
monthEmission	Array	Monthly data on carbon emission
annualEmission	Array	Annual data on carbon emission
totalAnnualEmission	Integer	Total annual data on carbon emission
cue	Double	Carbon usage effectiveness. -1.0 means no valid data.

8.151.7 Errors

Except the generic errors defined in chapter "Error handling".

8.152 Get carbon emission projection of a datacenter

8.152.1 Request URL

/rest/datacenters/{dcId}/carbonEmissionProjection

The parameter "dcId" is the ID of the datacenter. -4 means all datacenters.

8.152.2 HTTP request method

GET

8.152.3 Request example

https://localhost:8643/DcmConsole/rest/datacenters/2/carbonEmissionProjection

8.152.4 Request field description

N/A

8.152.5 Response example

```
[
  {
    "time": "2022-08-19T00:00:00+08:00",
    "value": 21462
  },
  {
    "time": "2022-08-20T00:00:00+08:00",
    "value": 21462
  },
  {
    "time": "2022-12-01T00:00:00+08:00",
    "value": 21462
  }
]
```

8.152.6 Response field description

Field	Type	Description
time	Date	Timestamp of the projection data
value	Integer	Value of the projection data

8.152.7 Errors

The following errors may be returned.

Error Code	Description
ERR_NO_CARBON_EMISSION_PROJECTION_DATA	No carbon emission projection data yet.

8.153 Get carbon emission projection of a group

8.153.1 Request URL

```
/rest/groups/{groupId}/carbonEmissionProjection
```

The parameter "groupId" is the ID of the row.

8.153.2 HTTP request method

GET

8.153.3 Request example

https://localhost:8643/DcmConsole/rest/groups/94/carbonEmissionProjection

8.153.4 Request field description

N/A

8.153.5 Response example

```
[
  {
    "time": "2022-08-19T00:00:00+08:00",
    "value": 12345
  },
  {
    "time": "2022-08-20T00:00:00+08:00",
    "value": 12345
  },
  {
    "time": "2022-12-01T00:00:00+08:00",
    "value": 12345
```

```
    }
]
```

8.153.6 Response field description

Field	Type	Description
time	Date	Timestamp of the projection data
value	Integer	Value of the projection data

8.153.7 Errors

The following errors may be returned.

Error Code	Description
ERR_NO_CARBON_EMISSION_PROJECTION_DATA	No carbon emission projection data yet.

8.154 Get annual renewable energy ratio of a datacenter

8.154.1 Request URL

```
/rest/datacenters/{dcId}/annualRenewableEnergyRatio
```

The parameter "dcId" is the ID of the datacenter. -4 means all datacenters.

8.154.2 HTTP request method

GET

8.154.3 Request example

https://localhost:8643/DcmConsole/rest/datacenters/2/annualRenewableEnergyRatio

8.154.4 Request field description

N/A

8.154.5 Response example

```
{
 "startingDate": "2023-01-01",
 "annualRenewableEnergyRatio": 0.5
}
```

8.154.6 Response field description

Field	Type	Description
startingDate	String	The starting date of annual renewable energy ratio.
annualRenewableEnergyRatio	Double	Annual renewable energy ratio. -1.0 means no valid data.

8.154.7 Errors

Generic errors defined in chapter "Error handling".

8.155 Get annual renewable energy ratio of a group

8.155.1 Request URL

```
/rest/groups/{groupId}/annualRenewableEnergyRatio
```

The parameter "groupId" is the ID of the group.

8.155.2 HTTP request method

GET

8.155.3 Request example

https://localhost:8643/DcmConsole/rest/groups/2/annualRenewableEnergyRatio

8.155.4 Request field description

N/A

8.155.5 Response example

```
{
 "startingDate": "2023-01-01",
 "annualRenewableEnergyRatio": 0.5
}
```

8.155.6 Response field description

Field	Type	Description
startingDate	String	The starting date of annual renewable energy ratio.
annualRenewableEnergyRatio	String	Annual renewable energy ratio. -1.0 means no valid data.

8.155.7 Errors

Generic errors defined in chapter "Error handling".