

Backend Project - Samuel Deotti

The idea behind this API is to develop a backend service that manages the individual reading of water and gas consumption. To simplify the data collection process, the service will use AI to obtain measurements from photos of the meters. This approach aims to streamline the process of capturing consumption data, ensuring accuracy and reducing the need for manual intervention.

Technologies Used

- **Node.js:** Provides a runtime environment for building fast and scalable server-side applications.
- **TypeScript:** Used for type safety, ensuring better code quality and maintainability.
- **Express:** A minimalist web framework for Node.js, used to build robust APIs with routing capabilities.
- **Sequelize:** An ORM (Object-Relational Mapping) library used for database interaction, making it easier to work with SQL databases like SQLite3.
- **SQLite3:** A lightweight, file-based database used for storing customer data and measurements, ideal for development and testing environments.

AI and Image Processing

- **Gemini API:** A specialized AI service used for reading measurements from images of water and gas meters, enabling accurate and automated data collection.
- **Large Language Model (LLM):** Used to confirm or correct values read by the AI, ensuring data integrity and accuracy in the measurement process.

Other Tools

- **Docker:** Used for containerizing the application, providing consistency across different environments and simplifying the deployment process.
- **Postman:** Utilized for API testing and debugging, helping ensure that endpoints function correctly and handle various scenarios.

How to Build and Run the Application

1. Clone the project from GitHub:

```
git clone https://github.com/samueldeotti/smart-meter.git
```

or

```
git clone git@github.com:samueldeotti/smart-meter.git
```

2. Navigate to the project directory:

```
cd smart-meter
```

3. Build and Run the Docker Compose services:

```
docker-compose up --build
```

How to Stop the Backend Application

To stop the application, press **Ctrl + C** in the terminal where Docker Compose is running, or run the following command in the project root directory:

```
bash docker-compose down
```

Backend Ports

The application will run on <http://localhost:8080/>

Backend Endpoints

- **POST /upload:** Responsible for receiving a base64 image, querying Gemini, and returning the measurement read by the API.
- **PATCH /confirm:** Responsible for confirming or correcting the value read by the LLM
- **GET:** Responsible for listing the measurements taken for a specific customer.