# Flask Clients API Documentation

## 📌 Overview

This Flask API provides CRUD (Create, Read, Update, Delete) operations for managing clients, with data stored in \*\*MongoDB\*\*.

## 🚀 Features

✅ Connects to \*\*MongoDB\*\*

✅ Handles \*\*CRUD operations\*\* (`GET`, `POST`, `PUT`, `DELETE`)

✅ Accepts \*\*JSON requests\*\*

✅ Works with \*\*Postman & curl\*\*

---

## 📂 Project Structure

```

flask-clients-api/

│── app/

│ ├── \_\_init\_\_.py # Initialize Flask app

│ ├── routes.py # API endpoints

│ ├── models.py # Database models (if needed in future)

│ ├── config.py # Configuration settings

│ ├── db.py # MongoDB connection

│── tests/

│ ├── test\_api.py # Unit tests for API

│── Dockerfile # Docker configuration

│── requirements.txt # Python dependencies

│── .env # Environment variables

│── .gitignore # Ignore unnecessary files

│── README.md # Project setup instructions

│── main.py # Entry point for Flask application

```

---

## 📌 API Endpoints

### \*\*1️⃣ Get All Clients\*\*

\*\*Request:\*\*

```http

GET /clients

```

\*\*Response:\*\*

```json

[

{ "name": "John Doe", "email": "john@example.com" }

]

```

---

### \*\*2️⃣ Add a New Client\*\*

\*\*Request:\*\*

```http

POST /clients

```

\*\*Headers:\*\*

```json

Content-Type: application/json

```

\*\*Body:\*\*

```json

{

"name": "Alice",

"email": "alice@example.com"

}

```

\*\*Response:\*\*

```json

{"message": "Client added successfully"}

```

---

### \*\*3️⃣ Get a Specific Client by Email\*\*

\*\*Request:\*\*

```http

GET /clients/alice@example.com

```

\*\*Response:\*\*

```json

{ "name": "Alice", "email": "alice@example.com" }

```

---

### \*\*4️⃣ Update a Client’s Details\*\*

\*\*Request:\*\*

```http

PUT /clients/alice@example.com

```

\*\*Headers:\*\*

```json

Content-Type: application/json

```

\*\*Body:\*\*

```json

{ "name": "Alice Smith" }

```

\*\*Response:\*\*

```json

{"message": "Client updated successfully"}

```

---

### \*\*5️⃣ Delete a Client\*\*

\*\*Request:\*\*

```http

DELETE /clients/alice@example.com

```

\*\*Response:\*\*

```json

{"message": "Client deleted successfully"}

```

---

## 🛠️ \*\*Setup & Running Locally\*\*

### \*\*1️⃣ Install Dependencies\*\*

```sh

pip install -r requirements.txt

```

### \*\*2️⃣ Set Up Environment Variables\*\*

Create a `.env` file:

```env

MONGO\_URI=mongodb://flask\_user:flaskpassword@localhost:27017/clients\_db?authSource=clients\_db

```

### \*\*3️⃣ Run the Flask App\*\*

```sh

python main.py

```

### \*\*4️⃣ Test with Postman or curl\*\*

```sh

curl http://127.0.0.1:5000/clients

```

---

## 🐳 \*\*Run with Docker\*\*

### \*\*1️⃣ Build Docker Image\*\*

```sh

docker build -t flask-clients-api .

```

### \*\*2️⃣ Run Docker Container\*\*

```sh

docker run -p 5000:5000 flask-clients-api

```

---

## 🛠️ \*\*Troubleshooting\*\*

- \*\*MongoDB connection error?\*\* Ensure MongoDB is running:

```sh

mongosh

use clients\_db

show collections

```

- \*\*`415 Unsupported Media Type`?\*\* Ensure you’re sending JSON with the correct headers.

- \*\*API not responding?\*\* Check logs:

```sh

python main.py

```

---

## 🚀 Next Steps

✅ \*\*Dockerize the Flask API\*\* 🐳

✅ \*\*Set Up CI/CD Pipelines\*\* ⚙️ (GitHub Actions + Jenkins)

✅ \*\*Deploy to Kubernetes\*\* ☸️

📌 \*\*Author:\*\* GK

📌 \*\*Date:\*\* March 2025