Enable Gateway REST API

User Guide

Contents

1. PURPOSE OF THIS DOCUMENT 3

2. GATEWAY API OVERVIEW 3

3. PROTOCOLS 3

A. REST APIs 3
B. MQTT 4

4. NETWORK REQUIREMENTS 4

5. COMMISSIONING PROCESS 4

6. SETUP CERTIFICATES TO SECURE REST API COMMUNICATION 6

7. COMMISSIONING ENGINEER: REST API VALIDATION 6

A. CREATE PRIVATE KEY AND SIGNING REQUEST FILES 6
B. UPLOAD CSR FILE INTO SIGNIFY INTERACT BUILDER PORTAL 7
C. TEST THE INTEGRATION USING POSTMAN APPLICATION 8

8. INTEGRATOR PREPARATION AND VALIDATION 11

A. INTEGRATOR - .CSR AND .KEY GENERATION 11
B. INTEGRATOR CERTIFICATE GENERATION 11

1. Purpose of this document

This document provides instructions to enable and test the Signify Interact Gateway APIs, including REST and MQTT integrations. All technical details, OpenSSL commands, commissioning steps, network requirements, and Postman testing procedures are included in this document.

2. Gateway API overview

The Gateway API enables on-premise integration with Signify Interact lighting systems. It provides local control and data reporting capabilities for integration without requiring continuous cloud connectivity.

Note: Cloud connectivity is required only for commissioning, project updates, and firmware updates.

During integration process, the following protocols are supported.

- **REST APIs** (Representational State Transfer Application Programming Interface) – Used for group control operations
- **MQTT** (Message Queuing Telemetry Transport) – Used for data reporting from sensors and drivers

The gateway sends data to the MQTT server of customer or integrator. The integrator must use a REST API call to configure the gateway with the MQTT server. Validation of the MQTT broker integration must be done by the integrator. For more information, refer to the Developer Portal.

3. Protocols

The supported protocols are REST APIs (Representational State Transfer Application Programming Interface) and MQTT (Message Queuing Telemetry Transport).

a. REST APIs

Refer to the Developer Portal to know more about the REST API.

The following functionality can be used with REST APIs.

- Group on/off control (method: 'onoff')
- Group dim level control (method: 'level')
- Group scene control (method: 'scene')
- MQTT broker configuration of the gateway (method: "bmsconfig")

b. MQTT

Refer to the Developer Portal to know more about the MQTT.

MQTT provides:

- Occupancy events from device and group level occupancy
- Faults such as driver failures and lamps failures, and burning hours like current burning hours reported by each driver to the sensor over DALI
- Energy consumption reported by each driver to the sensor over DALI

4. Network requirements

For network requirements, refer to the Network requirements section in Signify Interact Builder documentation.

Additional network requirements are:

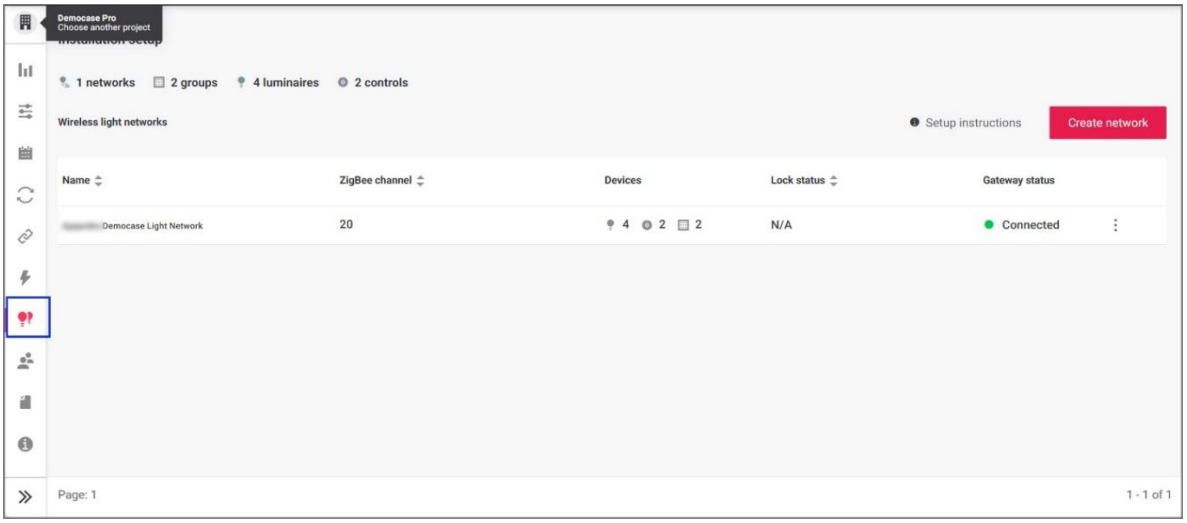
- A dedicated VLAN for all the lighting wireless gateways
- If the integration happens within the same VLAN as the gateway, the VLAN must support mDNS.
- If inter VLAN communication is needed to enable the BMS or the third-party integrator to connect with the lighting system, DNS must resolve the gateway name.

