enlighted™
A Siemens Company

Energy Manager API User Guide

Doc: 93-01343-01 Rev.15, 05032019
© 2018-19

Contents

Chapter 1: Overview 5
Common Request Headers 5
Common Response Headers 6
Status Codes 6
Error Codes 6
Chapter 2: User Authentication 7
Generating the REST API Key 7
Chapter 3: Energy Manager APIs 10
Get Energy Manager Aggregate Energy Consumption 10
Set Energy Manager Emergency 10
Get BACnet Health 11
Chapter 4: Organization APIs 12
Get Organization Details 12
Get All Campuses 13
Get All Buildings 14
Get All Floors 15
Get Floor Plan Image 17
Chapter 5: Area APIs 18
Get all Areas 18
Get Area Energy Consumption 19
Get Area Occupancy 20
Get Area Occupancy for a Floor 21
Get Area Outage 23
Set Area Emergency 24
Chapter 6: Fixture APIs 25
Get Sensor Details by Floor 25
Get Sensor Details 26
Get Sensor Location by Floor 27
Get All Fixtures by Area 29
Get Sensor Profiles 30

Assign Profile 32
Get Sensor Energy Consumption 33
Get Sensor Energy Consumption by Area 34
Manual Override for a Fixture 34
Auto Mode for a Fixture 35
Chapter 7: Plugload APIs 37
Get Plugload Details 37
Get All Plugloads by Floor 38
Get All Plugloads by Area 39
Get Plugload Energy Consumption 41
Get Plugload Energy Consumption by Area 41
Set Plugload Status 42
Chapter 8: Conference Room APIs 43
Get Switch Groups 43
Get Switch Scenes 44
Get Scene Light Levels for a Switch 45
Apply Scene 46
Manual Override for Switch Group 47
Chapter 9: Data APIs 49
Get Sensor Energy Data by Floor 49
Get Aggregate Sensor Energy Data by Floor 51
Chapter 10: Demand Response APIs 53
Schedule Demand Response (DR) for all Facilities 53
Schedule Demand Response (DR) for Selected Facilities 54
Update DR for all Facilities 55
Update DR for Selected Facilities 56
List DR 58
Cancel DR 59
Chapter 11: Tunable White APIs 60
Change Color 60
Scene Control 62
Chapter 12: Bluetooth Services 63

Get all BLE Fixtures by Floor 63
Set BLE Mode of Sensor 65
API Example 66

Chapter 1: Overview

The Enlighted lighting solution optimizes energy savings while enhancing occupant productivity, well-being, and security. An Energy Manager (EM) server stores sensor, status, and fault data from fixtures in the building, allowing users to configure fixture lighting behavior and monitor energy consumption and device status.

The Enlighted Energy Manager (EM) supports standard REST-based APIs including XML and JSON messaging formats. Developers can write applications using REST APIs to query and retrieve information from the Energy Manager. The sample responses to the web services are in JSON or XML format. Making authorized requests to the Energy Manager requires that you are granted permission to access data. The users must be authenticated to send or receive API requests to and from the Energy Manager. All APIs sample code responses are based on the following example.

Let's consider Acme Inc. organization structure with two campuses Hampton and Monroe campus. The Monroe campus comprises a single Main building with three floors: Floor one, Floor two, and Floor three. Floor 1 accommodates two areas Hardware and Software used by the respective engineering teams. The area comprises a grouping of a subset of devices including sensors, Enlighted Room Control (ERC) switches, plugloads, etc., on the floor. All devices are identified with a unique device ID and name. The sensors in an area can be enabled as zone sensors to report occupancy information.

In addition to the physical devices, groups are a cluster of sensors that communicate with each other in the area to control lighting based on occupancy or motion. A switch controls the light level of a group of selected fixtures on the floor. Each set of light levels controlled by the switch is called a scene.

After user authorization, the EM APIs allow users to monitor energy consumption, BACnet health, and turn lights on during an emergency, obtain organization, floor, area, and fixture details. The Demand Response APIs contribute to energy load reduction and savings during peak demand time.

Common Request Headers

The following table describes request headers common to API requests.

Header Name	Description	Example
Hostname	Hostname of the requested content. This parameter is required for HTTP 1.1	Em_ip_address or hostname (Energy Manager server)
Content-Type	The MIME type of the body of the request (used with POST and PUT requests). Both content types are valid.	Application/XML or Application/JSON
Accept	Content types that are acceptable for the response. The response is displayed in the format specified in this field.	Application/XML or Application/JSON
ApiKey	Username and API Key	{$Username} For example: bob
Authorization	SHA-1 authorization comprises Username, API key, and Timestamp. API Key can be obtained from EM in ADMINISTRATION -> USER MANAGEMENT menu.	{sha1sum of $Username$APIKey$ts}

ts	Timestamp: Time, date, and day of the API call in milliseconds and UTC format For example, 1448011348780	{$ts}

Common Response Headers

The following table describes response headers that are common to most responses.

Header Name	Description	Example
Server	The name of the server that generated the response	em_server (Energy Manager server)
Content-Type	The content type of the response content in the body	Application/JSON
Content-Length	The response length in bytes	7582
Date	The date and time the EM server responded	Sat, 21 Jul 2015 17:06:51 GMT

Status Codes

Status codes define the response of the request to the client. The following table lists the status codes.

Status code	Status Message	Description
200	Ok	Valid response, request successful

Error Codes

When there is an error, the header information contains:

- Content-Type: Application/XML
- An appropriate 3xx, 4xx status code.

The following table lists the error codes.

Error codes	Error Message	Description
302	Moved temporarily	User not authenticated
401	Authentication failed	User authentication failed
403	Request forbidden, Permission denied	No access to the user
404	API not found	API not valid or not found
408	API received after time expired	API canceled

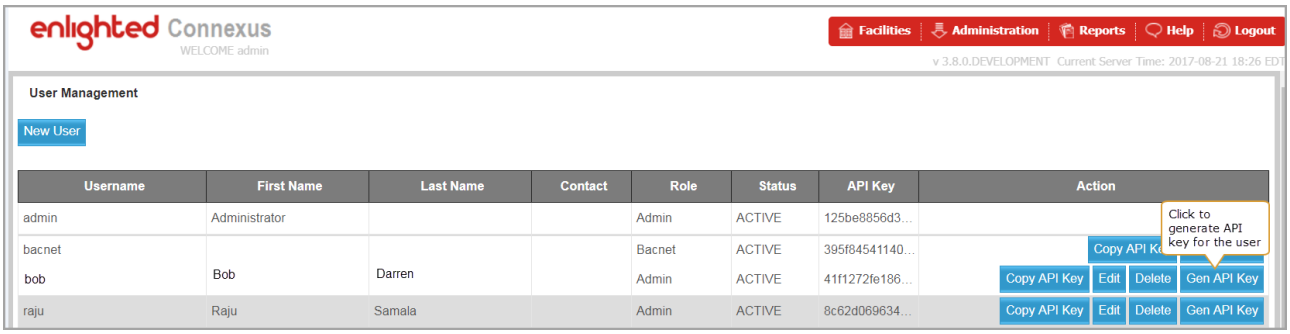
Chapter 2: User Authentication

Users must be authenticated to send or receive API requests to and from the Energy Manager. For each user, an API key is generated and provided to the user. The API calls include a timestamp of when the call was made to avoid replay attacks. For authentication, send the following headers along with the REST API.

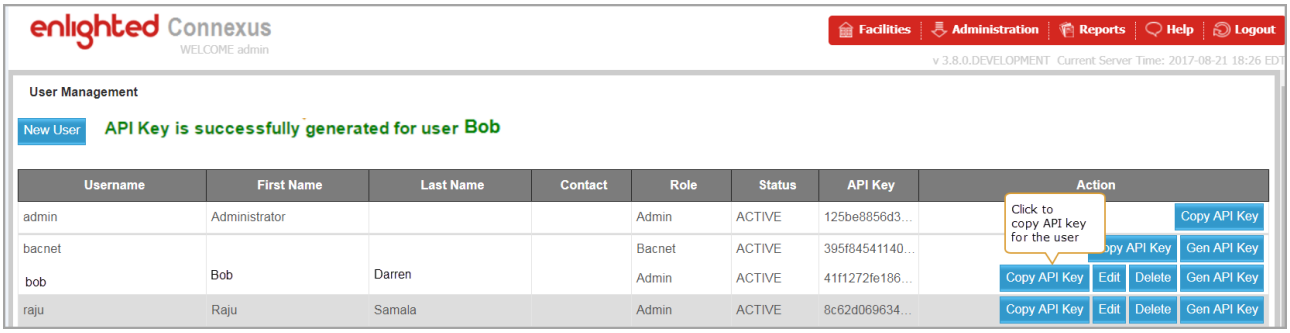
- Username: Unique API key identifier for the user
- Authorization: SHA-1 authentication key
- Timestamp: Time, date, and day of the API call

Generating the REST API Key

You can generate the API key from the *User Management* screen in Energy Manager. Log in to Energy Manager, navigate to ADMINISTRATION->USER MANAGEMENT. Click the *Gen API Key* button to generate the API key. In the confirmation pop-up that is displayed, click *OK* to confirm.



A confirmation message is displayed indicating that the key is successfully generated. Click *Copy API key* for the user.



For example, user Bob is assigned the following values:

- Username: bob
- API Key: 6eb6f07fd09b18dd61dd353dfb669820e7859cd3 (The API Key copied from the EM system)

Calculate timestamp and SHA-1 authorization values for the user (for example, bob) as follows:

1. Use the formula below to calculate timestamp (ts):

   ts=echo $(($(date +%s%N)/1000000)) = 1457033811032

- ts: 1457033811032 (value represents GMT date and time, Thursday, March 3, 2016, 7:36:51.032 PM)

2. For example, in Linux, use the following command to calculate SHA-1 authorization. *SHA1="$(echo -n "$username$apikey$ts" | sha1sum -t | awk '{print $1}')".*
   - Authorization: e20ac2c963ccfacf23a1f70287286443820e66d1

For API authentication, the following headers must be sent along with the REST API call:

APIkey: **bob** (Note: The APIkey here is the username)
Authorization: **e20ac2c963ccfacf23a1f70287286443820e66d1**
ts: **1457033811032**

API Example:

```
:~$ curl -s --get -H "ApiKey:  bob" -H "Authorization:
e20ac2c963ccfacf23a1f70287286443820e66d1" -H "ts:1457033811032" -H "Accept:
application/xml" -H "Content-Type: application/xml" -k
https://em_ip_address/ems/api/org/em/v1/energy -v -k
* Hostname was NOT found in DNS cache
*   Trying 127.0.0.1...
* Connected to localhost (127.0.0.1) port 443 (#0)
* successfully set certificate verify locations:
*   CAfile: none
  CApath: /etc/ssl/certs
* SSLv3, TLS handshake, Client hello (1):
* SSLv3, TLS alert, Client hello (1):
* SSLv3, TLS handshake, Server hello (2):
* SSLv3, TLS handshake, CERT (11):
* SSLv3, TLS handshake, Server key exchange (12):
* SSLv3, TLS handshake, Server finished (14):
* SSLv3, TLS handshake, Client key exchange (16):
* SSLv3, TLS change cipher, Client hello (1):
* SSLv3, TLS handshake, Finished (20):
* SSLv3, TLS change cipher, Client hello (1):
* SSLv3, TLS handshake, Finished (20):
* SSL connection using ECDHE-RSA-AES256-GCM-SHA384
* Server certificate:
* subject: C=US; ST=California; L=Sunnyvale; O=Enlighted Inc.; OU=Energy
Manager
* start date: 2015-12-15 06:37:22 GMT
* expire date: 2040-12-08 06:37:22 GMT
* issuer: C=US; ST=California; L=Sunnyvale; O=Enlighted Inc.; OU=Energy Manager
* SSL certificate verify result: self signed certificate (18), continuing
anyway.
> GET /ems/api/org/facility/v1/energy/1 HTTP/1.1
> User-Agent: curl/7.35.0
> Host: localhost
> ApiKey:  bob
> Authorization: e20ac2c963ccfacf23a1f70287286443820e66d1
> ts:  1457033811032
> Accept: application/xml
> Content-Type: application/xml
>
< HTTP/1.1 200 OK
< Date: Thu, 03 Mar 2016 19:42:55 GMT
* Server Apache-Coyote/1.1 is not blacklisted
< Server: Apache-Coyote/1.1
< Content-Type: application/xml
```

```
< Content-Length: 234
< Set-Cookie: JSESSIONID=8CE1F42FCBCCEE5AB69175D45673951F; Path=/ems/; HttpOnly
< Via: 1.1 127.0.0.1
< Vary: Accept-Encoding
```

Chapter 3: Energy Manager APIs

The Energy Manager (EM) stores sensor data, occupancy status, and fault data from fixtures in a building. The EM APIs allow users to monitor energy consumption, BACnet health, and turn lights on during an emergency.

