

Semester Project 3

WiFi2BLE REST API

Intro

Semester project: Distributed Software Systems with Embedded Elements

Krzysztof Sierszecki
Project Coordinator



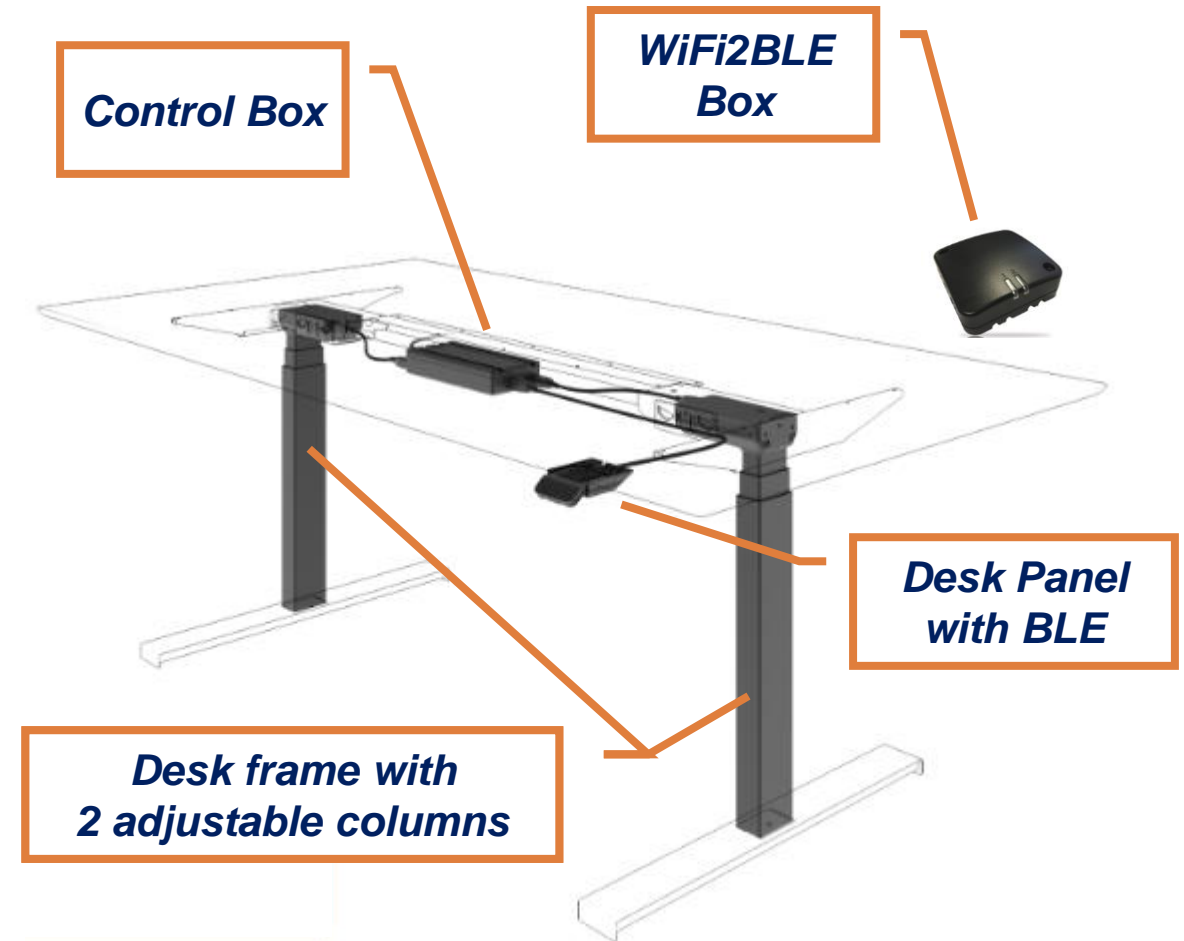
Generated by AI

Project Case Study: Desk Usage Supervision

- Obtain, visualize and analyze desk usage data for health, occupancy and maintenance
- Motorized desks are commonly used in office spaces as they can improve user working comfort
- Greater gains could be achieved by learning from the desk data, for example about the desk moving distance and frequency