5. Commissioning process

Prerequisite: A project is commissioned. Lights, groups, and gateways are added and deployed. Ensure that the gateway shows three solid blue LEDs. For more information, refer to the Commissioning guide.

Perform the following procedure to enable the building integrations license.

1. In the Signify Interact Builder portal (https://sme.interact-lighting.com/), click **Setup** in the left navigation pane.



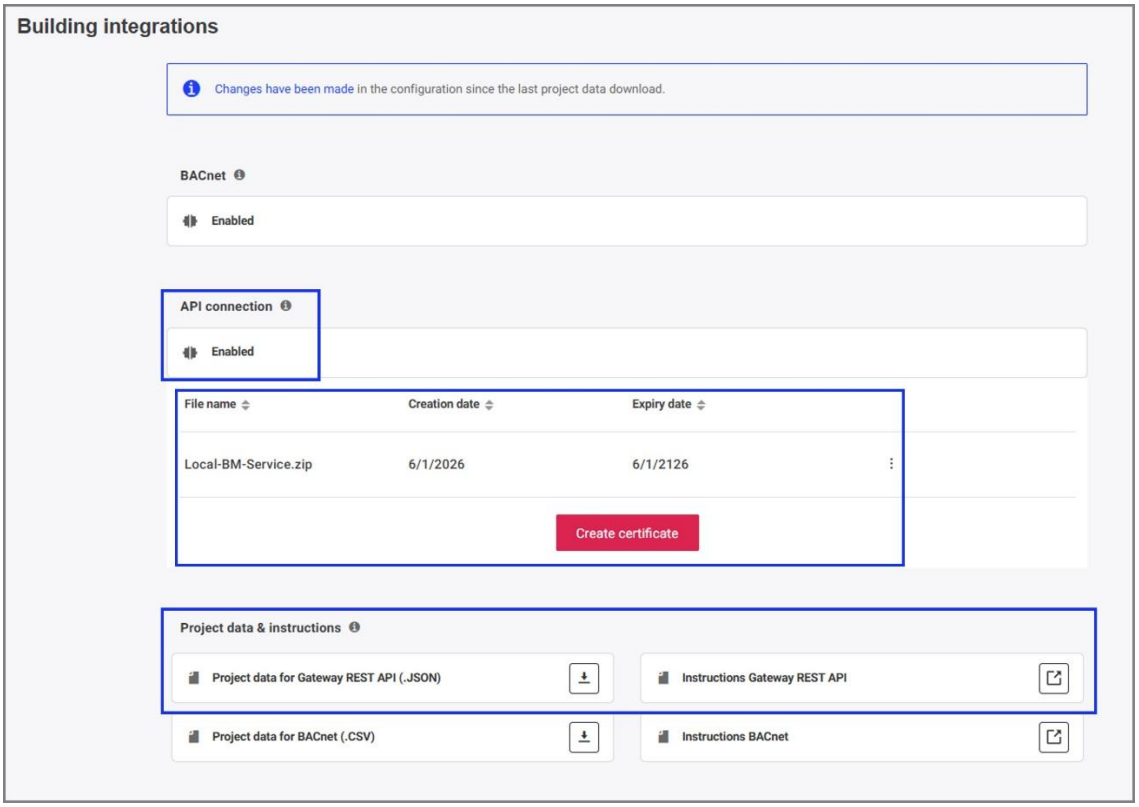
2. In the **Installation setup** page, make sure all networks are unlocked and **Sync changes** are done for all networks.
   Note: This process needs to be repeated every time there is a change done to the project like adding or deleting areas to make sure that the new or deleted areas are accessible/updated to the APIs.
3. Request a Gateway API license through the OTRS ticket or CSI manager. In the OTRS portal, navigate to the **Intelligent Buildings** → **Software Service** → **Gateway API** to raise the license request.