Get Energy Manager Aggregate Energy Consumption

Retrieves the aggregate energy consumption in watt-hour for all fixtures and plugloads in the building for the last 15 minutes managed by the Energy Manager (EM). The response is a JSON element with energy-lighting and energy-plugload parameters.

GET https: //{em_ip_address_or_hostname}/ems/api/org/em/v1/energy/

Parameters

Request	
em_ip_address or hostname	The Energy Manager's IP address or hostname
Response	
energy-lighting	Reports aggregate energy consumption in watt-hour (Wh) for the last 15 minutes for all lighting fixtures managed by the EM
energy-plugload	Reports aggregate energy consumption in watt-hour (Wh) for the last 15 minutes for all plugloads managed by the EM

Sample Code

Using the EM IP address, obtain the aggregate energy consumption for all lighting fixtures and plugloads in the building.

```
Get https://{192.80.3.2}/ems/api/org/em/v1/energy/

HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: application/json
Content-Length: 7582
Date: Sat, 21 Jul 2015 17:06:51 GMT
{
  "energy-lighting": "21550.96"
  "energy-plugload": "3650.64"
}
```

Set Energy Manager Emergency

During an emergency, this command automatically sets the lighting level for all fixtures controlled by the EM to the maximum level of 100% or full on, so the floor is entirely lit up.

POST https://{em_ip_address_or_hostname}/ems/api/org/em/v1/setEmergency?time=60

Parameter

Request	
em_ip_address or hostname	The Energy Manager's IP address or hostname
time	The emergency duration time. Set to 60 minutes by default
Response	
status	0 – Lights are on to a maximum level of 100% or full on 1 – Lights are off

Sample Code

Using the EM IP address, set all lights to full on for 60 minutes during an emergency.

```
Post https://{192.80.3.2}/ems/api/org/em/v1/setEmergency?time=60

HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: application/json
Content-Length: 7582
Date: Sat, 21 Jul 2015 17:06:51 GMT
{
  "status": "0"
}
```

Get BACnet Health

The Enlighted Energy Manager integrates seamlessly with the Building Management System (BMS) via the BACnet interface. Use this API to poll the Energy Manager to determine if BACnet is running or not running.

GET https://{em_ip_address_or_hostname}/ems/api/org/bacnet/health

Parameters

Request	
em_ip_address or hostname	The Energy Manager's IP address or hostname
Response	
status	0 – BACnet is running 1 – BACnet not running

Sample Code

Poll the EM with the IP address to see if BACnet is running. The response '0' indicates BACnet is running.

```
Get https://{192.80.3.2}/ems/api/org/bacnet/health

HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: application/json
Content-Length: 7582
Date: Sat, 21 Jul 2015 17:06:51 GMT
{
  "status": "0"
}
```

Chapter 4: Organization APIs

In the Energy Manager, the organization hierarchy includes campuses, buildings, floors, and areas. The organization can have one or more campuses. Each campus can have one or more buildings with each building comprising one or more floors. The Organization APIs provide methods that allow you to view an organization's details including the list of campuses, buildings, and floors.

Get Organization Details

Returns the organization's details set up at the time of Energy Manager installation. The data returned includes the list of all campuses with identifiers for the campuses in the organization. The company and campus IDs returned are used as the primary identifiers in other API calls.

GET https://{em_ip_address_or_hostname}/ems/api/org/company

Parameters

Request	
em_ip_address or hostname	The Energy Manager's IP address or hostname
Response	
id	Organization identifier from Energy Manager (EM)
name	Organization name as described in the Energy Manager
address	Organization address
contact	Organization contact phone information
email	Email address of the contact person in the company
completionstatus	For internal use
validdomain	The domain name for the company
price	Electricity rate in cents per kWh
timezone_name	Timezone information for the specified company location
campus	List of campuses in the company along with details for each campus
title 24 compliance	Enabled if the site is geographically located where Title 24 is required
latitude	Company's latitude coordinate
longitude	Company's longitude coordinate

Sample Code

Using your EM IP address, send a request to return the list of all campuses for the organization. The sample response includes the company ID, company name, along with the contact details for the organization. The command also returns the list of campuses. In the example below, the company, Acme Inc., has two campuses, Hampton and Monroe. The response displays organization details for Acme Inc., along with details of both the campuses.

```
Get https://{192.80.3.2}/ems/api/org/company

HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: application/json
Content-Length: 7582
Date: Sat, 21 Jul 2015 17:06:51 GMT

{
```

```
    "id": "1",
    "name": "Acme Inc.",
    "address": "Sunnycreek, CA\r\nUSA",
    "contact": "408 555-1212",
    "email": "peter@acme.com",
    "completionstatus": "3",
    "validdomain": "acme.com",
    "price": "0.15",
    "timezone_name": "America/Los_Angeles",
    "campus": [
        {
            "id": "3",
            "name": "Hampton Campus",
            "calcMethod": "SENSOR_BASED",
            "contractBaseline": "0.00"
        },
        {
            "id": "5",
            "name": "Monroe Campus",
            "calcMethod": "SENSOR_BASED",
            "contractBaseline": "0.00"
        },

    ],
    "ntpEnable": "Y",
    "ntpServers":
"0.us.pool.ntp.org,1.us.pool.ntp.org,2.us.pool.ntp.org,3.us.pool.ntp.org",
    "title24": {
        "compliance": {
            "flag": "No"
        },
          ……
        "formtype": ""
    },
    "latitude": "37.39428",
    "longitude": "-122.03806"
}
```


Get All Campuses

Returns the list of all campuses in the organization.

GET https://{em_ip_address_or_hostname}/ems/api/org/campus/list/1

Parameters

Request	
em_ip_address or hostname	The Energy Manager's IP address or hostname
Response	
id	Campus identifier from the Energy Manager (EM)
name	Campus name as described in the Energy Manager
calcMethod	Energy savings calculated based on the sensor's baseline energy or contract baseline energy. SENSOR_BASED – Savings based on the sensor's baseline energy CONTRACTUAL – Savings based on contract baseline energy. The contract baseline energy is calculated using the annual baseline energy savings stipulated in the customer's signed agreement for the campus at the time of installation.
contractBaseline	Baseline energy savings for the campus.

Sample Code

In the example, the details of both campuses for Acme Inc., are returned when you send a request with the EM IP address.

```
Get https://{192.80.3.2}/ems/api/org/campus/list/1

HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: application/json
Content-Length: 7582
Date: Sat, 21 Jul 2015 17:06:51 GMT
{
    "campus": [

        {
            "id": "3",
            "name": "Hampton Campus",
            "calcMethod": "SENSOR_BASED",
            "contractBaseline": "0.00"
        },
        {
            "id": "5",
            "name": "Monroe Campus",
            "calcMethod": "SENSOR_BASED",
            "contractBaseline": "0.00"
        }
    ]
}
```

Get All Buildings

Returns the list of all buildings in the campus for the organization. The data returned includes all buildings with building identifiers and details.

GET https://{em_ip_address_or_hostname}/ems/api/org/building/list/1

Parameters

Request	
em_ip_address or hostname	The Energy Manager's IP address or hostname
Response	
id	Building identifier from the Energy Manager (EM)
name	Building name as described in the Energy Manager
useOrgLocation	For internal use
visible	For internal use
useOrgLocationTz	For internal use
energySaveCalcMethod	Energy savings calculated based on the sensor's baseline energy or contract baseline energy. SENSOR_BASED – Savings based on the sensor's baseline energy CONTRACTUAL – Savings based on contract baseline energy. The contract baseline energy is calculated using the annual baseline energy savings stipulated in the customer's signed agreement for the campus at the time of installation.
contractBaseline	Baseline energy savings for the campus.

Sample Code

Using your EM IP address, send a request to return the list of all buildings in the campus. In the example below, the campus has a single building called 'Main Building' and its details are returned as shown.

```
Get https://{192.80.3.2}/ems/api/org/building/list/1

HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: application/json
Content-Length: 7582
Date: Sat, 21 Jul 2015 17:06:51 GMT

{
    "building": [
{
        "id": "1",
        "name": "Main Building",
        "useOrgLocation": "true",
        "visible": "true",
        "useOrgLocationTz": "true",
        "energySaveCalcMethod": "SENSOR_BASED",
        "contractBaseline": "0.00"
    }
 ]
}
```

Get All Floors

Returns the list of all floors in the building of an organization. The Floor ID returned for each floor in the building is used as the primary identifier in other API calls.

GET https://{em_ip_address_or_hostname}/ems/api/org/floor/list

Parameters

Request	
em_ip_address or hostname	The Energy Manager's IP address or hostname
Response	
Id**	Floor identifier. (**used as the primary identifier in other API calls)
name	Floor name as described in the Energy Manager
building	Building identifier
campus	Campus identifier
company	Company identifier. By default, Company ID is always '1'.
description	Floor description; this field will be empty if it is not described in the EM
floorPlanUrl	Floor plan image name. If the floor plan does not have a name, it is returned as *default_floor_plan.gif*

Sample Code

Using your EM IP address, send a request to return the list of all floors in the building. For example, if the Main building in Monroe campus has three floors: First Floor, Second Floor, and Third Floor, the response will include details for all three floors. Note that the Floor ID returned for each floor is used as the primary identifier in other API calls.

```
Get https://{192.80.3.2}/ems/api/org/floor/list

HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: application/xml
Content-Length: 7582
Date: Sat, 21 Jul 2015 17:06:51 GMT
{
  "floor": [
    {
      "id": "1",
      "name": "First Floor",
      "building": "1",
      "campus": "1",
      "company": "1",
      "description": "sales and marketing",
      "floorPlanUrl": "entire 930 floor plan.jpg"
    },
    {
      "id": "2",
      "name": "Second Floor",
      "building": "1",
      "campus": "1",
      "company": "1",
      "description": "product",
      "floorPlanUrl": "entire 930 floor plan.jpg"
    },
    {
      "id": "3",
      "name": "Third Floor",
      "building": "1",
      "campus": "1",
      "company": "1",
      "description": "CAD",
      "floorPlanUrl": "Floorplan140x80a.png"
    }
  ]
}
```

Get Floor Plan Image

Returns the floor plan image for a particular floor.

Request

GET https://{em_ip_address_or_hostname}/ems/api/org/floor/{floor_id}

Parameters

Request	
em_ip_address or hostname	The Energy Manager's IP address or hostname
floor_id	Floor identifier. Use the Floor ID returned in *Get All Floors* API, or in EM, select *Facility > Floor >* in the left panel, and click the *Settings* tab to obtain the Floor ID.
Response	
Floor plan image	Displays the floor plan image

Chapter 5: Area APIs

An area on a floor is a grouping of a subset of devices including gateways, sensors, Enlighted Room Control (ERC) switches, plugloads, etc., on a floor. Areas are assigned on a floor based on usage or a specific purpose profile of the area in EM.

Get all Areas

In the Energy Manager, the organization hierarchy includes campuses, buildings, floors, and areas. This API returns details for all areas on a floor. The Area ID returned for each area is used as the primary identifier in other API calls.

Request

GET https://{em_ip_address_or_hostname}/ems/api/org/area/v2/list/{floor_id}

Parameters

Request	
em_ip_address or hostname	The Energy Manager's IP address or hostname
floor_id	Floor identifier. Use the Floor ID returned in the *Get All Floors* API, or in EM, select *Facility > Floor >* in the left panel, and click the *Settings* tab to obtain the Floor ID.
Response	
Id**	Area identifier (**used as the primary identifier in other API calls)
name	Area name as described in the EM
description	Area description as described in the EM. The field will be empty if there is no description entered in the EM.
zonesensorenable	The area has been activated as a zone sensor area. The area starts reporting consolidated occupancy status information from individual sensors. Note that the zone sensor license must be purchased and available for the sensors to report occupancy information. Otherwise, occupancy data is not reported by the sensors. true – Sensors in the area are enabled as zone sensors. false – Sensors in the area are disabled as zone sensors.

