

Weather App

A simple Node.js + Express application that fetches and stores weather data using the OpenWeatherMap API.

Table of Contents

- Overview
 - Features
 - Installation
 - Usage
 - Starting the Server
 - Endpoints
 - Example Requests
 - Environment Variables (Optional)
 - Enhancements
 - License
-

Overview

This project demonstrates how to build a basic RESTful API with Node.js, Express, and Axios. It integrates with the OpenWeatherMap API to retrieve current weather data for a given city and optionally stores that weather info in-memory.

Features

1. **Fetch Weather by City:** Get real-time temperature and description (e.g., clear sky, scattered clouds).
 2. **Save Weather Info:** Store a city's weather data in an in-memory array.
 3. **List Saved Cities:** Retrieve all previously saved city entries.
 4. **Simple to Extend:** Swap in a real database, add front-end, or deploy to the cloud.
-

Installation

1. **Clone or Download** this repository to your local machine.
2. **Open a Terminal** in the project folder.

Install Dependencies:

```
npm install
```

3. This installs `express`, `axios`, and other required packages.
-

Usage

Starting the Server

In the **Terminal**, run:

```
npm start
```

- 1.

By default, the server will start on port **3000** and you should see:

arduino

Copy code

```
Server is running on http://localhost:3000
```

- 2.

Verify by opening your browser or a tool like [Thunder Client](#) or [Postman](#) at:

bash

Copy code

```
http://localhost:3000/api
```

You should see:

json

Copy code

```
{
  "message": "Welcome to the City Weather API!"
}
```

- 3.

Endpoints

1. **GET** `/api`
 - **Purpose:** Check if the server is running.

- **Response:** A welcome message in JSON.
- 2. **GET** `/api/weather/:city`
 - **Purpose:** Fetch the current weather for `:city` from OpenWeatherMap.
 - **Path Parameter:** `city` (e.g., `London`, `Mumbai`, `New York`).
 - **Response:** JSON with `city`, `temperature`, and `description`.
- 3. **POST** `/api/cities`
 - **Purpose:** Save a city's weather info to an in-memory array.

Request Body (JSON):

json

```
{
  "city": "Mumbai"
}
```

-
- **Response:** Weather info of the saved city.
- 4. **GET** `/api/cities`
 - **Purpose:** Retrieve all saved city weather data.
 - **Response:** An array of city weather objects.

Example Requests

GET `/api/weather/Paris`

json

```
{
  "success": true,
  "data": {
    "city": "Paris",
    "temperature": 20.31,
    "description": "light rain"
  }
}
```

- 1.
2. **POST** `/api/cities`

Body:

json

```
{
```

```
    "city": "Berlin"
  }
```

○

Sample Response:

json

```
{
  "success": true,
  "data": {
    "city": "Berlin",
    "temperature": 17.74,
    "description": "clear sky",
    "savedAt": "2025-01-08T12:34:56.789Z"
  }
}
```

○

GET /api/cities

json

```
{
  "success": true,
  "data": [
    {
      "city": "Berlin",
      "temperature": 17.74,
      "description": "clear sky",
      "savedAt": "2025-01-08T12:34:56.789Z"
    },
    {
      "city": "Paris",
      "temperature": 20.31,
      "description": "light rain",
      "savedAt": "2025-01-08T12:35:21.456Z"
    }
  ]
}
```

3.

Environment Variables (Optional)

By default, this code hardcodes your OpenWeatherMap API key in `server.js`. A better practice is to store it in an environment variable. To do so:

Install `dotenv`:

bash

```
npm install dotenv
```

1.

Create a file named `.env` (and add `.env` to your `.gitignore`):

makefile

Copy code

```
OPENWEATHER_API_KEY=YOUR_ACTUAL_KEY_HERE
```

2.

At the top of `server.js`:

js

```
require('dotenv').config();  
const OPENWEATHER_API_KEY = process.env.OPENWEATHER_API_KEY;
```

3.

4. Restart the server. Now your key won't be exposed in the public repo.

Enhancements

- **Use a Database:** Store weather info in MongoDB/PostgreSQL so it persists beyond server restarts.
- **Add Error Handling:** More descriptive error messages for network timeouts, invalid API key, or city not found.
- **Front-End:** Integrate a simple React, Vue, or plain HTML/JS page to display and update data.
- **Deployment:** Push to platforms like [Render](#) or [Railway](#) so it's accessible online.

License

This project is open-source; you may use it as a starter for your own weather applications. Check the LICENSE file if included, or consider MIT License.