4. Provide the Project ID information which is available in the project URL.



Result: Once the APIs have been activated, the status of **API connection** is updated as **Enabled**, and you can proceed with creating certificates and downloading the appropriate data and instructions.



6. Setup certificates to secure REST API communication

Every client that connects to the gateway must create his own personal set of:

- Certificate Signing Request (.csr)
- Private key (.key) - This file is private and should not be shared with anyone.
- Certificates (.crt)

In practice, at least two sets of certificates must be created:

- The commissioning engineer creates his personal set to validate the correct functioning of the gateway.
- The integrator creates his personal set for his application.

7. Commissioning engineer: REST API validation

a. Create private key and signing request files

The commissioning engineer performs the following procedure to generate the OpenSSL certificate.

1. Install OpenSSL on your computer.
   - For Windows: In this URL - Win32/Win64 OpenSSL Installer for Windows - Shining Light Productions), navigate to the **Downloads** table and click the latest version of Win64 OpenSSL vX.X.X Light, EXE or MSI.
   - For MacOS: first install brew (https://brew.sh/) and then install OpenSSL using brew (https://formulae.brew.sh/formula/openssl@3)
2. Navigate to the OpenSSL bin folder *C:\Program Files\OpenSSL-Win64\bin*
3. In MacOS, right-click on the folder and select **New terminal at folder** option.
   In Windows, open Command Prompt (CMD) and navigate to that folder using

   ```
   cd C:\Program Files\OpenSSL-Win64\bin
   ```

4. Run the following command.

   ```
   openssl req -new -newkey ec -pkeyopt ec_paramgen_curve:prime256v1 -
   keyout Local-BM-Service.key -out Local-BM-Service.csr -subj
   "/C=NL/O=Signify/OU=Interact/CN=LCN1870-XXXXXX.local" -addext
   "subjectAltName=DNS:LCN1870-XXXXXX.local"
   ```

   Tip: Make sure to execute this as a single command.

   Where:

   - **XXXXXX** are the last 6 digits of the gateway MAC
   - **LCN1870** is the gateway type. Adjust this for the gateway type of your project.

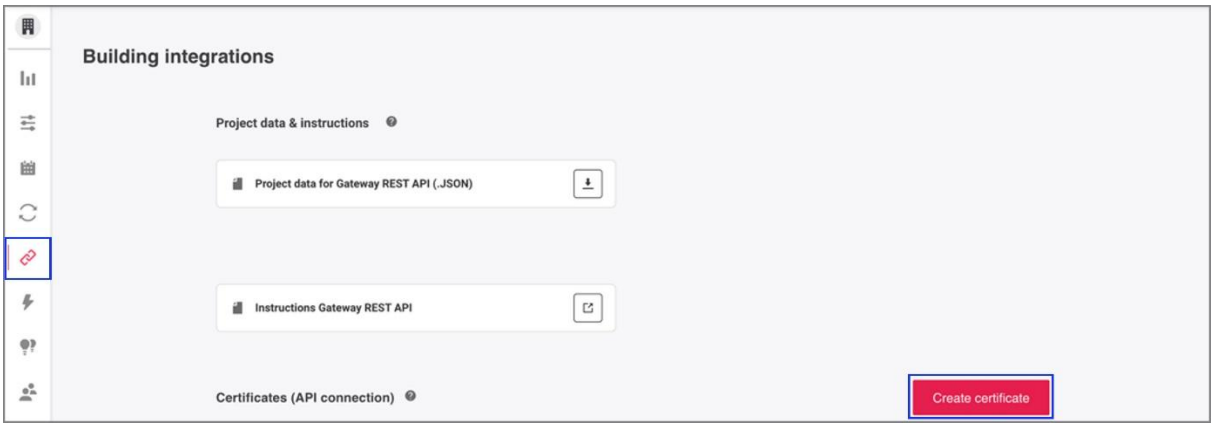
For example, the Host URL for the gateway with MAC address 001788FFFE0E1DEF is LCN1870-0E1DEF.local

