API Endpoints Reference

Format for each endpoint: • **Method & Path** • **Headers** (if any) • **Request Body / Params** • **Response** • **Error Cases**

## Authentication Routes (/auth)

### POST /auth/login

• **Headers:** None  
• **Body:**

{ "username": "string", "password": "string" }

• **Response (200):**

{ "token": "<JWT token>" }

• **Errors:**

* 400: Missing fields
* 401: Invalid credentials
* 500: Server error

### POST /auth/add-user

• **Headers:** None  
• **Body:** None  
• **Response (200):** Plain text: Admin user added  
• **Errors:**

* 400: Admin user already exists
* 500: Error adding user

## Static & Sensor Data Routes (/data)

**Note:** All of these require Authorization: Bearer <token> header.

### GET /data/all

• **Method & Path:** GET /data/all  
• **Headers:**

Authorization: Bearer <token>

• **Params:** None  
• **Response (200):**

{  
 "rooms": [  
 {  
 "name": "string",  
 "lights": [ true/false , true/false ],  
 "temperature": number,  
 "humidity": number,  
 "smoke": number,  
 "noise\_level": number  
 },  
 ...  
 ],  
 "machines": [  
 {  
 "name": "string",  
 "manufacturer": "string",  
 "model": "string",  
 "installation\_date": "ISO date string",  
 "location": "string",  
 "power": true|false,  
 "temperature": number,  
 "vibration": number,  
 "power\_usage": number,  
 "noise\_level": number,  
 "maintenance": "string"  
 },  
 ...  
 ],  
 "cartons\_num": number  
}

• **Errors:** 500: Error fetching combined data

### GET /data/rooms

• **Method & Path:** GET /data/rooms  
• **Headers:**

Authorization: Bearer <token>

• **Response (200):** Array of room objects:

[  
 {  
 "name": "string",  
 "lights": [ true/false , true/false ],  
 "temperature": number,  
 "humidity": number,  
 "smoke": number,  
 "noise\_level": number  
 },  
 ...  
]

### GET /data/machines

• **Method & Path:** GET /data/machines  
• **Headers:**

Authorization: Bearer <token>

• **Response (200):** Array of machine objects:

[  
 {  
 "name": "string",  
 "manufacturer": "string",  
 "model": "string",  
 "installation\_date": "ISO date string",  
 "location": "string",  
 "power": true|false,  
 "temperature": number,  
 "vibration": number,  
 "power\_usage": number,  
 "noise\_level": number,  
 "maintenance": "string"  
 },  
 ...  
]

### GET /data/warehouse

• **Method & Path:** GET /data/warehouse  
• **Headers:**

Authorization: Bearer <token>

• **Response (200):**

{ "cartons\_num": number }

### GET /data/analytics/week

• **Method & Path:** GET /data/analytics/week  
• **Headers:**

Authorization: Bearer <token>

• **Response (200):**

{  
 "cartons\_production\_week": [ number, number, ..., number ],  
 "sales\_num\_week": [ number, number, ..., number ]  
}

• **Description:** Two arrays of length 7 representing daily totals for the last 7 days.

### GET /data/analytics/month

• **Method & Path:** GET /data/analytics/month  
• **Headers:**

Authorization: Bearer <token>

• **Response (200):**

{  
 "cartons\_production\_month": [ number, number, ..., number ],  
 "sales\_production\_month": [ number, number, ..., number ]  
}

• **Description:** Two arrays of length 30 representing daily totals for the last 30 days.

## Sensor Data Update Route (/data/update)

### PUT /data/update

• **Method & Path:** PUT /data/update  
• **Headers:** (optional) Bearer token if enabled  
• **Body:**

{  
 "rooms": [  
 {  
 "name": "string",  
 "temperature": number,  
 "humidity": number,  
 "smoke": number,  
 "noise\_level": number  
 },  
 ...  
 ],  
 "machines": [  
 {  
 "name": "string",  
 "temperature": number,  
 "vibration": number,  
 "power\_usage": number,  
 "noise\_level": number,  
 "maintenance": "string"  
 },  
 ...  
 ]  
}

• **Response (200):** Full updated SensorData document:

{  
 "\_id": "...",  
 "rooms": [ /\* as above \*/ ],  
 "machines": [ /\* as above \*/ ],  
 "createdAt": "ISO date",   
 "updatedAt": "ISO date",  
 "\_\_v": 0  
}

• **Errors:**

* 400: rooms and machines must be arrays
* 500: Error updating sensor data

## Transaction Routes (/tx)

**Note:** Authentication commented out but can be enabled.

### POST /tx/sale

• **Method & Path:** POST /tx/sale  
• **Headers:** (optional) Bearer token  
• **Body:**

{ "cartons\_sold": number, "DateTime": "ISO date string", "Buyer": "string" }

• **Response (200):** Sale document JSON:

{  
 "\_id": "...",  
 "cartons\_sold": number,  
 "DateTime": "ISO date string",  
 "Buyer": "string",  
 "\_\_v": 0  
}

• **Errors:**

* 400: Invalid fields / Insufficient cartons
* 500: Error adding sale

### POST /tx/cartons

• **Method & Path:** POST /tx/cartons  
• **Headers:** (optional) Bearer token  
• **Body:**

{ "cartons\_produced": number, "DateTime": "ISO date string" }

• **Response (200):**

{  
 "addition": {  
 "\_id": "...",  
 "cartons\_produced": number,  
 "DateTime": "ISO date string",  
 "\_\_v": 0  
 },  
 "cartons\_num": number  
}

• **Errors:**

* 400: Invalid fields
* 500: Error adding cartons

## Toggle Routes (/toggle)

**Note:** All require Authorization: Bearer <token>.

### POST /toggle/lights

• **Method & Path:** POST /toggle/lights  
• **Headers:**

Authorization: Bearer <token>

• **Body:**

{ "room\_name": "string", "light\_num": number }

• **Response (200):**

{ "room\_name": "string", "lights": [ true|false, ... ] }

• **Errors:**

* 400: room\_name (string) and light\_num (number) required
* 404: No static data found / Room not found
* 400: light\_num out of range
* 500: Error toggling light

### POST /toggle/machine

• **Method & Path:** POST /toggle/machine  
• **Headers:**

Authorization: Bearer <token>

• **Body:**

{ "machine\_name": "string" }

• **Response (200):**

{ "machine\_name": "string", "power": true|false }

• **Errors:**

* 400: machine\_name (string) required
* 404: No static data found / Machine not found
* 500: Error toggling machine power