

# IoT Sensor Backend – Documentation

Node.js • MongoDB Atlas • MQTT

## 1. Introduction :

This project implements a backend service that collects temperature readings from IoT devices.

It supports two modes of data ingestion:

1. HTTP REST API, commonly used by mobile/IoT gateways
2. MQTT, a lightweight message protocol widely used for IoT communication

All collected readings are stored in MongoDB Atlas, and the backend exposes an endpoint to fetch the most recent reading for a specific device.

This implementation was created as part of a Backend Internship Pre-Assessment Assignment.

## 2.Objectives :

- Accept sensor data using POST `/api/sensor/ingest`
- Retrieve latest sensor reading using **GET** `/api/sensor/:deviceId/latest`
- Validate inputs before inserting into the database
- Provide a fallback timestamp when missing
- Store all readings in MongoDB using Mongoose schema
- Implement MQTT subscriber for real-time ingestion

### 3. Database Schema :

Each reading stored in MongoDB follows this structure:

Field	Type	Description
deviceId	String	Unique identifier for a device
temperature	Number	Temperature reading in °C
timestamp	Number	Epoch time (ms)
createdAt	Date	Auto-generated document timestamp

### 4. Api Endpoints:

#### 1. POST /api/sensor/ingest

Stores a new sensor reading.

Sample Payload :

```
{  
  "deviceId": "sensor-01",  
  "temperature": 31.4,  
  "timestamp": 1705312440000  
}
```

#### 2. GET /api/sensor/:deviceId/latest

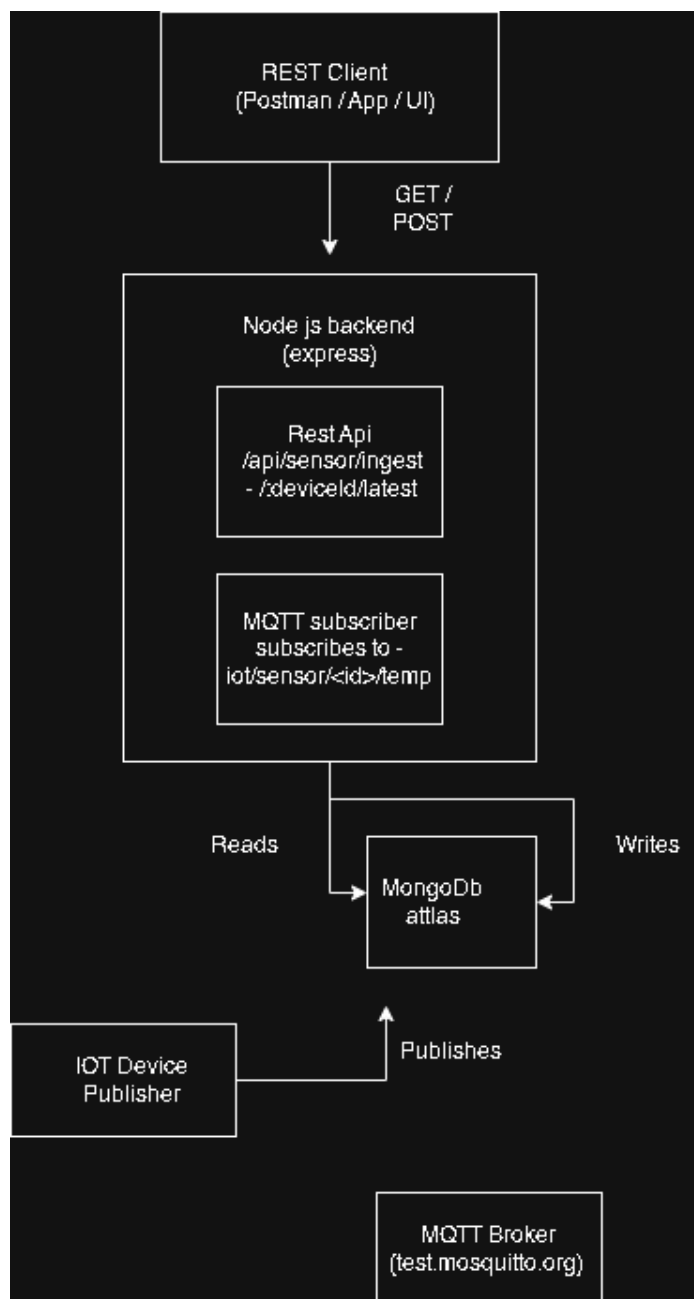
Fetches the most recent record for the given device.

Response Example:

```
{  
  "deviceId": "sensor-01",
```

```
"temperature": 29.2,  
"timestamp": 1705312440000,  
"createdAt": "2024-01-15T08:20:00.000Z"  
}
```

## 5. System Architecture



## **6. Conclusion :**

This backend demonstrates a practical IoT data ingestion pipeline using REST and MQTT.

The implementation is lightweight, easy to extend, and suitable for real-world IoT applications where devices frequently send sensor readings.

Additional improvements could include:

- Authentication for protected endpoints
- Dashboard for live data viewing
- Integration with actual IoT sensors
- Real MQTT broker deployment