This will ask for a password for the .key file. Store the password in a safe place.
This creates:

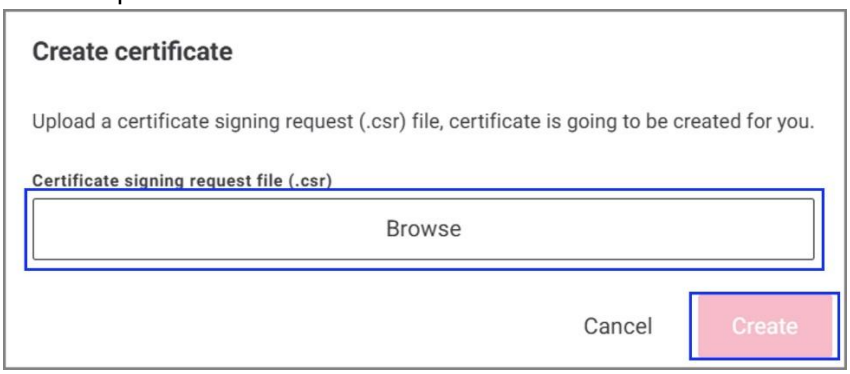
- Local-BM-Service.csr: The signing request file
- Local-BM-Service.key: This file is private to the commissioning engineer and shall not be shared with anyone.

b. Upload CSR file into Signify Interact Builder portal

1. In the Signify Interact Builder portal (https://sme.interact-lighting.com/), select the required project.
2. Click **Building integrations** in the left navigation pane.

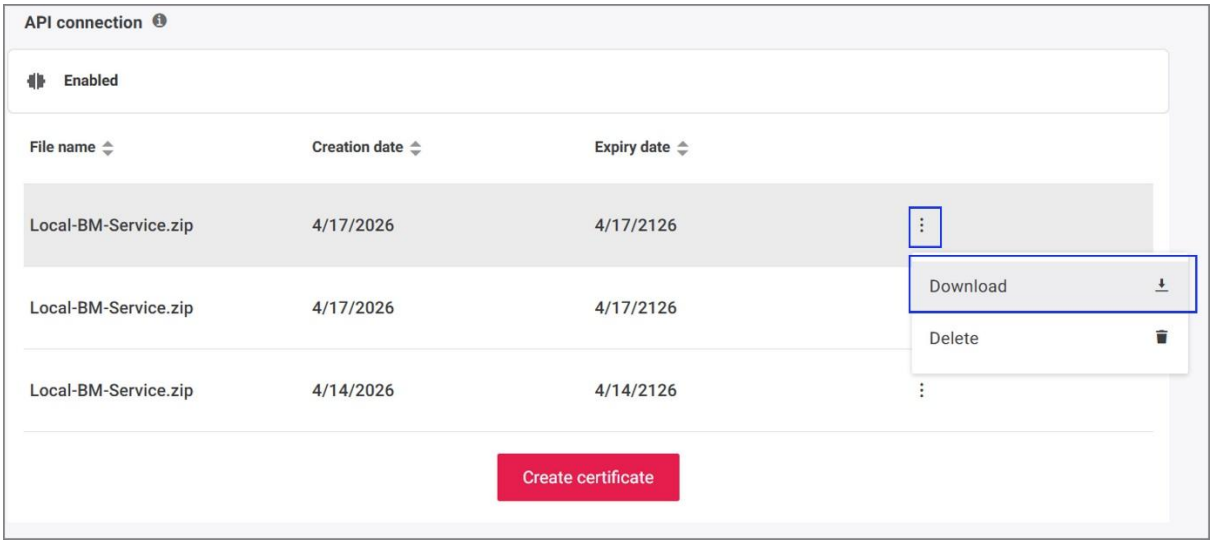


3. In the **Building integrations** page, click **Create certificate**.
4. In the **Create certificate** dialog-box, click **Browse** and select the **Local-BM-Service.csr** file to upload.



