Weather Data Backend API

This project provides a backend service to fetch historical weather data from the Open-Meteo API, store it in MongoDB, and allow users to retrieve the data through an API. The application allows users to store and view weather data, including temperature information for given latitude and longitude, within a specified date range.

Tech Stack

- **Node.js**: JavaScript runtime for building the backend.
- **Express.js**: Web framework for Node.js.
- **MongoDB**: NoSQL database for storing weather data.
- **Mongoose**: ODM (Object Data Modeling) library for MongoDB and Node.js.
- **Open-Meteo API**: Used to fetch historical weather data.
- **dotenv**: For managing environment variables.
- **Cors**: For handling cross-origin requests

Environment Variables

This project relies on the following environment variables:

- `MONGODB_URI`: The connection string to your MongoDB database.
- `PORT`: The port number on which the server will run (default is `3000`).

API Endpoints

POST /store-weather-data

Purpose: Fetch historical weather data from the Open-Meteo API and store it in MongoDB.

Request Body:

json
Copy code

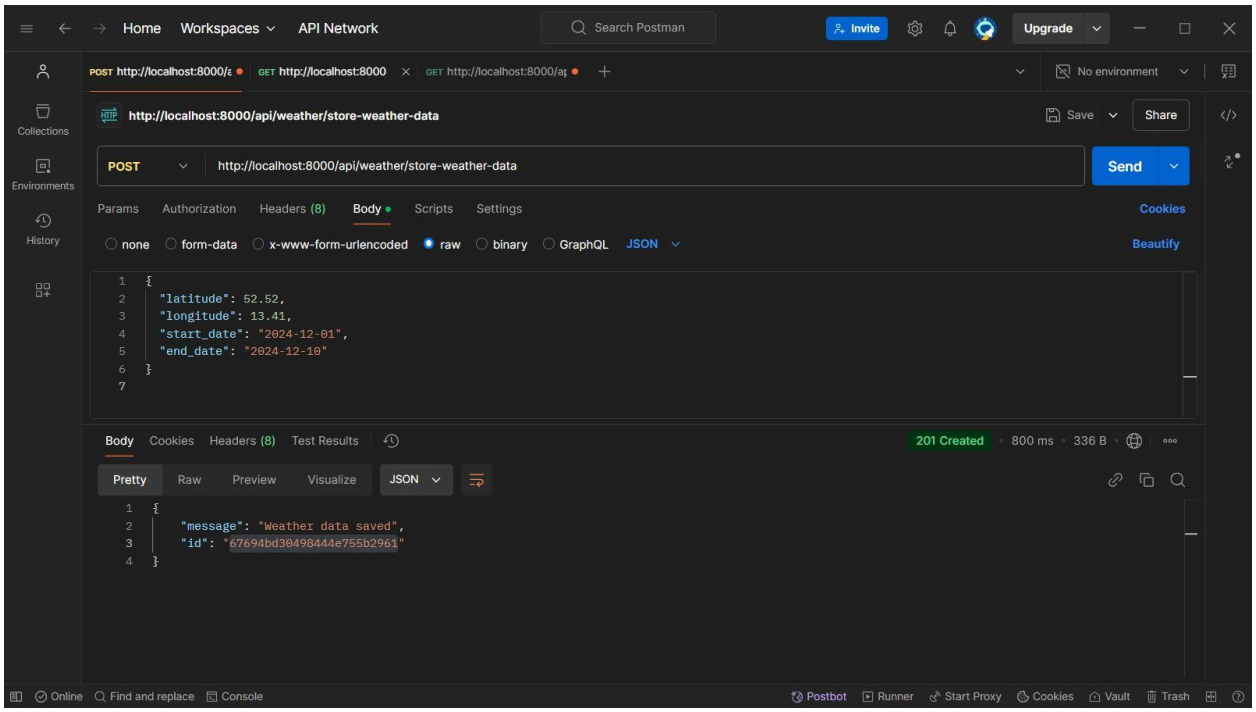
```
{
  "latitude": <float>,
  "longitude": <float>,
  "start_date": "YYYY-MM-DD",
  "end_date": "YYYY-MM-DD"
```

```
}
```

