EMSIB – IoT & Control Component

User Documentation

Authors: Aday García López & Cristian Costa

Course: Software Engineering

University: Lodz University of Technology

Academic Year: 2025/2026

Table of contents

1. Introduction

This document provides the User Documentation for the IoT & Control Component of the EMSIB system. The purpose of this document is to explain how to run, access, and use the component from a user or system operator perspective.

The IoT & Control Component is responsible for managing sensor data, device control, and real-time state monitoring inside an intelligent building. This documentation reflects the system state after the Second Integration phase.

2. Target Users

This documentation is intended for:

- System operators responsible for running the EMSIB services.
- Developers integrate this component with other EMSIB modules.
- Technical users testing or monitoring building devices.

No advanced knowledge of the internal implementation is required to use the component.

3. Component Functionality Overview

The IoT & Control Component allows users to:

- Send telemetry data from sensors.
- Register and control building devices.
- Apply optimization scenarios.
- Monitor the current state of devices in real time.

All interactions with the component are performed using REST API endpoints.

4. System Requirements

To run the IoT & Control Component, the following requirements must be met:

- Python 3.10 or higher.
- Docker (optional, recommended).
- Internet browser or API testing tool such as Postman or curl.

5. Running the Component

5.1 Running Locally

To run the component locally:

1. Install the required dependencies:
   pip install -r requirements.txt
2. Start the application using Uvicorn:
   uvicorn src.main:app --reload
3. The service will be available at:
   http://127.0.0.1:8000

5.2 Running with Docker

The component can also be executed using Docker.

1. Build the Docker image:
   docker build -t iot-control-component .
2. Run the container:
   docker run -p 8000:8000 iot-control-component

After this, the service will be accessible through the same address.

6. Accessing the API

Once the component is running, users can interact with it using HTTP requests.

FastAPI automatically provides interactive API documentation at:

`http://127.0.0.1:8000/docs`

This interface allows users to test all available endpoints directly from the browser.

7. Main API Operations

7.1 Sending Telemetry Data

Users can send telemetry data from sensors using the telemetry endpoint.

POST /iot/telemetry

This operation stores sensor data inside the system and updates the current state.

7.2 Controlling Devices

Users can send control commands to devices.

POST /iot/device-control/{deviceId}/command

This endpoint allows the system to logically execute device commands such as turning devices on or off.

7.3 Applying Optimization Scenarios

Optimization scenarios generated by higher-level EMSIB modules can be applied using:

POST /iot/optimization/apply

The component processes the scenario and updates device states accordingly.

7.4 Monitoring Device State

To retrieve the current state of a specific device:

GET /iot/state/{deviceId}

To retrieve the global system state:

GET /iot/state/live

These endpoints allow real-time monitoring of all connected devices.

8. Typical Usage Scenario

A typical usage flow of the IoT & Control Component is as follows:

1. Sensors send telemetry data to the system.
2. Device states are updated in real time.
3. Optimization scenarios are received and applied.

4. Users monitor device states through the API.

This workflow supports intelligent energy management inside the building.

9. Error Handling and Limitations

The component performs basic validation of incoming data. Invalid requests are rejected with appropriate HTTP error responses.

At the current integration stage, the component uses in-memory storage and simulated device execution. Persistent storage and real device communication are planned for future versions.

10. Conclusions

The IoT & Control Component provides an accessible and flexible interface for managing sensors and devices within the EMSIB system. The component is easy to deploy, simple to use, and ready to be integrated with other EMSIB services.

This documentation allows users to quickly understand how to operate the component without requiring knowledge of its internal implementation.