

API documentation

The Weather API provides real-time weather information for various cities. It allows users to get weather data, add new weather data, update existing data, and delete data. This API is suitable for developers who need to incorporate weather information into their applications or services.

The base URL for the API is: `http://localhost:5000`

Endpoint Structure:

The API supports the following endpoints:

- GET /cities: Retrieve a list of city names and their respective weather data.
- POST /cities: Add new weather data for a city.
- GET /cities/:cityName: Retrieve weather data for a specific city.
- PUT /cities/:cityName: Update weather data for a specific city.
- DELETE /cities/:cityName: Delete weather data for a specific city.

Request Parameters:

- page (optional): The page number to retrieve when querying the GET /cities endpoint. Default value is 0.
- cityName (required): The name of the city for which to retrieve, update, or delete weather data. Must be a string.

Request Headers:

None

Request Examples:

- GET /cities: Retrieves the first page of weather data for all cities.

```
url -X GET https://weather-api.com/api/v1/cities
```

- GET /cities?page=2: Retrieves the second page of weather data for all cities.

```
curl -X GET https://weather-api.com/api/v1/cities?page=2
```

- POST /cities: Adds new weather data for a city.

```
curl -X POST -H "Content-Type: application/json" -d '{"cityName": "New York", "temperature": 25, "windSpeed": 10, "precipitation": 0.5}' https://weather-api.com/api/v1/cities
```

- GET /cities/New York: Retrieves weather data for the city of New York.

```
curl -X GET https://weather-api.com/api/v1/cities/New%20York
```

- PUT /cities/New York: Updates weather data for the city of New York.

```
curl -X PUT -H "Content-Type: application/json" -d '{"temperature": 28}'  
https://weather-api.com/api/v1/cities/New%20York
```

- DELETE /cities/New York: Deletes weather data for the city of New York.

```
curl -X DELETE https://weather-api.com/api/v1/cities/New%20York
```

Response Structure:

- GET /cities and GET /cities/:cityName endpoints return a JSON object containing the following properties:
 - cityName (string): The name of the city.
 - temperature (number): The temperature in degrees Celsius.
 - windSpeed (number): The wind speed in kilometers per hour.
 - precipitation (number): The precipitation amount in millimeters.
- POST /cities and PUT /cities/:cityName endpoints return a JSON object containing the newly added or updated weather data, respectively.
- DELETE /cities/:cityName endpoint returns a JSON object containing a success message.

Regenerate response

GET /weather/cityNames

Description:

Gets the page parameter from the request query or uses the default value of 0. It then retrieves a maximum of 10 weather data entries from the database, skips entries based on the page number, and returns the cityName property of the entries as a JSON response.

Request Query Parameters:

- page (optional): Number of pages to skip. Default value is 0.

Request Headers:

None

Response:

Status Code: 200 OK

Response Body:

```
{ "cities": [ { "cityName": "New York" }, { "cityName": "Los Angeles" }, ... ] }
```

Possible Error Codes:

- 500 Internal Server Error: If there is an internal server error.

4.2. POST /weather

Description:

Creates a new weather data entry in the database and returns the saved data as a JSON response.

Request Body:

```
{ "cityName": "New York", "temperature": 20, "windSpeed": 30, "precipitation": 0.3 }
```

Request Headers:

- Content-Type: application/json

Response:

Status Code: 201 Created

Response Body:

```
{ "cityName": "New York", "temperature": 20, "windSpeed": 30, "precipitation": 0.3 }
```

Possible Error Codes:

- 400 Bad Request: If the request body is not valid.
- 500 Internal Server Error: If there is an internal server error.

4.3. GET /weather/:cityName

Description:

Retrieves the weather data of a specific city from the database based on the `cityName` parameter in the request URL. If no data is found, it throws an exception with a 404 status code. Otherwise, it returns the retrieved data as a JSON response.

Request Parameters:

- `cityName`: Name of the city whose weather data is to be retrieved.

Request Headers:

None

Response:

Status Code: 200 OK

Response Body:

```
{ "cityName": "New York", "temperature": 20, "windSpeed": 30, "precipitation": 0.3 }
```

Possible Error Codes:

- 404 Not Found: If no weather data is found for the given city name.
- 500 Internal Server Error: If there is an internal server error.

4.4. PUT /weather/:cityName

Description:

Updates the weather data of a specific city in the database based on the `cityName` parameter in the request URL. It creates a new `WeatherModel` instance with the `cityName` parameter from the request URL and the request body data. It then validates the data, updates the weather data of the specified city in the database, and returns a JSON response with a success message.

Request Parameters:

- `cityName`: Name of the city whose weather data is to be updated.

Request Body:

```
{ "temperature": 30, "windSpeed": 40, "precipitation":
```