Sample Code

Send a request to return the list of all areas on the first floor with ID '1'. For example, if the first floor has two areas: Hardware and Software, the details of the areas along with their corresponding Area IDs '9' and '27' will be returned as shown below. The Area IDs are used as the primary identifiers in other API calls.

```
Get https://{192.80.3.2}/ems/api/org/area/v2/list/1

HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: application/xml
Content-Length: 7582
Date: Sat, 21 Jul 2015 17:06:51 GMT

{
  "area": [
   {
{
  "id": "9",  ** Area ID**
```

```
  "name": "Hardware",
  "description": "hw eng",
  "zonesensorenable": "true",
},
{
  "id": "27", **Area ID**
  "name": "Software",
  "description": "sw eng",
  "zonesensorenable": "false",
  }
 ]
}
```

Get Area Energy Consumption

Returns the aggregate energy consumption in watt-hour for all devices in the area for the last 15 minutes.

Request

GET https://{em_ip_address_or_hostname}/ems/api/org/area/v1/energy/{area_id}

Parameters

Request	
em_ip_address or hostname	The Energy Manager's IP address or hostname
area_id	Area identifier. Use the Area ID returned in the *Get all Areas* API, or in EM, select *Facility > Floor > Area* in the left panel, and click the *Settings* tab to get the Area ID.
Response	
energy-lighting	Reports aggregate energy consumption in watt-hour for the last 15 minutes for all lighting fixtures in the area
energy-plugload	Reports aggregate energy consumption in watt-hour for the last 15 minutes for all plugloads in the area

Sample Code

For example, the *Get all Areas* API returned two areas: Hardware area with ID '9' and Software area with ID '27'. Send a request to the EM to return the watt-hour for all devices in the area for the last 15 minutes for Area ID '9'.

```
Get https://192.80.3.2/ems/api/org/area/v1/energy/9

HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: application/json
Content-Length: 7582
Date: Sat, 21 Jul 2015 17:06:51 GMT

{
  "energy-lighting": "5580.96"
  "energy-plugload": "380.63"
}
```

Get Area Occupancy

The sensors capture and stream occupancy data to the Energy Manager. The Get Area Occupancy API returns the current occupancy status of the area. The occupancy status is provided only when the sensors in the area are enabled as zone sensors to report occupancy data.

For the sensors to start reporting occupancy data to the EM, you will need zone sensor licenses based on the number of areas configured in the system. For example, if you have 12 areas to be monitored for zone occupancy, you will need 12 zone licenses. Refer to the Chapter *Procuring License* in the *Energy Manager Admin Guide* to purchase zone sensor licenses.

After you have purchased and uploaded the zone license file in EM, the sensors in the area on the floor plan must be enabled as zone sensors. When an area has been enabled as a zone sensor area, the zone sensors start reporting occupancy status information for the area. Otherwise, the occupancy data is not reported by the sensors in the area. Refer to the sections *Adding an Area* and *Activating Zone Sensors in the Area* in the *Energy Manager Admin Guide* to add areas on the floor plan and assign sensors to a zone.

Request

GET https://{em_ip_address_or_hostname}/ems/api/org/area/v1/occ/{area_id}

Parameters

Request	
em_ip_address or hostname	The Energy Manager's IP address or hostname
area_id	Area identifier. Use the Area ID returned in the *Get all Areas* API, or in EM, select *Facility > Floor > Area* in the left panel, and click the *Settings* tab to get the Area ID.
Response	
occupancyState	0 – Area unoccupied 1 – Area occupied -1 – Fault. Sensors are not reporting occupancy -2 – Zone sensor license is not available. Sensors monitoring the occupancy status of the zone is disabled. -3 – Unknown area

Sample Code

For example, the *Get all Areas* API returned two areas: Hardware area with ID '9' and Software area with ID '27'. To get the area occupancy for Hardware area, use the EM IP address and Area ID '9' in the API command. The occupancy state for the Hardware area is returned as unoccupied.

```
Get https://192.80.3.2/ems/api/org/area/v1/occ/9

HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: application/json
Content-Length: 7582
Date: Sat, 21 Jul 2015 17:06:51 GMT

{
  "occupancyStatus":
    {"occupancyState": "0"}
}
```

Get Area Occupancy for a Floor

The Get Area Occupancy for a Floor returns the current occupancy status of all areas on a floor. The occupancy status is provided only when the sensors in the area are enabled as zone sensors to report occupancy data.

Request

GET https://{em_ip_address_or_hostname}/ems/api/org/facility/v1/getOccupancyStateofFloorAreas/{floor_id}

Parameters

Request	
em_ip_address or hostname	The Energy Manager's IP address or hostname
floor_id	Floor identifier. Use the Floor ID returned in the *Get All Floors* API, or in EM, select *Facility > Floor >* in the left panel, and click the *Settings* tab to obtain the Floor ID.
Response	
Id	Area identifier
name	Area name as described in the EM
description	Area description as described in the EM. The field will be empty if there is no description entered in the EM.
zonesensorenable	The area has been activated as a zone sensor area. The area starts reporting consolidated occupancy status. Note that the zone sensor license must be purchased and available for the sensors to report occupancy information. Otherwise, occupancy data is not reported by the sensors in the area. true – Sensors in the area are enabled as zone sensors. false – Sensors in the area are disabled as zone sensors.
occupancyState	0 – Area unoccupied 1 – Area occupied -1 – Fault. Sensors are not reporting occupancy.

Note: For this API, you will **always** get an additional area. Its <id> field will ALWAYS be "0". Its name will ***always*** be "Unassigned". The value of <occupancyState> will be "1" if ANY sensor on that floor is NOT in an area, but its current state is "occupied". If all such sensors are "vacant", then this value will be "0".

Sample Code

Send a request to return the occupancy state for all areas on first floor with ID ‘1’. For example, the first floor with ID ‘1’ in the Main building has two areas: Hardware and Software. The details of the areas along with their corresponding Area IDs and occupancy state for all areas on the floor are returned.

```
Get https://{192.80.3.2}/ems/api/org/facility/v1/getOccupancyStateofFloorAreas/1

HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: application/json
Content-Length: 7582
Date: Sat, 21 Jul 2015 17:06:51 GMT


{
   “area”: [
  {
   "id": "9",  ** Area ID**
   "name": "Hardware",
   "description": "hw eng",
   "zonesensorenable": "true"
   “occupancyState”: “1”,
  },
  {
   "id": "27", **Area ID**
   "name": "Software",
   "description": "sw eng",
   "zonesensorenable": "false",
   “occupancyState”: “1”,
  },
  {
   "id": "0",
   "name": "Unassigned",
   "description": "Unassigned area",
   “occupancyState”: “1”,
  }
 ]
}
```

Get Area Outage

Returns the number of fixtures not functioning in the area along with the total number of fixtures available in the area.

Request

GET https://{em_ip_address_or_hostname}/ems/api/org/area/v1/out/{area_id}

Parameters

Request	
em_ip_address or hostname	The Energy Manager's IP address or hostname
area_id	Area identifier. Use the Area ID returned in the *Get all Areas* API, or in EM, select *Facility > Floor > Area* in the left panel, and click the *Settings* tab to get the Area ID.
Response	
totalSensors	Total number of fixtures in the area
outSensors	Number of fixtures not functioning in the area

Sample Code

For example, the *Get all Areas* API returned two areas: Hardware area with ID '9' and Software area with ID '27'. Send a request to return the total number of fixtures and number of fixtures not functioning in the area with ID '9'. The sample example below shows that of the ten fixtures in the area, one fixture is not functioning.

```
Get https://192.80.3.2/ems/api/org/area/v1/out/9

HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: application/json
Content-Length: 7582
Date: Sat, 21 Jul 2015 17:06:51 GMT

{
  "totalSensors":"10"
  "outSensors":"1"
}
```

Set Area Emergency

During an emergency, this command automatically sets the lighting level of all fixtures in the area to a maximum level of 100% (full on), so the area is lit up.

Request

POST https://{em_ip_address_or_hostname}/ems/api/org/area/v1/setEmergency/{area_id}?time=60

Parameters

Request	
em_ip_address or hostname	The Energy Manager's IP address or hostname
area_id	Area Identifier. Use the Area ID returned in the *Get all Areas* API, or in EM, select *Facility > Floor > Area* in the left panel, and click the *Settings* tab to get the Area ID.
time	The emergency duration time. The default emergency time is set to 60 minutes.
Response	
status	0 – Lights are on to a maximum level of 100% or full on 1 -- Lights are off

Sample Code

For example, the *Get all Areas* API returned two areas: Hardware area with ID '9' and Software area with ID '27'. Using the EM IP address and Area ID, set all lights in the area with ID '9' to turn on for 60 minutes.

```
Post https://192.80.3.2/ems/api/org/area/v1/setEmergency/9?time=60

HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: application/json
Content-Length: 7582
Date: Sat, 21 Jul 2015 17:06:51 GMT
{
  "status": "0"
}
```

Chapter 6: Fixture APIs

Fixtures or sensors are a subset of devices, and sensors that use the same lighting configuration belong to a profile. Profiles define the configuration and lighting behavior of the sensors.

Get Sensor Details by Floor

Returns sensor details for all sensors on the floor for a given period at every five-minute interval. For example, if the duration is from 10:00 am to 10:10 am, data is sent at every five-minute interval. The API retrieves data for a maximum period of one hour.

The Fixture ID that is returned is used as the primary identifier in other API calls.

Request

GET https://{em_ip_address_or_hostname}/ems/api/org/sensor/v2/stats/floor/{floorId}/{from_date}/{to_date}

Parameters

Request	
em_ip_address or hostname	The Energy Manager's IP address or host name
floor_id	Floor identifier. Use the Floor ID returned in the *Get All Floors* API, or in EM, select *Facility > Floor >* in the left panel, and click the *Settings* tab to obtain the Floor ID.
from_date	Start date to display energy consumption data Format: (yyyyMMddHHmm) year/month/day/hour/minutes
to_date	End date to stop displaying energy consumption data Format: (yyyyMMDDHHmm) year/month/day/hour/minutes
Response	
Id**	Fixture identifier (**used as the primary identifier in other API calls)
power	Average load of the fixture in watts during the last five-minute interval
temperature	Temperature in Fahrenheit
lightLevel	Ambient light level reading at the sensor (lux)
captureTime	Five-minute time duration represented as (yyyy-MM-dd HH:mm) year-month-day-hour-minutes

Sample Code

Request for sensor details for all sensors on the first floor with Floor ID '1' from 10 am through 11:30 am on Feb. 12, 2019. Data will be sent at every five-minute interval, which will be 10:05 am and 10:10 am, and so on. The Fixture ID returned for each sensor is used as the primary identifier in other API calls.

```
Get
https://192.80.3.2/ems/api/org/sensor/v2/stats/floor/1/20190212010000/20190212021130

HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: application/xml
Content-Length: 7582
Date: Sat, 21 Jul 2015 17:06:51 GMT
{
  "sensor": [
    {
      "id": "28",                  **Fixture ID**
      "power": "27.98",
```

```
      "temperature": "78",
        "lightLevel":"99"
        "captureTime": "2019-02-12 10:05"
    },
    {
      "id": "29",            **Fixture ID**
      "power": "12.62",
      "temperature": "74",
      "lightLevel":"50"
      "captureTime": "2019-02-12 10:05"
    },
   "sensor": [
    {
      "id": "1327",
      "power": "27.98",
      "temperature": "74",
      "lightLevel":"99"
      "captureTime": "2019-02-12 10:10"
    },
    {
      "id": "1377",
      "power": "12.62",
      "temperature": "70",
      "lightLevel":"50"
      "captureTime": "2019-02-12 10:10"
 …..
  ]
}
```


Get Sensor Details

This API reports information for an individual sensor at every five-minute interval. The data for the last five minutes from the sensor is reported.

0 — 5 — 10 — 15

make call @ 11 data from 6-11 or 5-10?