5. Click **Create** and a new .zip file will be created in the **Building integrations** page.

6. Click the three-dot Actions menu and click **Download** to download the generated .zip file.



Note: The .zip file contains **Local-BM-Service.crt** file and **SiteCA#####.crt** file where ##### refers to your project ID.

7. In the **Building integrations** page, download the **Project data for Gateway REST API (.JSON)** file.

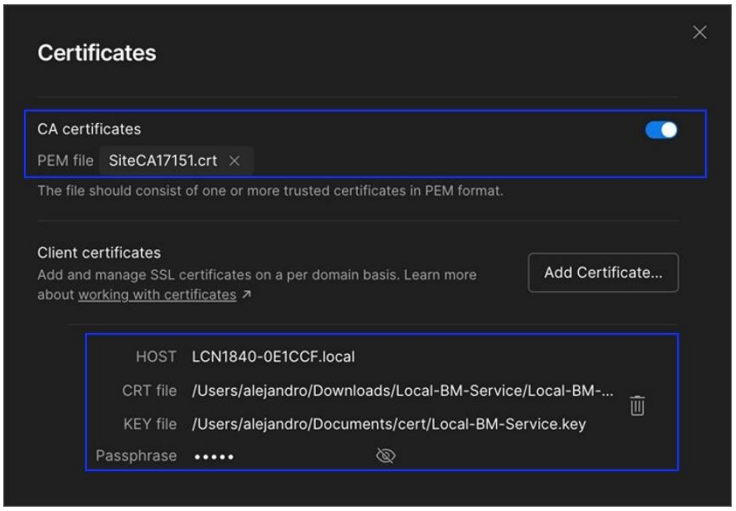


c. Test the integration using Postman application

Perform the following procedure to test the local integration using Postman application.

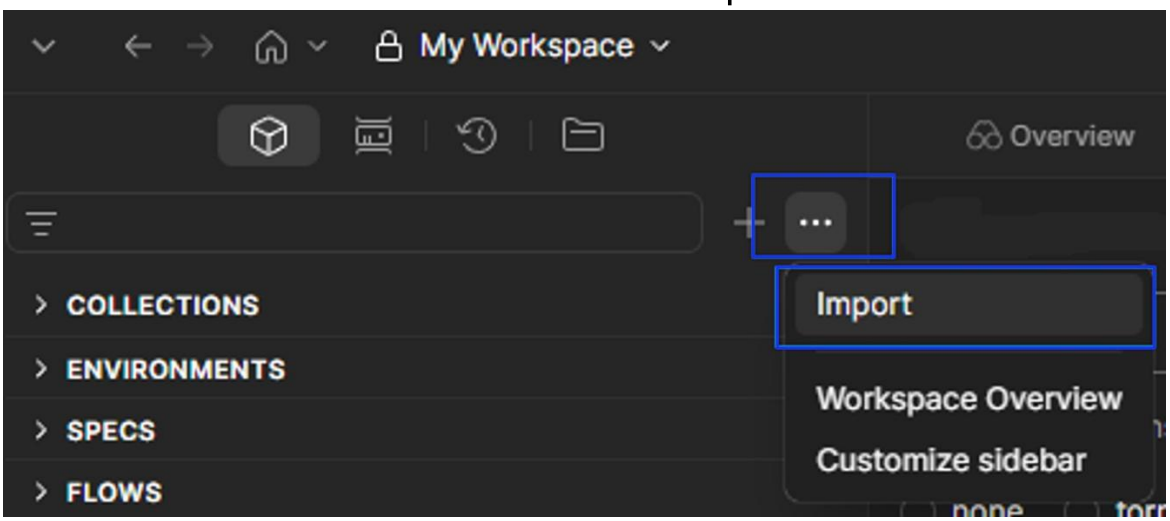
1. Download the **Postman** application (https://www.postman.com/downloads/) and install it on your computer.
2. Open the **Postman** application and select **Settings** in the header.
3. Select the **Certificates** tab.