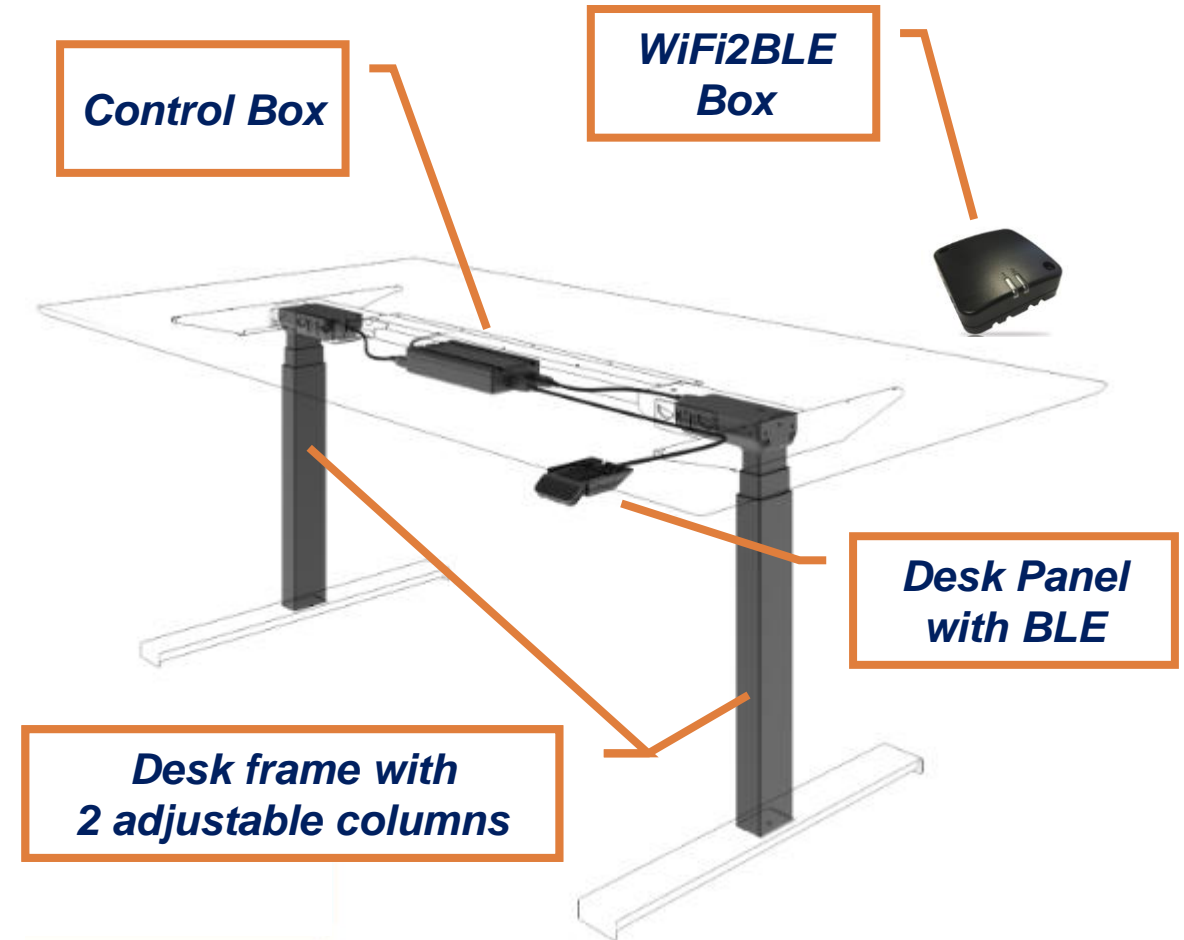
Desk System Operation

- Desks columns are controlled by the intelligent Control Box that is connected to the Desk Panel
- The desk panel accepts user commands to adjust desk height up and down
- The WiFi2BLE box exposes desk information over a Wi-Fi by translating the desk Bluetooth Low Energy (BLE) protocol to a Web API
- This allows for monitoring and controlling desks remotely
- Desk Panel has as built-in anti-collision sensor, display, and storage of favorite positions 🤖



Web API Data

- Number of desks connected to WiFi2BLE box
- Desk ID, Name, Manufacturer
- Position
 - Get and Set
- Speed
- Status
- Last errors with timestamps
- Activation counter
- Sit/stand counter



Python WiFi2BLE Simulator

- Simulation of the WiFi2BLE box Web API
- To speed up project development and testing
- Get independent of the hardware and NDAs
- Mitigate the Project Coordinator temporary “disabilities”
- Requirements: Python 3
- Very basic, for now
 - Not complete, yet
 - Desk dynamics missing



Creative use of AI 😊

REST API Basics

- Understanding REST API: <https://www.freecodecamp.org/news/how-to-use-rest-api/>
 - Course material? Web Technologies? Operating Systems?
- Base URL: <http://127.0.0.1:8000>
 - Assuming local execution
 - For testing only
- Expecting path format: `/api/<version>/<api_key>/<endpoint>`
 - Versioned API, `<version>`: v1
 - “Secured”, `<api_key>`: 32 characters, for example: E9Y2LxT4g1hQZ7aD8nR3mWx5P0qK6pV7
- Object type: JSON
 - Content type: `application/json`