Request

GET https://{em_ip_address_or_hostname}/ems/api/org/sensor/v3/details/{fixture_id}

Parameters

Request	
em_ip_address or hostname	The Energy Manager's IP address or host name
fixture_id	Sensor identifier. Use the Sensor ID returned by the *Get Sensor Details by Floor* API or in EM, select *Facility > Floor* in the left panel, *Floor Plan* tab, and double-click the sensor icon in the floor plan to obtain the Sensor ID.
Response	
id	Sensor identifier
name	Sensor name. For example, *Sensorxxxxxx*
power	The average load of the fixture in watts for the last five-minute interval
temperature	Temperature in Fahrenheit
lightLevel	Returns the light level (0-100%) information of the fixture 0% - Light level is set to off 100% - Light level is set to full on

Sample Code

Using the sensor ID returned by the *Get Sensor Details by Floor* API, request for data such as power, temperature, and light level settings from the sensor with ID '28'. For example, request data for the last five minutes for sensor with ID '28'.

```
Get https://192.80.3.2/ems/api/org/sensor/v3/details/28

HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: application/xml
Content-Length: 7582
Date: Sat, 21 Jul 2015 17:06:51 GMT

{
 "sensor":[
 {
  "id": "28",
  "name": "Sensor00ce56",
  "power": "69.07",
  "temperature": "74",
  "lightLevel": "50"
  }
 ]
}
```

Get Sensor Location by Floor

Returns the sensor location for all sensors on the floor. The data is returned as X, Y coordinates of the measured and scaled down distance from the lower-left corner of the uploaded floor plan.

Request

GET https://{em_ip_address_or_hostname}/ems/api/org/fixture/location/list/floor/{floor_id}/1

Parameters

Request	
em_ip_address or hostname	The Energy Manager's IP address or host name
floor_id	Floor identifier. Use the Floor ID returned in the *Get All Floors* API, or in EM, select *Facility > Floor >* in the left panel, and click the *Settings* tab to obtain the Floor ID.
Response	
id	Fixture identifier
name	Sensor name. For example, *Sensorxxxxxx*
xaxis	Scaled reference of the distance measured in the horizontal dimension from the lower-left corner of the uploaded floor plan
yaxis	Scaled reference of the distance measured in the vertical dimension from the lower-left corner of the uploaded floor plan
macaddress	Sensor's MAC address
groupid	Profile group identifier. The ID of the profile associated with the fixture.
Ignore the remaining parameters displayed	

Sample Code

Request the x- and y-axis coordinates for all sensors on Floor 1 with ID '1'.

```
Get https://192.80.3.2/ems/api/org/fixture/location/list/floor/1/1

HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: application/xml
Content-Length: 7582
Date: Sat, 21 Jul 2015 17:06:51 GMT

{
  "fixture": [
    {
      "id": "361",
      "name": "Sensor00c017",
      "xaxis": "966",
      "yaxis": "391",
      "macaddress": "0:c0:17",
      "groupid": "185",
      …
      …
    },
    {
      "id": "1364",
      "name": "Sensor0629dd",
      "xaxis": "270",
      "yaxis": "90",
      "macaddress": "6:29:dd",
      "groupid": "196",
      …
      …

    },
    {
      "id": "835",
      "name": "Sensor013ec0",
      "xaxis": "756",
      "yaxis": "462",
      "macaddress": "1:3e:c0",
      "groupid": "55",
      …
      …

    }
  ]
}
```

Get All Fixtures by Area

The API returns fixture location for all fixtures in an area. The data is returned as X, Y coordinates of the measured and scaled down distance from the lower-left corner of the uploaded floor plan.

Request

GET https://{em_ip_address_or_hostname}/ems/api/org/fixture/v1/location/list/area/{area_id}

Parameters

Request	
em_ip_address or hostname	The Energy Manager's IP address or hostname
area_id	Area Identifier. Use the Area ID returned in the *Get all Areas* API, or in EM, select *Facility > Floor > Area* in the left panel, and click the *Settings* tab to get the Area ID.
Response	
id	Sensor identifier
name	Sensor name. For example, *Sensorxxxxxx*
xaxis	Scaled reference of the distance measured in the horizontal dimension from the lower-left corner of the uploaded floor plan
yaxis	Scaled measurement of distance in the vertical dimension from the lower-left corner of the uploaded floor plan
macaddress	Sensor's MAC address
groupid	Profile group identifier. The ID of the profile associated with the fixture.

Sample Code

Request the x- and y-axis coordinates for all sensors in the area with ID '9'.

```
Get https://192.80.3.2/ems/api/org/fixture/v1/location/list/area/9

HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: application/xml
Content-Length: 7582
Date: Sat, 21 Jul 2015 17:06:51 GMT

{
  "fixture": [
    {
      "id": "361",
      "name": "Sensor00c017",
      "xaxis": "966",
      "yaxis": "391",
      "macaddress": "0:c0:17",
      "groupid": "185",
      …
      …

    },
    {
      "id": "1364",
      "name": "Sensor0629dd",
      "xaxis": "270",
      "yaxis": "90",
      "macaddress": "6:29:dd",
      "groupid": "196",
```

```
      …
      …

    },
    {
      "id": "835",
      "name": "Sensor013ec0",
      "xaxis": "756",
      "yaxis": "462",
      "macaddress": "1:3e:c0",
      "groupid": "55",
      …
      …

    }
  ]
```


Get Sensor Profiles

The API returns the current and applicable profiles that can be assigned to the sensor. The list of applicable profiles displayed is derived from the same profile template as the current profile. The profile group ID returned is used as the primary identifier in other API calls.

Request

GET https:/{em_ip_address_or_hostname}/ems/api/org/fixture/v1/getFixtureApplicableProfiles/{fixture_id}

Parameters

Request	
em_ip_address or hostname	The Energy Manager's IP address or hostname
fixture_id	Fixture or sensor identifier. Use the Fixture ID returned by the *Get Sensor Details by Floor* API or in EM, select *Facility > Floor* in the left panel, *Floor Plan* tab, and double-click the fixture icon in the floor plan to obtain the Fixture ID.
Response	
id	Fixture or sensor identifier
name	Sensor name. For example, *Sensorxxxxxx*
currentProfile	Displays the name of the current profile assigned to the sensor.
groupId**	Profile group identifier. The ID of the profile associated with the fixture. (**used as the primary identifier in other API calls)
applicableProfiles	Lists profiles derived from the same profile template as the current profile.

Sample Code

Using the Fixture ID returned by the *Get Sensor Details by Floor* API, request the current profile name assigned to the sensor. As shown in the response below, *Open Office Max 35* is the profile assigned to fixture with ID '28'. In addition to the profile assigned to the fixture, the API also returns the list of applicable profiles and the corresponding profile Group ID derived from *Open Office Max 35.* The Group ID is used as the primary identifier in other API calls.

```
Get https:/192.80.3.2/ems/api/org/fixture/v1/getFixtureApplicableProfiles/27

HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: application/json
Content-Length: 7582
Date: Sat, 21 Jul 2015 17:06:51 GMT

{
  "id": "28",                                **fixture-id**
  "name": "Sensor00ceed",                    **fixture name**
  "currentProfile": "Open Office Max 35",    **profile name**
  "groupId": "27"                            **groupID**
  "applicableProfiles":                      **list of applicable profiles**
  [
   {
     "name": "Open Office Max 35",
     "groupId": "27"
   },
   {
     "name": "Open Office Max 70",
     "groupId": "28"
   },
   {
     "name": "Open Office Max 50",
     "groupId": "29"
   },
   {
     "name": "default.Open Office_Normal",
     "groupId": "46"
   },
   {
     "name": "default.Open Office_AlwaysOn",
     "groupId": "47"
   },
   {
     "name": "default.Open Office_Dim",
     "groupId": "48"
   }
 ]
}
```

Assign Profile

Assigns a profile to the fixture. The fixture will operate with the new behavior.

Request

POST https://{em_ip_address_or_hostname}/ems/api/org/fixture/v1/op/assignProfile/{fixture_id}/{group_id}

Parameters

Request	
em_ip_address or hostname	The Energy Manager's IP address or host name
fixture_id	Fixture or sensor identifier. Use the Fixture ID returned in the *Get Sensor Details by Floor* or in EM, select *Facility > Floor* in the left panel, *Floor Plan* tab, and double-click the fixture icon in the floor plan to obtain the Fixture ID.
group_id	Profile group identifier to which the fixture needs to be assigned (returned in *Get Sensor Profiles* API)
Response	
status	0 – Profile assigned successfully to the sensor -1 – Profile not assigned to the sensor
Msg	Error message description. For example, "either fixture or group does not exist. Please provide valid inputs".

Sample Code

The sample code shows how to assign a profile to a fixture. Use the Fixture ID returned by the *Get Sensor Details by Floor,* and Group ID returned by the *Get Sensor Profiles* to assign a new profile to the fixture. For example, use the profile *Open Office Max 70* with GroupId '27' to assign it to the fixture with ID '28'. The response indicates that the profile has been assigned successfully to the fixture.

```
Post https://192.80.3.2/ems/api/org/fixture/v1/op/assignProfile/28/27

HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: application/json
Content-Length: 7582
Date: Sat, 21 Jul 2015 17:06:51 GMT
{
  "status": "0"
}
```

Get Sensor Energy Consumption

Reports energy consumed in watt-hour for the last 15 minutes by the fixture.

Request

GET https://{em_ip_address_or_hostname}/ems/api/org/fixture/v1/energy/{fixture_id}

Parameters

Request	
em_ip_address or hostname	The Energy Manager's IP address or host name
fixture_id	Fixture or sensor identifier. Use the Fixture ID returned in the *Get Sensor Details by Floor* or in EM, select *Facility > Floor* in the left panel, *Floor Plan* tab, and double-click the fixture icon in the floor plan to obtain the Fixture ID.
Response	
energy	Reports energy consumed in watt-hour for the last 15 minutes by the lighting fixture

Sample Code

Send a request to the EM using the Fixture ID '28' returned by the *Get Sensor Details by Floor* API to return the energy consumed in watt-hour for the last 15 minutes.

```
Get https://192.80.3.2/ems/api/org/fixture/v1/energy/28

HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: application/json
Content-Length: 7582
Date: Sat, 21 Jul 2015 17:06:51 GMT
{
  "energy": "50.96"
}
```

Get Sensor Energy Consumption by Area

Returns the aggregate energy consumed in watt-hour by all fixtures in the area for the last 15 minutes.

Request

GET https://{em_ip_address_or_hostname}/ems/api/org/fixture/v1/energy/area/{area_id}

Parameters

Request	
em_ip_address or hostname	The Energy Manager's IP address or hostname
area_id	Area Identifier. Use the Area ID returned in the *Get all Areas* API, or in EM, select *Facility > Floor > Area* in the left panel, and click the *Settings* tab to get the Area ID.
Response	
energy	Reports aggregate energy consumed in watt-hour for the last 15 minutes by the lighting fixtures in the area

Sample Code

Send a request to the EM using the Area ID '9' returned by the *Get all Areas* API to return the energy consumed in watt-hour for the last 15 minutes by all fixtures in the area.

```
Get https://192.80.3.2/ems/api/org/fixture/v1/energy/area/9

HTTP/1.1 200 OK

Server: Apache-Coyote/1.1
Content-Type: application/json
Content-Length: 7582
Date: Sat, 21 Jul 2015 17:06:51 GMT
{
  "energy": "250.96"
}
```

Manual Override for a Fixture

Allows you to manually override the fixture or sensor profile settings and dim the light in the fixture to an absolute level for a period of time in minutes.

Request

POST https://{em_ip_address_or_hostname}/ems/api/org/fixture/v1/op/dim/{mode}/{amount}?time=60

```
<fixtures>
  <fixture>
    <id>27</id>
  </fixture>
</fixtures>
```

Parameters

Request	
em_ip_address or hostname	The Energy Manager's IP address or host name
fixture	Fixture or sensor identifier. Use the Fixture ID returned in the *Get Sensor Details by Floor* or in EM, select *Facility > Floor* in the left panel, *Floor*

	Plan tab, and double-click the fixture icon in the floor plan to obtain the Fixture ID.
mode	Set mode option abs – Sets the mode to absolute light level dimming
amount	The percentage ranges from 0-100% light levels
time	The manual override duration time. The default time is set to 60 minutes.
Response	
status	0 – Light dimmed successfully 1 – Light has not dimmed

Sample Code

Manually override the sensor profile settings and dim the light to 50% for the fixture with ID '27' for a period of 60 minutes. The status shows that the fixture light was set to dim successfully.

```
Post https://192.80.3.2/ems/api/org/21/v1/op/dim/abs/50?time=60
<fixtures>
  <fixture>
    <id>27</id>
  </fixture>
</fixtures>

HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: application/json
Content-Length: 7582
Date: Sat, 21 Jul 2015 17:06:51 GMT
{
  "status": "0"
}
```

Auto Mode for a Fixture

Sets the sensors to Auto mode. In this mode, the level of occupancy in the area determines the fixture dim level.