4. In the **Certificates** dialog-box, perform the following actions and close the dialog-box.

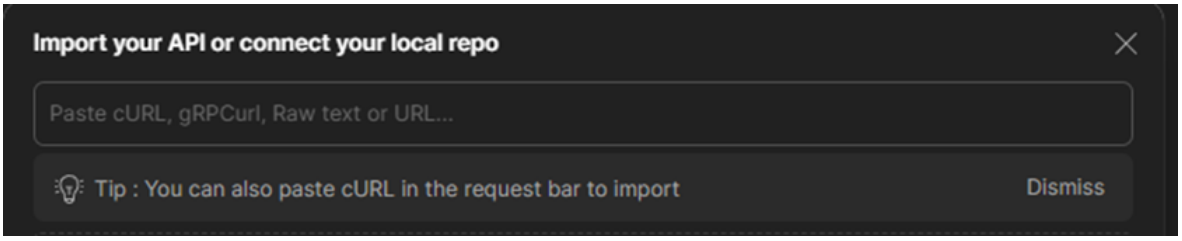


- Enable the **CA certificates** toggle button and upload the **SiteCAxxxx.crt** file from your computer.
- For **HOST**, add the gateway URL to be tested. Example: **LCN1870-xxxxxx.local** where xxxxxx stands for the last six digits of the gateway's MAC address.
- For **CRT file**, upload the **Local-BM-Service.crt** file from your computer.
- For KEY file, upload the **Local-BM-Service.key** file from your computer.
- For **Passphrase**, add the password you used while creating the signing request and key file.

5. Click the three-dots actions menu and select **Import**.



6. Paste the following cURL in this window.



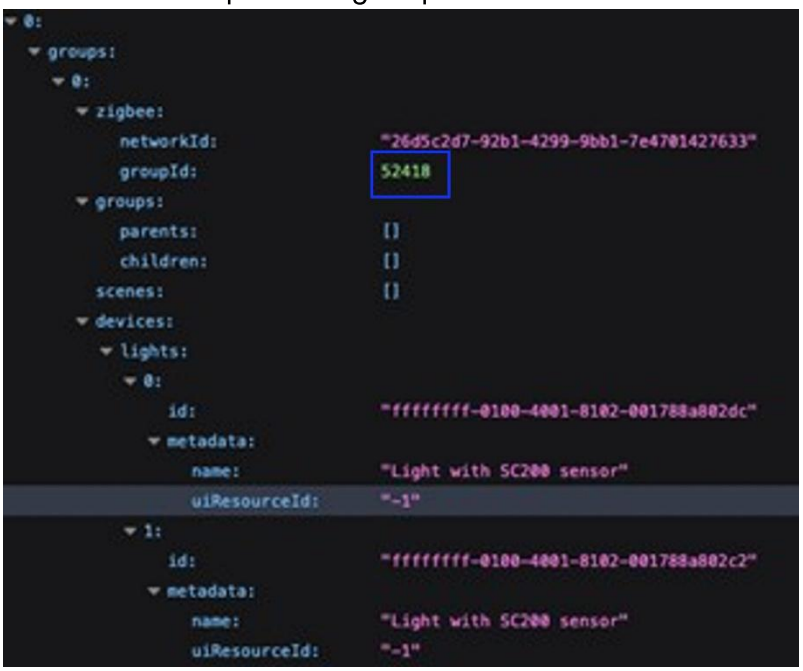
```
curl --url 'https://LCN1870-0E22E1.local/api/v1/zigbee/groups/YYYY/methods' \
  --header 'Content-Type: application/json' \
  --data '{
    "methodName": "onoff",
    "payload": {
      "on": true
    }
  }'
```

Replace LCN1870-0E22E1 with your gateway name and YYYY stands for the group ID which will be available in the JSON file. An example is shown below to get the group ID.