Response:

json
Copy code

```
{
  "message": "Weather data saved",
  "id": "<saved_entry_id>"
}
```



GET /list-weather-files

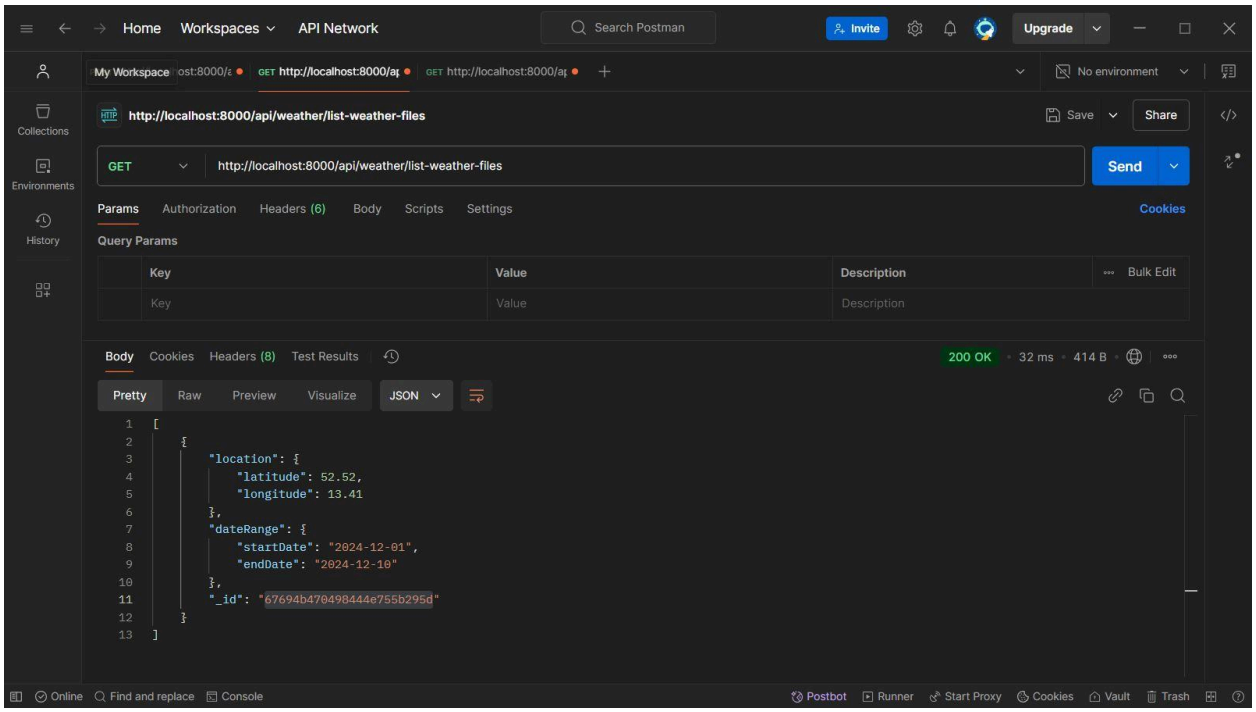
Purpose: List all weather data files stored in the MongoDB database.

Response:

json
Copy code

```
[
  {
    "_id": "<file_id>",
    "location": {
      "latitude": <float>,
      "longitude": <float>
    },
```

```
    "dateRange": {
      "startDate": "YYYY-MM-DD",
      "endDate": "YYYY-MM-DD"
    }
  },
  ...
]
```