Request

POST https://{em_ip_address_or_hostname}/ems/api/org/fixture/v1/op/auto

```
 <fixtures>
  <fixture>
    <id>27</id>
  </fixture>
</fixtures>
```

Parameters

Request	
em_ip_address or hostname	The Energy Manager's IP address or host name
fixture	Fixture or sensor identifier. Use the Fixture ID returned in the *Get Sensor Details by Floor* or in EM, select *Facility > Floor* in the left panel, *Floor Plan* tab, and double-click the fixture icon in the floor plan to obtain the Fixture ID.
Response	
status	0 – Auto mode successful; lights are dimmed based on occupancy

	1 – Auto mode unsuccessful

Sample Code

Sets the sensor with ID ‘27’ to auto mode. The response is successful which means lights are dimmed based on occupancy.

```
POST https://192.80.3.2/ems/api/org/21/v1/op/auto
<fixtures>
  <fixture>
    <id>27</id>
  </fixture>
</fixtures>

HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: application/json
Content-Length: 7582
Date: Sat, 21 Jul 2015 17:06:51 GMT
{
  "status": "0"
}
```

Chapter 7: Plugload APIs

A plugload switches power outlets on or off based on occupancy input received from the sensors.

Get Plugload Details

The API returns the plugload details.

Request

GET https:// {em_ip_address_or_hostname}/ems/api/org/plugload/v1/details/{plugload_id}

Parameters

Request	
em_ip_address or hostname	The Energy Manager's IP address or hostname
plugload_id	Plugload identifier. Use the Plugload ID returned in the *Get All Plugloads by Floor*, or in Energy Manager, select *Facility > Floor* in the left panel, *Floor Plan* tab, and double-click the plugload icon on the floor plan to obtain the Plugload ID.
Response	
id	Plugload identifier
name	Plugload name. For example, plugloadxxxxxx.
xaxis	Scaled measurement of distance in the horizontal dimension from the lower-left corner of the uploaded floor plan
yaxis	Scaled measurement of distance in the vertical dimension from the lower-left corner of the uploaded floor plan
location	Location of the plugload in the building
floorId	Floor identifier
campusid	Campus identifier
buildingid	Building identifier
macaddress	Plugload MAC address
version	Plugload version number
voltage	Plugload voltage
active	true – Plugload working false – Plugload not working
Ignore the remaining parameters displayed	

Sample Code

Using the Plugload ID '361' returned by the *Get All Plugloads by Floor* API, request EM for plugload details. The response includes the plugload details for the plugload.

```
Get https://192.80.3.2/ems/api/org/plugload/v1/details/361

HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: application/xml
Content-Length: 7582
Date: Sat, 21 Jul 2015 17:06:51 GMT
{
  "id": "361",
  "name": "Plugload06000c",
  "xaxis": "-38",
  "yaxis": "-2",
```

```
  "location": "Main campus -> HQ building -> Demo area",
  "floor": null,
  "floorId": "16",
  "campusid": "1",
  "buildingid": "1",
  "macaddress": "6:0:c",
  "version": "2.8.0 b2588",
  "voltage": "120",
…
  "active":"true"
  ……
}
```

Get All Plugloads by Floor

The API returns plugload location for all plugloads on a floor. The data is returned as X, Y coordinates of the measured and scaled down distance from the lower-left corner of the uploaded floor plan. The Plugload ID that is returned is used as the primary identifier in other API calls.

Request

GET https://{em_ip_address_or_hostname}/ems/api/org/plugload/v1/location/list/floor/{floor_id}/1

Parameters

Request	
em_ip_address or hostname	The Energy Manager's IP address or host name
floor_id	Floor identifier. Use the Floor ID returned in the *Get All Floors* API, or in EM, select *Facility > Floor >* in the left panel, and click the *Settings* tab to obtain the Floor ID.
Response	
Id**	Plugload identifier (**used as the primary identifier in other API calls)
name	Plugload name. For example, plugloadxxxxxx.
xaxis	Scaled measurement of distance in the horizontal dimension from the lower-left corner of the uploaded floor plan
yaxis	Scaled measurement of distance in the vertical dimension from the lower-left corner of the uploaded floor plan
macaddress	Plugload MAC address
groupid	Profile group identifier. The ID of the profile associated with the plugload.
Ignore the remaining parameters displayed	

Sample Code

Use the Floor ID '1' returned by the *Get All Floors* API to return the x- and y-axis coordinates of the scaled down distance from the lower-left corner of the floor plan for all plugloads on Floor 1.

```
Get https://192.80.3.2/ems/api/org/plugload/v1/location/list/floor/1/1

HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: application/xml
Content-Length: 7582
Date: Sat, 21 Jul 2015 17:06:51 GMT

{
  "plugload": [
```

```
    {
      "id": "361",
      "name": "Plugload00c017",
      "status": "0",
      "xaxis": "966",
      "yaxis": "391",
      "macaddress": "0:c0:17",
      "groupid": "185",
      …
    },
    {
      "id": "364",
      "name": "Plugload0629dd",
      "status":"1",
      "xaxis": "270",
      "yaxis": "90",
      "macaddress": "6:29:dd",
      "groupid": "196",
      ..
    },
    {
      "id": "835",
      "name": "Plugload013ec0",
      "status":"0",
      "xaxis": "756",
      "yaxis": "462",
      "macaddress": "1:3e:c0",
      "groupid": "55",
      ..
    }
  ]
}
```


Get All Plugloads by Area

The API returns the plugload location for all plugloads in an area. The data is returned as X, Y coordinates of the measured and scaled down distance from the lower-left corner of the uploaded floor plan.

Request

GET https://{em_ip_address_or_hostname}/ems/api/org/plugload/v1/location/list/area/{area_id}/1

Parameters

Request	
em_ip_address or hostname	The Energy Manager's IP address or host name
area_id	Area Identifier. Use the Area ID returned in the *Get all Areas* API, or in EM, select *Facility > Floor > Area* in the left panel, and click the *Settings* tab to get the Area ID.
Response	
id	Plugload identifier
name	Plugload name. Default plugload name is in the format *plugloadxxxxxx*.
xaxis	Scaled measurement of distance in the horizontal dimension from the lower-left corner of the uploaded floor plan
yaxis	Scaled measurement of distance in the vertical dimension from the lower-left corner of the uploaded floor plan
macaddress	Plugload MAC address
groupid	Profile group identifier. The ID of the profile associated with the plugload.

Ignore the remaining parameters displayed

Sample Code

For example, the *Get all Areas* API returned two areas: Hardware area with ID '9' and Software area with ID '27'. Request the EM for the x- and y- coordinates of the scaled down distance of the floor plan for all plugloads in the area with area ID '9'.

```
Get https://192.80.3.2/ems/api/org/plugload/v1/location/list/area/9/1

HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: application/xml
Content-Length: 7582
Date: Sat, 21 Jul 2015 17:06:51 GMT

{
  "plugload": [
    {
      "id": "361",
      "name": "Plugload00c017",
      "status": "0",
      "xaxis": "966",
      "yaxis": "391",
      "macaddress": "0:c0:17",
      "groupid": "185",
      …
    },
    {
      "id": "1364",
      "name": "Plugload0629dd",
      "status":"1",
      "xaxis": "270",
      "yaxis": "90",
      "macaddress": "6:29:dd",
      "groupid": "196",
      ..
    },
    {
      "id": "835",
      "name": "Plugload013ec0",
      "status":"0",
      "xaxis": "756",
      "yaxis": "462",
      "macaddress": "1:3e:c0",
      "groupid": "55",
      ..
    }
  ]
}
```

Get Plugload Energy Consumption

The API returns managed and unmanaged energy consumed for the last 15 minutes by the plugload.

Request

GET https://{em_ip_address_or_hostname}/ems/api/org/plugload/v1/energy/{plugload_id}

Parameters

Request	
em_ip_address or hostname	The Energy Manager's IP address or hostname
plugload_id	Plugload identifier. Use the Plugload ID returned in the *Get All Plugloads by Floor* API, or in Energy Manager, select *Facility* > *Floor* in the left panel, *Floor Plan* tab, and double-click the plugload icon on the floor plan to obtain the Plugload ID.
Response	
managed energy	Reports managed energy consumption in watt-hour by the plugload
unmanaged energy	Reports unmanaged energy consumption in watt-hour by the plugload

Sample Code

Send a request to the EM using Plugload ID '361' returned by the *Get All Plugloads by Floor* API to return the energy consumed in watt-hour for the last 15 minutes by the plugload.

```
Get https://192.80.3.2/ems/api/org/plugload/v1/energy/361

HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: application/json
Content-Length: 7582
Date: Sat, 21 Jul 2015 17:06:51 GMT
{
  "managed-energy": "50.96"
  "unmanaged-energy": "5.4"
}
```

Get Plugload Energy Consumption by Area

Returns the aggregate managed and unmanaged energy consumed in watt-hour by all plugloads in the area.

Request

GET https://{em_ip_address_or_hostname}/ems/api/org/plugload/v1/energy/area/{area_id}

Parameters

Request	
em_ip_address or hostname	The Energy Manager's IP address or hostname
area_id	Area Identifier. Use the Area ID returned in the *Get all Areas* API, or in EM, select *Facility* > *Floor* > *Area* in the left panel, and click the *Settings* tab to get the Area ID.
Response	
managed energy	Reports aggregate managed energy consumption in watt-hour by all the plugloads in the area

unmanaged energy	Reports aggregate unmanaged energy consumption in watt-hour by all the plugloads in the area

Sample Code

Send a request to the EM using Area ID '9' returned by the *Get all Areas* API to return the energy consumed in watt-hour for the last 15 minutes by all plugloads in the area.

```
Get https://192.80.3.2/ems/api/org/plugload/v1/energy/area/9

HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: application/json
Content-Length: 7582
Date: Sat, 21 Jul 2015 17:06:51 GMT
{
  "managed-energy": "150.96"
  "unmanaged-energy": "15.4"
}
```

Set Plugload Status

Sets the managed plugload status to ON or OFF.

Request

POST https://{em_ip_address_or_hostname}/ems/api/org/area/v1/setPlugloadStatus/{plugload_id}/{Status}

Parameters

Request	
em_ip_address or hostname	The Energy Manager's IP address or hostname
plugload_id	Plugload identifier. Use the Plugload ID returned in the *Get All Plugloads by Floor* API, or in Energy Manager, double-click the plugload icon in the floor plan to obtain the Plugload ID.
status	Set to '0' to turn off or '1' to turn on the plugload
Response	
status	0 – Plugload turned on 1 – Plugload turned off

Sample Code

Using the Plugload ID '361' returned by *Get All Plugloads by Floor,* set the plugload to ON. The response is successful which means the plugload is turned on.

```
Post https://192.80.3.2/ems/api/org/area/v1/setPlugloadStatus/361/0

HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: application/json
Content-Length: 7582
Date: Sat, 21 Jul 2015 17:06:51 GMT
{
  "status": "0"
}
```

Chapter 8: Conference Room APIs

The Conference Room APIs interact with the Energy Manager for controlling lights in a conference room. Typical scenes in a conference room include the use of a projector or board room meeting. Groups used for lighting control are called switch groups. A virtual switch is added to the floor plan when a Switch Group is created. A virtual switch controls the light levels of selected fixtures. Each set of light level settings for fixtures is called a scene. A maximum of six scenes can be created. Each fixture in a scene can have different light levels to achieve the desired effect. The following section describes details of the respective APIs along with their response values.

Get Switch Groups

Groups used for lighting control are called switch groups. This API returns all switch groups on a floor in the building. The response includes the Switch Group ID and Switch name for each switch group. The Switch name is used as the primary identifier in other API calls.