Endpoints: Get all desks

→ Get all desks: retrieve list of unique IDs of all desks connected

→ URL: /desks

→ Method: GET

→ Response

→ Status: 200 OK, Body: JSON array of desk IDs

→ Status: 404 Not Found, if the endpoint does not exist

→ Example:

→ GET `http://127.0.0.1:8000/api/v1/E9Y2LxT4g1hQZ7aD8nR3mWx5P0qK6pV7/desks/`

→ 200 `["cd:fb:1a:53:fb:e6", "ee:62:5b:b8:73:1d"]`

Endpoints: Get desks by ID

→ Get desks by ID: retrieves details of a specific desk by its ID

→ URL: /desks/{desk_id}

→ Method: GET

→ Parameters:

→ {desk_id} (string): ID of the desk to retrieve

→ Response

→ Status: 200 OK, Body: JSON object of the desk if found

→ Status: 404 Not Found, if the desk with the specified ID does not exist

→ Example:

→ GET `http://127.0.0.1:8000/api/v1/E9Y2LxT4g1hQZ7aD8nR3mWx5P0qK6pV7/desks/cd:fb:1a:53:fb:e6`

→ 200 {"id": "cd:fb:1a:53:fb:e6", "name": "DESK 4486", "manufacturer": "Linak A/S", "position": 990, "speed": 0, "status": "Normal"}

Endpoints: Update desks by ID

→ Update desks by ID: updates position

→ URL: /desks/{desk_id}

→ Method: PUT

→ Parameters:

→ {desk_id} (string): ID of the desk to retrieve

→ Request Body

→ JSON object with update position field

→ Response

→ Status: 200 OK, Body: JSON with updated position, if new desk position accepted

→ Status: 404 Not Found, if the desk with the specified ID does not exist

→ Status: 400 Bad Request, if the request body is malformed

→ Example:

→ PUT `http://127.0.0.1:8000/api/v1/E9Y2LxT4g1hQZ7aD8nR3mWx5P0qK6pV7/desks/cd:fb:1a:53:fb:e6`
 `{"position": 990}`