Select or define a **Collection name** where the API request should be stored.



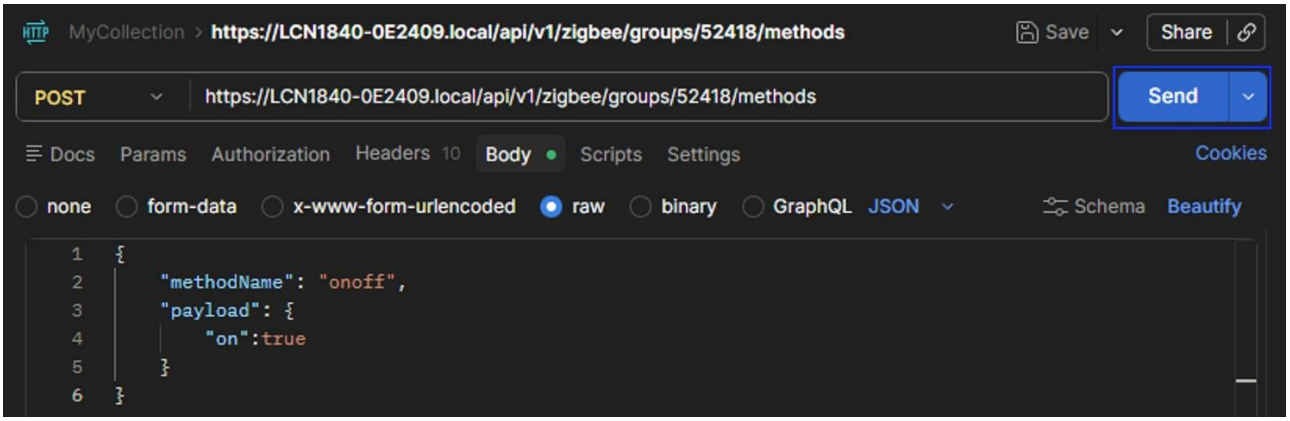
JSON file example with group ID:



Tip: To find a groupId for your project, upload the project .json file in a company approved AI chat application and prompt it with "Provide a brief summary of the groups and zones, including their corresponding Zigbee group IDs, as defined in the attached project file".

7. This results in an API request and click **Send** to execute it.

   Note: To execute this API call, your laptop must be in the same VLAN as the gateway.



   Result: The lights of the selected group should be turned on.

8. In the command, replace **`"on": true`** with **`"on": false`** to turn off the lights for the selected group.
9. Test other gateways similarly with the respective gateway URL and group ID.

8.Integrator preparation and validation

a. Integrator - .csr and .key generation

Similar to the commissioning engineer, the integrator must create his own personal set of:

- Certificate Signing Request (.csr)
- Private key (.key) - This file is private to the integrator and should not be shared with anyone.
- Certificates (.crt)

The integrator can choose tool set which he needs for this requirement, if the certificate adheres to minimal requirements. For more information, refer to Developer portal.

b. Integrator certificate generation

1. The integrator hands over the .csr file to the commissioning engineer.
2. The commissioning engineer uses the .csr file and Signify Interact Builder portal to produce the certificates for the integrator.
3. The commissioning engineer hands over the integrator certificates to the integrator.

The integrator must implement his certificates and private key in his application, which allows him to make API calls.

If the integrator wants to make use of a MQTT broker, he must use the REST API method “bmsconfig” to configure the gateway with broker settings. At that moment, the gateway will send metrics to the MQTT broker, which can be validated by the integrator.

Find out how Signify can transform your business
www.signify.com

© 2026 Signify Holding. All rights reserved. The information provided herein is subject to change, without notice. Signify does not give any representation or warranty as to the accuracy or completeness of the information included herein and shall not be liable for any action in reliance thereon. The information presented in this document is not intended as any commercial offer and does not form part of any quotation or contract, unless otherwise agreed by Signify.

All trademarks are owned by Signify Holding or their respective owners.

R02, 03 August 2026