Request

GET https://{em_ip_address_or_hostname}/ems/api/org/switchgroups/list/{property}/{pid}

Parameters

Request	
em_ip_address or hostname	The Energy Manager's IP address or host name
property	Property type: Company, or Floor
pid	Company or Floor ID. By default, the Company ID is always '1'. For Floor ID, use the Floor ID returned in the *Get All Floors* API, or in EM, select *Facility* > *Floor* > in the left panel, and click the *Settings* tab to obtain the Floor ID.
Response	
id	Switch Group identifier in EM
name**	Switch name (**used as the primary identifier in other API calls)

Sample Code

Using the Floor ID returned in the *Get All Floors* API, request for the list of all switch groups on the floor. The Switch Group ID and Switch name for all switch groups on Floor 1 are returned. For example, if Floor 1 has two switch groups, the Switch Group ID and Switch name are returned as shown below. The Switch name is used as the primary identifier in other API calls.

```
Get https://192.80.3.2/ems/api/org/switchgroups/list/floor/1
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: application/json
Content-Length: 7582
Date: Sat, 21 Jul 2015 17:06:51 GMT

{
  "gemsGroups": {
    "gemsGroup": [
      {
        "id": "19",                **switch group identifier**
```

```
        "name": "SwitchGrp1"        **switch name**
      },
      {
        "id": "29",                 **switch group identifier**
        "name": " SwitchGrp2"       **switch name**
      }
    ]
  }
}
```

Get Switch Scenes

The API returns the virtual switch and scene details. A virtual switch is added to the floor plan in the Energy Manager when a Switch Group is created. A virtual switch controls the light levels of selected fixtures. Each set of light level settings for fixtures is called a scene. For example, in a conference room, typical scenes would include using a projector when the lights should be off (all off) or during a meeting when the lights should be on (all on).

The scene identifiers returned are used as the primary identifier in other API calls.

Request

GET https://{em_ip_address_or_hostname}/ems/api/org/switch/v1/getSwitchScenes/{floor_id}/{switch_name}

Parameters

Request		
em_ip_address or hostname	The Energy Manager's IP address or hostname	
floor_id	Floor identifier. Returned in *Get All Floors* API, or in EM, select *Facility* > *Floor* > in the left panel, and click the *Settings* tab to obtain the floor_id.	
switch_name	Switch name. Returned in *Get Switch Groups* API, or in EM, select *Facility* > *Floor* in the left panel, click *Floor Plan* tab, and hover the cursor over the switch icon to see the switch name.	
Response		
id	Switch Group Identifier to which the scene belongs.	
name	Switch name	
scenes	List of scenes. A maximum of six scenes can be created for a switch. For each scene, the following parameters are displayed.	
	Id**	Scene identifier (**used as the primary identifier in other API calls)
	switchid**	Virtual switch identifier that is created when a Switch Group is created. (**used as the primary identifier in other API calls) Note: The Virtual Switch ID and Switch Group Identifier are one-to-one mapped, but are two different entities.
	name	Scene name
	sceneOrder	Order of scenes on the associated Enlighted Room Control (ERC) switch

Sample Code

Using the switch name returned in the *Get Switch Groups* and Floor ID returned in *Get All Floors*, request EM to provide the virtual switch and scene details for the switch. For example, if the switch, SwitchGrp2, is configured for two scenes *All on*, *All off*, the following switch details are returned as shown below. The

scene identifiers (for example, 157, 158) returned for the scenes are used as the primary identifiers in other API calls.

```
Get https://192.80.3.2/ems/api/org/switch/v1/getSwitchScenes/1/SwitchGrp2

HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: application/json
Content-Length: 7582
Date: Sat, 21 Jul 2015 17:06:51 GMT

{
  "id":"29",           **switch Group identifier**
  "name":"SwitchGrp2", **switch name**
  "scenes":[
    {
      "id":"157",      **scene ID (primary id in other API calls)**
      "switchid":"1",  **switch ID (primary id in other API calls)**
      "name":"All On", **scene name**
      "sceneOrder":"1" **scene order**
    },
    {
      "id":"158",      **scene ID**
      "switchid":"1",  **switch ID***
      "name":"All Off",**scene name**
      "sceneOrder":"2" **scene order**
    }
  ]
}
```

Get Scene Light Levels for a Switch

When a switch group is created, a virtual switch is added to the floor plan, which controls the light levels of the selected fixtures. Each set of light level settings for fixtures is called a scene. A maximum of six scenes can be created. Each fixture in a scene can have different light levels to achieve the desired effect.
The API returns the scene light levels for the group of fixtures controlled by the switch.

Request

GET https://{em_ip_address_or_hostname}/ems/api/org/scene/v1/list/getSceneLevels/{switch_id}

Parameters

Request	
em_ip_address or hostname	The Energy Manager's IP address or hostname
switch_id	Switch identifier. Returned in *Get Switch Scenes* API.
Response	
id	Identifier for internal use
switchid	Switch identifier
sceneid	Scene identifier. Up to a maximum of six scenes can be defined
fixtureid	Fixture identifier of the fixture for which the scene applies
lightlevel	Returns the light level (0-100%) for each fixture for the scene. 0% - Light level is set to off 100% - Light level is set to full on.
scenecolor	Current color temperature value in Kelvin degress

Sample Code

Using the virtual Switch ID '1' returned by the *Get Switch Scenes* API, request the scene light levels for the fixtures. The API returns the light levels for the scene controlled by the switch with ID '1'.

```
Get https://192.80.3.2/ems/api/org/scene/v1/list/getSceneLevels/1

HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: application/json
Content-Length: 7582
Date: Sat, 21 Jul 2015 17:06:51 GMT

{
  "sceneLevel":[
  {
    "id": "6",             **identifier**
    "switchid": "1",       **switch identifier**
    "sceneid": "157",      **scene identifier for all on**
    "fixtureid": "885",    **fixture identifier**
    "lightlevel": "100"    **light level**
    "sceneColor": "3200"   **color temp value**
  }
  {
    "id": "6",            **identifier**
    "switchid": "1",      **switch identifier**
    "sceneid": "158",     **scene identifier for all off**
    "fixtureid": "886",   **fixture identifier**
    "lightlevel": "60"    **light level**
    "sceneColor": "3600" **color temp value**
  }
 ]
}
```

Apply Scene

Applies a scene to a group of fixtures controlled by the switch. Each set of light level settings for fixtures is called a scene. For example, in a conference room, typical scenes would include using a projector when the lights should be off (all off) or during a meeting when the lights should be on (all on).

Request

POST https://{em_ip_address_or_hostname} /ems/api/org/switch/v1/op/applyScene/{switch_id}/{scene_id} ?time=60

Parameters

Request	
em_ip_address or hostname	The Energy Manager's IP address or host name
switch_id	Switch identifier (returned in *Get Switch Scenes* API)
scene_id	Scene identifier (returned in *Get Switch Scenes* API)
time	The scene duration time. The default time is set to 60 minutes.
Response	
id	Scene identifier
status	0 – Success; Other than '0': Failure **Failure codes**: 1 – Switch not available 2 – Switch group not available

	3 – Fixtures and plugloads not available in the switch group 4 – List of associated fixtures and plugloads in the switch group empty -1 – Failure condition

Sample Code

Apply the scene 'All off' with Scene ID '158' to the Switch with ID '1' for a duration of one hour. Both the Scene ID and Switch ID are returned in the *Get Switch Scenes* API. The response '0' indicates that the 'All off' scene was applied to the switch successfully.

```
POST https://192.80.3.2/ems/api/org/switch/v1/op/applyScene/1/158?time=60

HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: application/json
Content-Length: 7582
Date: Sat, 21 Jul 2015 17:06:51 GMT
{
  "status": "0"
}
```

Manual Override for Switch Group

Allows you to manually override fixture profile settings associated with a switch and dim the fixtures for a period of time on the floor.

Request

POST https://{em_ip_address_or_hostname}/ems/api/org/switch/v1/op/dimGroup/{floor_id}/{switch_name}/{dimLevel}?time=60

Parameters

Request	
em_ip_address or hostname	The Energy Manager's IP address or host name
floor_id	Floor identifier. Returned in *Get All Floors* API, or in EM, select *Facility* > *Floor* > in the left panel, and click the *Settings* tab to obtain the Floor ID.
switch_name	Switch name. Returned in *Get Switch Groups* API. However, note that all switches on the floor are returned. The recommended path to determine a specific switch name in EM is select *Facility* > *Floor* in the left panel, click *Floor Plan* tab, and hover the cursor over the switch icon to see the switch name.
dimLevel	Specifies the dim level to which the fixtures in the group should be set to (0-100%)
time	The manual override duration time. The default time is set to 60 minutes.
Response	
status	0 – Override success 1 – Override failed

Sample Code

Manually override fixture profile settings on floor 1 associated with switch 'SwitchGrp1' (returned in the *Get Switch Groups* API) to dim the fixtures for 60 minutes using the EM IP address. In the example below, all fixtures controlled by the 'SwitchGrp1' switch on Floor 1 will be dimmed for 60 minutes.

```
Post https://192.80.3.2/ems/api/org/switch/v1/op/dimGroup/1/SwitchGrp1/50?time=60

HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: application/json
Content-Length: 7582
Date: Sat, 21 Jul 2015 17:06:51 GMT
{
  "status": "0"
}
```

Chapter 9: Data APIs

These APIs provide data such as energy savings, temperature, light level, and power for all sensors.

Get Sensor Energy Data by Floor

This API returns the aggregate energy consumption for all fixtures on the floor. When the specified time range in the request exceeds an hour, the API publishes aggregate data response every 15 minutes only for the first one-hour regardless of the time interval specified in the request. The response is always limited to four messages for the one-hour period. For example, if you request energy data for a two-hour period for up to ten sensors on the floor, the API returns the aggregate energy data for all sensors on the floor for the first one-hour publishing four response messages at a 15-minute interval.

Request

GET https://{em_ip_address_or_hostname}/ems/api/org/sensor/v2/sensorEnergyStats/15min/floor/{floorID}/{from_date}/{to_date}

Parameters

Request	
em_ip_address or hostname	The Energy Manager's IP address or hostname
floorID	Floor Identifier. Use the Floor ID returned in the *Get All Floors* API, or in EM, select *Facility > Floor* in the left panel, and click the *Settings* tab.
from_date	Start date to display energy consumption data Format: (yyyyMMddHHmmss) year/month/day/hour/minutes/seconds
to_date	End date to stop displaying energy consumption data Format: (yyyyMMDDHHmmss) year/month/day/hour/minutes/seconds
Response	
id	Sensor Identifier
name	Floor name
baseenergy	Displays the base energy of the fixture. Base energy is when the fixture's light level is always maintained at full-on without making any changes to the light level.
energy	Energy consumed by the fixture on the floor
savedEnergy	Energy saved by the fixture on the floor
occSavings	Energy savings for the fixture on the floor determined by occupancy nearby the fixture
ambientSavings	Ambient savings for the fixture on the floor based on the ambient light level measured by the fixture
tuneupSavings	Shows savings due to profiles set up for fixtures
manualSavings	Manual savings for the fixture on the floor based on when the fixture was controlled manually
dimLevel	Shows the dim level output of the fixtures (0-100%) 0% - Light level is set to off 100% - Light level is set to full-on

Sample Code

Using the Floor ID '1' returned by the *Get All Floors* API, request the aggregate energy consumption for fixtures on the floor for two hours from 5:30am through 7:30am on Feb. 12, 2019. The sample response shows details returned for fixtures on the floor for the first one-hour of the specified time interval.

```
GET
https://192.80.3.2/ems/api/org/sensor/v2/sensorEnergyStats/15min/floor/1/20190312043500
/20190314043500

{
  "sensorEnergyStatss": {
    "sensorEnergyStats": {
      "intervalStart":   "2019-02-12T04:35:00+05:30",
      "sensor": [
        {
          "id":     "4",
          "name":   "Sensor00c05f",
          "baseEnergy":   "21.6",
          "energy":       "0.00",
          "savedEnergy":  "21.60",
          "occSavings":   "16.20",
          "ambientSavings":"0.00",
          "tuneupSavings":"5.40",
          "manualSavings":"0.00",
          "dimLevel":     "0.0"
        },
        {
          "id":     "5",
          "name":   "Sensor00caa9",
          "baseEnergy":   "21.6",
          "energy":       "12.57",
          "savedEnergy":  "9.03",
          "occSavings":   "0.00",
          "ambientSavings":"0.00",
          "tuneupSavings":"9.03",
          "manualSavings":"0.00",
          "dimLevel":     "60.0"
        },
        …

      ]
    }
  }
}
```

Get Aggregate Sensor Energy Data by Floor

This API returns the aggregate energy consumption for all fixtures on the floor. When the specified time range in the request exceeds an hour, the API publishes aggregate data response every 15 minutes for the first one-hour regardless of the time interval specified in the request. The response is thus limited to four messages for the first one-hour period. For example, if you request energy data for a two-hour period for up to ten sensors on the floor, the API returns the aggregate energy data for all sensors on the floor only for the first one-hour publishing four response messages at a 15-minute interval.