→ 200 `{"position": 990}`

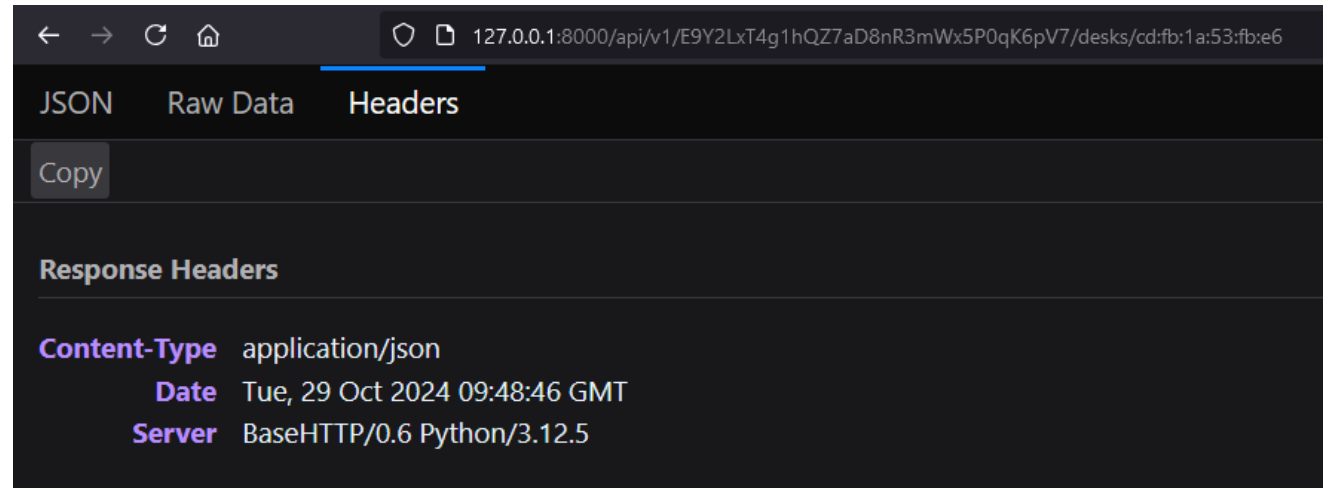
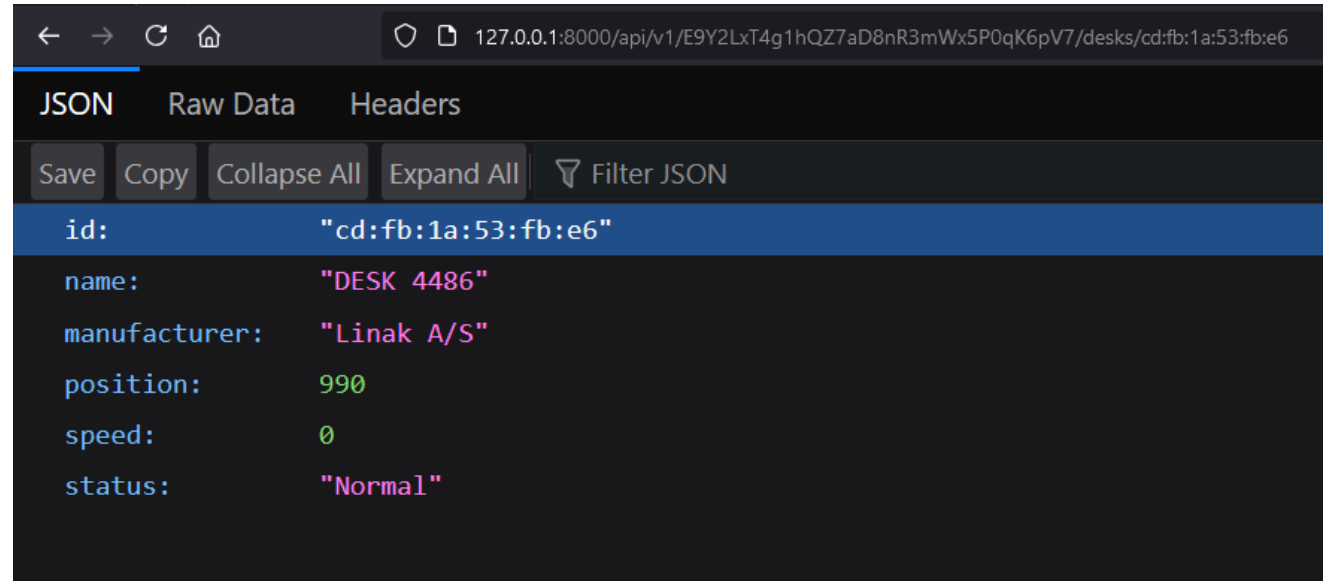
Testing REST API

→ Browser, Firefox

→ Hoppscotch

→ Open-source alternative to Postman

→ <https://docs.hoppscotch.io/>



Testing: Get all desks

GET

▼

http://127.0.0.1:8000/api/v1/E9Y2LxT4g1hQZ7aD8nR3mWx5P0qK6pV7/desks/

Parameters

Body

Headers

Authorization

Pre-request Script

Tests

Query Parameters

Key	Value
-----	-------

Status: 200 • OK Time: 30 ms Size: 42 B

JSON

Raw

Headers

3

Test Results

Response Body

1 ▼

2

3

4

[

"cd:fb:1a:53:fb:e6",

"ee:62:5b:b8:73:1d"

]

Hoppscotch in action

Testing: Get desks by ID

GET

▼

http://127.0.0.1:8000/api/v1/E9Y2LxT4g1hQZ7aD8nR3mWx5P0qK6pV7/desks/cd:fb:1a:53:fb:e6

Parameters

Body

Headers

Authorization

Pre-request Script

Tests

Query Parameters

Key	Value
-----	-------

Status: 200 • OK Time: 26 ms Size: 126 B

JSON

Raw

Headers

3

Test Results

Response Body

1

▼

{

2

"id": "cd:fb:1a:53:fb:e6",

3

"name": "DESK 4486",

4

"manufacturer": "Linak A/S",

5

"position": 990,

6

"speed": 0,

7

"status": "Normal"

8

}

Hoppscotch in action

Testing: Update desks position

PUT

http://127.0.0.1:8000/api/v1/E9Y2LxT4g1hQZ7aD8nR3mWx5P0qK6pV7/desks/cd:fb:1a:53:fb:e6

Parameters

Body

Headers

Authorization

Pre-request Script

Tests

Content Type

application/json

Override

Raw Request Body

1

{**"position"**: 990}

Status: 200

OK

Time: 14 ms

Size: 18 B

JSON

Raw

Headers

3

Test Results

Response Body

1

{

2

"position": 990

3

}

Hoppscotch in action

Thank you 🙌