GET /weather-file-content/:id

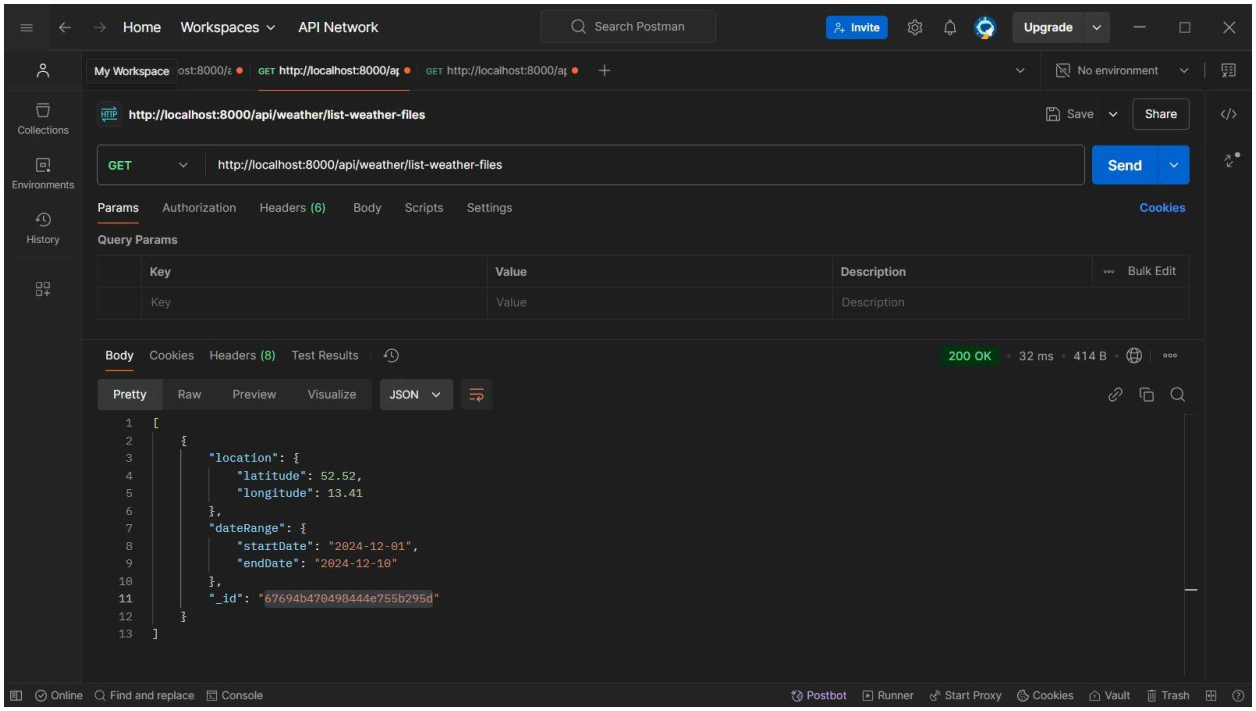
Purpose: Retrieve the content of a specific weather data file by its ID.

Response:

json
Copy code

```
{
  "_id": "<file_id>",
  "location": {
    "latitude": <float>,
    "longitude": <float>
  },
  "dateRange": {
    "startDate": "YYYY-MM-DD",
    "endDate": "YYYY-MM-DD"
```

```
  },
  "data": {
    "daily": {
      "temperature_2m_max": [...],
      "temperature_2m_min": [...],
      "temperature_2m_mean": [...],
      "apparent_temperature_max": [...],
      "apparent_temperature_min": [...],
      "apparent_temperature_mean": [...]
    }
  }
}
```



Testing the API

You can test the API using any API testing tool like Postman or cURL.

Example Requests:

POST /store-weather-data

bash

Copy code

```
POST http://localhost:3000/api/weather/store-weather-data
Content-Type: application/json
```

```
{
```
Copy code
```
  "latitude": 52.52,
  "longitude": 13.41,
  "start_date": "2024-12-04",
  "end_date": "2024-12-18"
}
```