Request

GET https://{em_ip_address_or_hostname}/ems/api/org/facility/v2/facilityEnergyStats/15min/floor/{floorID}/{from_date}/{to_date}

Parameters

Request	
em_ip_address or hostname	The Energy Manager's IP address or host name
floorID	Floor identifier. Use the *Get All Floors* API to obtain the Floor ID, or in EM, select *Facility* > *Floor* in the left panel, and click the *Settings* tab.
from_date	Start date to display energy consumption data Format: (yyyyMMddHHmmss) year/month/day/hour/minutes/seconds
to_date	End date to stop displaying energy consumption data Format: (yyyyMMDDHHmmss) year/month/day/hour/minutes/seconds
Response	
id	Floor identifier
name	Floor name
intervalStart	Time from when data is retrieved
baseenergy	Displays the base energy of the fixtures. Base energy is when the fixture's light is always maintained at full-on without making any changes to the light.
energy	The aggregate energy consumed by the fixtures on the floor
saveEnergy	The aggregate energy saved by the fixtures on the floor
occSavings	Energy savings for the fixtures on the floor determined by occupancy nearby the fixture
ambientSavings	Ambient savings for the fixtures on the floor based on ambient light level measured by the fixture
manualSavings	Manual savings for the fixtures on the floor based on when the fixtures were manually controlled

Sample Code

Using the Floor ID '1' returned by the *Get All Floors* API, request the aggregate energy consumption for ten fixtures on the floor for two hours from 5:30am through 7:30am on Feb. 12, 2019. The sample response shows details returned for ten fixtures on the floor for the first one-hour of the specified time interval.

```
GET
https://192.80.3.2/ems/api/org/facility/v2/facilityEnergyStats/15min/floor/1/2019031204
3500/20190314043500
{
  "facilityEnergyStatss": {
    "facilityEnergyStats": {
      "id": "1",
      "intervalStart":"2019-03-14T04:00:00+05:30",
```

```
      "name":         "7th   Floor",
      "baseEnergy":   "38.85",
      "energy":       "16.00",
      "savedEnergy":  "22.85",
      "occSavings":   "16.20",
      "ambientSavings":"0.00",
      "tuneupSavings": "6.65",
        "manualSavings":"0.00"
 }
 {
      "id": "1",
      "intervalStart":"2019-03-14T04:00:00+05:45",
      "name":         "7th   Floor",
      "baseEnergy":   "48.85",
      "energy":       "15.00",
      "savedEnergy":  "25.85",
      "occSavings":   "16.20",
      "ambientSavings":"0.00",
      "tuneupSavings": "6.65",
        "manualSavings":"0.00"
 }
 {     "id": "1",
      "intervalStart":"2019-03-14T04:00:00+06:00",
      "name":         "7th   Floor",
      "baseEnergy":   "38.85",
      "energy":       "16.00",
      "savedEnergy":  "22.85",
      "occSavings":   "16.20",
      "ambientSavings":"0.00",
      "tuneupSavings": "6.65",
        "manualSavings":"0.00"
}
{
      "id": "1",
      "intervalStart":"2019-03-14T04:00:00+06:15",
      "name":         "7th   Floor",
      "baseEnergy":   "38.85",
      "energy":       "16.00",
      "savedEnergy":  "22.85",
      "occSavings":   "16.20",
      "ambientSavings":"0.00",
      "tuneupSavings": "6.65",
        "manualSavings":"0.00"
}
}
```

Chapter 10: Demand Response APIs

These services provide the ability to quickly override schedules in advance during demand response (DR) periods.

Schedule Demand Response (DR) for all Facilities

Schedules a Demand Response (DR) for all facilities for a specific period from the start time.

Request

POST https://{em_ip_address_or_hostname}/ems/api/org/dr/scheduledr

```
{
    "pricelevel": "LOW",
    "duration": "60",
    "starttime": "2015-07-21T12:15:49+05:30",
    "dridentifier": "DR16",
    "pricing": "15.0"
 }
```

Parameters

Request	
em_ip_address or hostname	The Energy Manager's IP address or host name
Body	
pricelevel	DR pricing: HIGH MODERATE LOW SPECIAL
duration	The DR duration time in minutes
starttime	Time from when DR should be in effect. Format: (yyyyMMddHHmm) year/month/day/hour/minutes
dridentifier	User assigned Demand Response (DR) identifier
pricing	DR pricing that is effective for the energy consumed during the DR period (optional)
Response	
status	0 – DR schedule successful 1 – DR not scheduled
msg	Message description

Sample Code

Schedule a Demand Response (DR) at LOW price level for all facilities for 60 minutes starting at 5:30am on 07-21-2015 at 15 cents. The response indicates that the DR schedule was successful.

```
If the user is accessing web service with proper session/authenticated token:
POST https://192.80.3.2/ems/api/org/dr/scheduledr
{
    "pricelevel": "LOW",
    "duration": "60",
    "starttime": "2015-07-21T12:15:49+05:30",
    "dridentifier": "DR16",
    "pricing": "15.0"
 }
```

```
Response:
HTTP/1.1 200: OK
  {
    "status": "0",
    "msg": "DR Scheduled successful"
  }

If the user is accessing web service without creating session:
HTTP/1.1 302: Moved Temporarily
```

Schedule Demand Response (DR) for Selected Facilities

Schedules a Demand Response (DR) for a list of selected facilities in the organization for a specific period from the start time.

Request

POST https://{em_ip_address_or_hostname}/ems/api/org/dr/scheduledr/facilities/{facility specification list}

```
{
    "pricelevel": "LOW",
    "duration": "60",
    "starttime": "2015-07-21T12:15:49+05:30",
    "dridentifier": "DR16",
    "pricing": "15.0"
 }
```

Parameters

Request	
em_ip_address or hostname	The Energy Manager's IP address or hostname
facility specification list	Schedules a demand response for a list of selected facilities. List of facilities separated by commas. Each facility contains facility type and facility ID: {facility type}_{facility ID} Facility Type: Can be either a company, campus, building, or floor. Facility ID: A unique identifier that is assigned to the company or organization, campus, building or floor. (Facility ID is returned in *Get All Floors* API). **Note**: Specifying a facility other than the floor automatically assigns DR to all floors under the facility. Assigning a campus will assign DR to all floors for all buildings on the campus. For example, 1. To assign DR to the first and second floors of a building, type floor_1,floor_2. DR will be assigned to floor 1 and floor 2. 2. To assign DR to all floors in the Headquarters Building with ID 22, type Building_22. DR will be assigned to all floors in building 22.
Body	
pricelevel	DR pricing: HIGH MODERATE LOW SPECIAL
duration	The DR duration time in minutes
starttime	Time from when DR should be in effect. Format: (yyyyMMddHHmm) year/month/day/hour/minutes

dridentifier	User assigned DR identifier
pricing	DR pricing that is effective for the energy consumed during the DR period (optional)
Response	
status	0 – DR schedule successful 1 – DR not scheduled
msg	Message description

Sample Code

For example, schedule a Demand Response (DR) at LOW price level for Floor 1 in Building 51, Campus 1, for 60 minutes starting at 5:30am on 07-21-2015 at 15 cents. The response indicates that the DR schedule was successful.

```
If the user is accessing web service with proper session/authenticated token:
POST
https://192.80.30.2/ems/api/org/dr/scheduledr/facilities/floor_1,building_51,campus
_1
{
    "pricelevel": "LOW",
    "duration": "60",
    "starttime": "2015-07-21T12:15:49+05:30",
    "dridentifier": "DR16",
    "pricing": "15.0"
 }


HTTP/1.1 200: OK
  {
    "status": "0",
    "msg": "DR Scheduled successful"
  }
```

Update DR for all Facilities

Update the current running or scheduled DR for all facilities.

Request

POST https://{em_ip_address_or_hostname}/ems/api/org/dr/updatedr

```
{
    "pricelevel": "LOW",
    "duration": "60",
    "starttime": "2015-07-21T12:15:49+05:30",
    "dridentifier": "DR16",
    "pricing": "15.0"
 }
```

Parameters

Request	
em_ip_address or hostname	The Energy Manager's IP Address or host name
Body	
pricelevel	DR pricing: HIGH MODERATE

	LOW SPECIAL
duration	The DR duration time in minutes
starttime	Time from when DR should be in effect. Format: (yyyyMMddHHmm) year/month/day/hour/minutes, for example, 2015-11-18T17:43:00
dridentifier	DR identifier provided at the time of scheduling Maximum value: 60 minutes
pricing	DR pricing that is effective for the energy consumed during the DR period (optional)
Response	
status	0 – DR schedule successful 1 – DR not scheduled
msg	Message description

Sample Code

Update the currently running Demand Response (DR) from LOW to HIGH price level for all facilities for 60 minutes starting at 5:30am on 07-21-2015 at 15 cents. The response indicates that the DR update was successful.

```
If the user is accessing web service with proper session/authenticated token:
POST https://{192.80.30.2}/ems/api/org/dr/updatedr
{
    "pricelevel": "HIGH",
    "duration": "60",
    "starttime": "2015-07-21T12:15:49+05:30",
    "dridentifier": "DR16",
    "pricing": "15.0"
 }

HTTP/1.1 200: OK
{
    "status": "0",
    "msg": "DR updated successfully"
}
```

Update DR for Selected Facilities

Update the current running or scheduled DR for a list of selected facilities in the organization for a specific period from the start time.

Request

POST https://{em_ip_address_or_hostname}/ems/api/org/dr/updatedr/facilities/{facility specification list}

```
{
    "pricelevel": "LOW",
    "duration": "60",
    "starttime": "2015-07-21T12:15:49+05:30",
    "dridentifier": "DR16",
    "pricing": "15.0"
 }
```

Parameters

Request	
em_ip_address or hostname	The Energy Manager's IP address or hostname
facility specification list	Schedules a demand response for selected facilities. List of facilities separated by commas. Each facility contains facility type and facility ID: {facility type} {facility ID} Facility Type: Can be either a company, campus, building, or floor. Facility ID: A unique identifier that is assigned to the company or organization, campus, building or floor. (Facility ID is returned in *Get All Floors* API). **Note**: Specifying a facility other than the floor automatically assigns DR to all floors under the facility. Assigning a campus will assign DR to all floors for all buildings on the campus. For example, 1. To assign DR to the first and second floors of a building, type floor_1,floor_2. DR will be assigned to floor 1 and floor 2. 2. To assign DR to all floors in the Headquarters Building with ID 22, type Building_22. DR will be assigned to all floors in building 22.
Body	
pricelevel	DR pricing: HIGH MODERATE LOW SPECIAL
duration	The DR duration time in minutes
starttime	Time from when DR should be in effect. Format: (yyyyMMddHHmm) year/month/day/hour/minutes, for example, 2015-11-18T17:43:00
dridentifier	DR identifier provided at the time of scheduling
pricing	DR pricing that is effective for the energy consumed during the DR period (optional)
Response	
status	0 – DR schedule successful 1 – DR not scheduled
msg	Message description

Sample Code

Update the currently running Demand Response (DR) event from LOW to HIGH price level for Floor 1 in Building 51, Campus 1, for 60 minutes starting at 5:30am on 07-21-2015 at 15 cents. The response indicates that the DR update was successful.

```
If user is accessing web service with proper session/authenticated token:
POST
https://192.80.30.2/ems/api/org/dr/updatedr/facilities/floor_1,building_51,campus
_1

{
    "pricelevel": "HIGH",
    "duration": "60",
    "starttime": "2015-07-21T12:15:49+05:30",
    "dridentifier": "DR16",
    "pricing": "15.0"
 }

HTTP/1.1 200: OK
{
    "status": "0",
    "msg": "DR updated successfully"
}
```

List DR

Obtains a list of all running and scheduled DRs.

Request

GET https://{em_ip_address_or_hostname}/ems/api/org/dr/listdr

Parameters

Request	
em_ip_address or hostname	The Energy Manager's IP address or hostname
Response	
Displays a list of current or scheduled target Demand Responses for the organization	

Sample Code

Display list of all running and scheduled DRs.

```
If the user is accessing web service with proper session/authenticated token:
GET https://93.2.8.1/ems/api/org/dr/listdr

HTTP/1.1 200: OK
{
  "dRTargets": {
    "drTarget": {
      "id": "39",
      "pricelevel": "LOW",
      "pricing": "15.0",
      "duration": "3600",
      "targetreduction": "10",
      "starttime": "2015-07-21T13:15:49+05:30",
      "dridentifier": "DR18",
      "drstatus": "Active",
      "drtype": "MANUAL",
      "option": "true",
      "jitter": "0"
    }
  }
}
```

Cancel DR

Cancels the currently running DR or a scheduled DR.

Request

POST https://{em_ip_address_or_hostname}/ems/api/org/dr/canceldr

```
{
    "dridentifier": "DR16",
}
```

Parameters

Request	
em_ip_address or hostname	The Energy Manager's IP address or hostname
Body	
dridentifier	DR identifier provided at the time of scheduling
Response	
status	0 – DR canceled successfully 1 – DR not canceled
msg	Status message description

Sample Code

Cancel the scheduled Demand Response (DR) event using the DR identifier. The response indicates that the DR canceled successfully.

```
If the user is accessing web service with proper session/authenticated token:
POST https://192.80.30.2/ems/api/org/dr/canceldr
{
    "dridentifier": "DR16",
 }
HTTP/1.1 200: OK
{
    "status": "0",
    "msg": "DR Cancelled successfully"
}
```

Chapter 11: Tunable White APIs

The APIs set the color temperature and adjust the light level for fixtures that support Tunable White.

Change Color

Changes the color temperature of the fixtures to a specified setting for the specified duration.

Request

POST https: //{em_ip_address_or_hostname}/ems/api/org/sensor/tunable-white/change-color/value/target colorvalue/time

```
<fixtures>
  <fixture>
   <id>63</id>
    <name>Sensor71a4bc</name>
  </fixture>
  <fixture>
    <id>145</id>
    <name>Sensor71a45a</name>
  </fixture>
</fixtures>
```

Parameters

Request	
em_ip_address or hostname	The Energy Manager's IP address or hostname
targetcolorvalue	Color Temperature value indicating the requested color temperature in Kelvin degrees. Valid range depends on the Fixture Manufacturer Table (FMT) supplied by the manufacturer for the fixture.
time	Duration in minutes
Response	
Status	If the request is successfully handled "Change color success" message is displayed. When sensor details are missing or inaccurate in the Post command, "Sensors list is empty while requesting API" message is displayed.

Sample Code

Change the color temperature value to 3200 for 60 minutes for fixtures with ID '63' and '145'. The response 'Change color success' indicates that the color change is applied to the Tunable White fixtures successfully. Note that the response is a group response for all fixtures. The response is reported as a success if the message is successfully sent to the sensors.

```
POST https://10.8.1.90/ems/api/org/sensor/tunable-white/change-
color/value/3200/60

<fixtures>
  <fixture>
   <id>63</id>
    <name>Sensor71a4bc</name>
  </fixture>
  <fixture>
```

```
    <id>145</id>
    <name>Sensor71a45a</name>
  </fixture>
</fixtures>


Response Success:

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
   <response>
              <status>0</status>
              <msg>change color success</msg>
   </response>
```

Scene Control

When a switch group is created in the Energy Manager, a virtual switch is added to the floor which controls the light and color intensity of the selected fixtures. Each set of light and color intensity settings for fixtures is called a scene.

Using this API, set the light and color intensity of the scene for the fixtures in the switch group to achieve the desired effect. Use the Scene ID and Switch ID returned by the *Get Scene Light Levels for a Switch* API call to set the light levels for each scene.

Request

POST
https://{em_ip_address_or_hostname}/ems/services/org/switch/op/dim/switch/switchid/scene/sceneid/percentage/time

Parameters

Request	
em_ip_address or hostname	The Energy Manager's IP address or hostname
switchid	Switch Identifier. The Switch ID returned in the *Get Scene Light Levels for a Switch* API
sceneid	Scene Identifier. The Scene ID returned in the *Get Scene Light Levels for a Switch* API
percentage	Percentage of light level 0% - Light levels set to off 100% - Light levels set to full on. 101 - Auto: Sets the light level based on the lighting profile behavior 102 – Sets the scene light level configured for the fixtures when a switch is created on the floor plan in the Energy Manager.
time	Duration in minutes
Response	
Status	0 – Success 1 – Failure

Sample Code

Using the Scene ID and Switch ID returned by the *Get Scene Light Levels for a Switch* API, set the color intensity for the Scene with ID '157' belonging to the switch with ID '1' to 100% Full on for 10 mins. The response '0' indicates that the scene is applied to the switch successfully.

```
POST https://10.8.1.90/ems/services/org/switch/op/dim/switch/1/scene/157/100/10

{
  "status": "0"
}
```

Chapter 12: Bluetooth Services

The APIs control the Bluetooth sensors on a floor.

Get all BLE Fixtures by Floor

The API returns the sensor location for all BLE sensors on the floor. The data is returned as X, Y coordinates of the measured and scaled down distance from the lower-left corner of the uploaded floor plan. The API also returns the sensor model number.

The Product codes and their descriptions are described below.

Product Code	Description
SU-4S-H	High Bay Smart Sensor with Bluetooth
SU-4S-LRW	Ruggedized Sensor with Bluetooth - White
SU-4S-LRB	Ruggedized Sensor with Bluetooth - Bronze
SU-4S-HRW	High Bay Ruggedized Sensor with Bluetooth - White
SU-4S-HRB	High Bay Ruggedized Sensor with Bluetooth – Bronze
SU-4E-01	Compact Sensor with Bluetooth
FS-D22	Two-wire Fixture Mount Sensor with Bluetooth
CS-D2	Two-wire Compact Sensor with Bluetooth

Request

GET https://{em_ip_address_or_hostname}/ems/api/org/fixture/blefixture/list/floor/{floor_id}

Parameters

Request	
em_ip_address or hostname	The Energy Manager's IP address or host name
floor_id	Floor identifier. Use the Floor ID returned in the *Get All Floors* API, or in EM select *Facility > Floor >* in the left panel, and click the *Settings* tab.
Response	
id	Sensor identifier
name	Sensor name. For example, Sensorxxxxxx.
x-axis	Scaled reference of the distance measured in the horizontal dimension from the lower-left corner of the uploaded floor plan
y-axis	Scaled reference of the distance measured in the vertical dimension from the lower-left corner of the uploaded floor plan
groupid	Profile group identifier. The ID of the profile associated with the fixture.
macAddress	MAC address
modelNo	Sensor product code. See product codes table above.

Sample Code

Using the Floor ID '1' returned by the *Get All Floors* API, request EM to return the x- and y- axis of the scaled down distance from the lower-left corner of the floor plan for all BLE sensors on Floor 1.

```
GET
https://{em_ip_address_or_hostname}/ems/api/org/fixture/blefixture/list/floor/2

HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: application/xml
Content-Length: 7582
Date: Sat, 25 Feb 2012 17:06:51 GMT

{
  "fixture": [
    {
      "id":   "361",
      "name":   "Sensor00c017",
      "xaxis":   "966",
      "yaxis":   "391",
      "groupid":   "185",
      "modelNo": "SU-4E-01",
      "macaddress":   "0:c0:17"
      ,
      …
    },
    {
      "id":   "1364",
      "name":   "Sensor0629dd",
      "xaxis":   "270",
      "yaxis":   "90",
      "groupid":   "196",
      "modelNo": "SU-4E-01"
      "macaddress":   "6:29:dd"
      ..
    },
    {
      "id":   "835",
      "name":   "Sensor013ec0",
      "xaxis":   "756",
      "yaxis":   "462",
      "groupid":   "55",
      "modelNo": "SU-4E-01",
      "macaddress":   "1:3e:c0"
      ..
    }
  ]
}
```

Set BLE Mode of Sensor

The API sets the BLE mode of the sensor.

Request

POST https://{em_ip_address_or_hostname}/ems/api/org/fixture/op/assignblemode/{blemode}

```
<fixtures>
  <fixture>
    <id>27</id>
  </fixture>
</fixtures>
```

Parameters

Request	
em_ip_address or hostname	The Energy Manager's IP address or host name
blemode	Sensor BLE mode: OFF – BLE mode is off SCAN – The sensor scans for BLE signals BEACON – The sensor sends out BLE signals
id	A unique identifier of the sensor
Response	
status	nnn – Number of sensors where the BLE mode was updated successfully -1 – Error message
msg	Error message description

Sample Code

Set the BLE Mode of the sensor for two fixtures to OFF. The response indicates that two fixtures have been updated successfully. If there is an error, the error message is displayed.

```
POST https://192.80.30.2/ems/api/org/fixture/op/assignblemode/OFF
<fixtures>
  <fixture>
    <id>27</id>
  </fixture>
<fixture>
    <id>28</id>
  </fixture>
</fixtures>

HTTP/1.1 200: OK
  {
    "status": "2"
  }
```

or

```
HTTP/1.1 200: OK
  {
    "status": "-1",
    "msg": "Error..."
  }
```

API Example

Consider you have an AV touchscreen, and you would like to override fixture light settings for the switch on a floor. To use the override API, you will need to know the Switch name and Floor ID of the floor in the building. Note that all API calls are secure HTTPS calls and should be made with associated username and specific APIkey. Here are the steps that you will need to perform.

1. Authenticate with username and password.
2. Determine the Floor ID. Use the *Get All Floors* API to obtain the list of floors in the building for the organization.
3. Next, determine the switch name that was programmed in the Energy Manager (EM). In EM, select *Facility > Floor* in the left panel, click *Floor Plan* tab to view the floor plan. Hover the cursor on the switch to obtain the switch name. You can also use the *Get Switch Groups* API to obtain the Switch name, but note that this command will list all switch names on the floor.
4. Using the Floor ID returned by the *Get All Floors* API and Switch, override the fixture profile settings controlled by the switch to dim the fixtures for a period of time using the *Manual Override for Switch Group* API.

Sample Code

```
**Obtain floor-id**

Get https://{192.80.3.2}/ems/api/org/floor/list

HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: application/xml
Content-Length: 7582
Date: Sat, 21 Jul 2015 17:06:51 GMT

{
  "floor": [
    {
      "id": "1",
      "name": "First Floor",
      "building": "1",
      "campus": "1",
      "company": "1",
      "description": "Sales and Marketing",
      "floorPlanUrl": "entire 930 floor plan.jpg"
    },
    {
      "id": "2",
      "name": "Second Floor",
      "building": "1",
      "campus": "1",
      "company": "1",
      "description": "Product",
      "floorPlanUrl": "entire 930 floor plan.jpg"
    }

  ]
}
```

```
** Manually override the switch settings using the Floor ID and Switch name**

Post
https://192.80.3.2/ems/api/org/switch/v1/op/dimGroup/1/SwitchGrp1/10?time=60

HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: application/json
Content-Length: 7582
Date: Sat, 21 Jul 2015 17:06:51 GMT
{
  "status": "0"
}

**Overrides the Switch settings and lowers the fixture light level to 10% for
one hour.**
```

Revision Changes

Version	New Updates/Features	Release
Rev. 15	• Added Tunable white APIs • Added Get Area Occupancy for a Floor	May 2019
Rev. 14	• Described Switch Group and virtual Switch ID and updated conference APIs. • Added Tunable white APIs	April 2019
Rev. 13	• Added Apply Scene API, updated Get Aggregate Sensor Energy Data	April 2019
Rev. 12	• Added company, campuses, buildings APIs in Chap.4 • Included API Example	March 2019
Rev. 11	• Added Schedule DR and Update DR API by Facility	Sept. 2018
Rev. 10	• Updated Authorization timestamp description • Updated Light level parameter description	April 2018
Rev. 09	• Removed 'ss' seconds from_date and to_date parameters • Updated Authorization value • Added units for timestamp • Updated BLE modes in Set BLE Mode of Sensor API	April 2018
Rev. 08	Updated description for x-axis and y-axis	March 2018
Rev. 07	Updated description for Get Switch Scenes	August 2017
Rev. 06	Internal revisions, formatting updates	May 2017
Rev. 05	Internal revisions, formatting updates	May 2017
Rev. 04	Internal revisions, formatting updates	May 2017
Rev. 03	Updated API authentication order Added Bluetooth APIs	June 2016
Rev. 02	Updated Generate API key Explained how to calculate timestamp and SHA1-key with an example	March 2016
Rev. 01	First Release	Dec. 2015

Copyright © 2018-19 Enlighted, Inc., 930 Benecia Ave., Sunnyvale, CA 94085 U.S.A. All rights reserved. All other brand or product names are trademarks of their respective companies or organizations.