GET /list-weather-files

bash
```
GET http://localhost:3000/api/weather/list-weather-files
```

GET /weather-file-content/:id

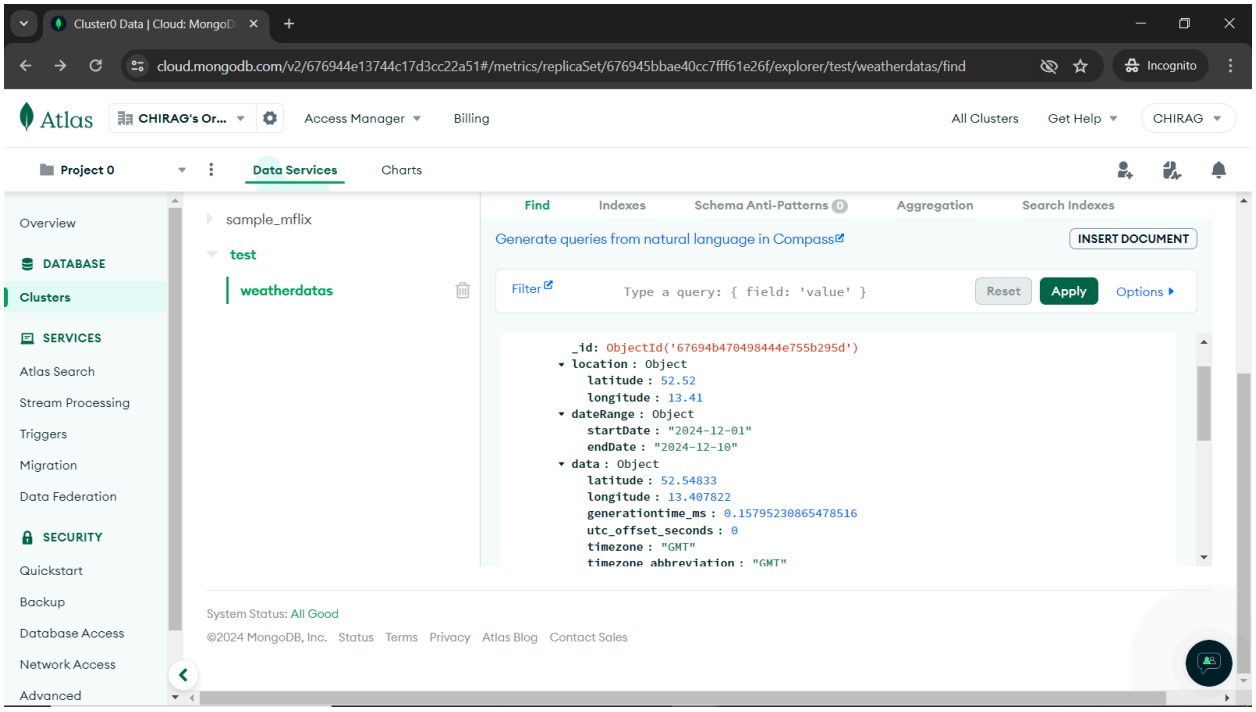
bash
Copy code
```
GET http://localhost:3000/api/weather/weather-file-content/<file_id>
```

Dependencies

Here is a list of required dependencies for this project:

- **express**: Web framework for Node.js.
- **mongoose**: MongoDB object modeling for Node.js.
- **dotenv**: Loads environment variables from a `.env` file.
- **cors**: Middleware to handle cross-origin requests.
- **body-parser**: Middleware to parse incoming request bodies (used by Express.js).
- **nodemon**

MongoDB


Cluster0 Data | Cloud: MongoD
cloud.mongodb.com/v2/676944e13744c17d3cc22a51#/metrics/replicaSet/676945bbae40cc7fff61e26f/explorer/test/weatherdatas/find
Incognito
Atlas
CHIRAG's Or...
Access Manager
Billing
All Clusters
Get Help
CHIRAG
Project 0
Data Services
Charts
Overview
DATABASE
Clusters
SERVICES
Atlas Search
Stream Processing
Triggers
Migration
Data Federation
SECURITY
Quickstart
Backup
Database Access
Network Access
Advanced
sample_mflix
test
weatherdatas
Find
Indexes
Schema Anti-Patterns
Aggregation
Search Indexes
Generate queries from natural language in Compass
INSERT DOCUMENT
Filter
Type a query: { field: 'value' }
Reset
Apply
Options
_id: ObjectId('67694b470498444e755b295d')
location : Object
latitude : 52.52
longitude : 13.41
dateRange : Object
startDate : "2024-12-01"
endDate : "2024-12-10"
data : Object
latitude : 52.54833
longitude : 13.407822
generationtime_ms : 0.15795230865478516
utc_offset_seconds : 0
timezone : "GMT"
timezone_abbreviation : "GMT"
System Status: All Good
©2024 MongoDB, Inc. Status Terms Privacy Atlas Blog Contact Sales