

# Weather.GOV API Project

**Description:** As a user I need to be able to call a REST endpoint to get a weather forecast for a certain date by a location's latitude and longitude.

## Requirements:

- Create a Spring web service using Java 21 that exposes a REST endpoint `/weather/forecast/{lat},{long}?date={date}&metric={boolean}` that returns a JSON formatted response containing the forecast details for the requested date.
  - *Lat/Long* is a required path parameter
    - Example JB Hunt corporate Lat/Long is 36.244/-94.149
    - Validation is recommended and the input should be rounded to 3 decimal places
  - *date* (optional parameter) The date of the forecast to be fetched. If not provided, default to the current date.
  - *metric* (optional parameter) Boolean, if true, convert the weather forecast values to metric. If not provided, default to false (Weather.gov API uses Imperial units by default).
- The forecast response should be similar but does not have to be exact to the following:

```
1  {
2    "latitude": 36.244,
3    "longitude": -94.149,
4    "date": "2025-03-15",
5    "forecast": "Day: Mostly Sunny then Slight Chance Rain Showers - Night: Partly Cloudy",
6    "pop": 65,
7    "temperature": {
8      "high": 70,
9      "low": 45,
10     "unit": "F"
11   },
12   "wind": {
13     "maxSpeed": 15,
14     "minSpeed": 5,
15     "direction": "NNW",
16     "unit": "mph"
17   }
18 }
```

- **Weather Data:**

- Utilize Weather.gov's free API for weather forecast data. Read the documentation and determine which endpoints to call to obtain the necessary weather forecast data.
  - Weather.gov API Documentation can be found here: <https://www.weather.gov/documentation/services-web-api> and <https://weather-gov.github.io/api/>
  - The Weather.gov API only returns weather up to 7 days out from the current date, handle requests with dates in the past or outside of this range with appropriate response messages.
  - Lat/Long values are used to determine which weather office and grid X/Y to use to obtain the areas weather forecast
  - Weather.gov splits daily forecasts into their own day and night forecast, these should be combined into a single day forecast (i.e. use the Day and Night temperature or wind speeds to determine the High/Low temperature or Min/Max wind speed for the entire day).
  - Weather.gov displays wind speed as a string in the format of “5 to 15 mph” or just “10 mph”, parse and extract these values accordingly to determine the min/max for the day
  - POP (probability of precipitation) of an entire day can be taken as the max pop between day and night.
  - If the user requests metric values, convert the temperature from Fahrenheit to Celsius and wind speed from MPH to KPH. You are responsible for implementing these calculations.
  - Forecast should be the combined Day and Night ‘shortForecast’ in the format “Day: {dayForecastText} – Night: {nightForecastText}”

- Use Spring Data to connect with a database.

- A requested Lat/Long’s determined Office ID along with its respective grid X and grid Y coordinates should be stored to a table. This should first be checked and used if it exists first before calling the Weather.gov API.
- Fetched forecasts should be stored to a table so that when a user requests a forecast for a Lat/Long, it should first check the table if the forecast data exists and use that before calling the Weather.gov API.
  - If the stored forecast data is over an hour old, get a new forecast for that date from the Weather.gov API (multiple forecasts can be stored for a

single date so pulling saved forecast should go off most recent forecast timestamp).

- Optional: Incorporate Redis caching in any part of the solution
  
- **Testing**
  - Create Integration and JUnit tests (JUnit 5) to cover and test different scenarios
    - Use mocked weather data from the weather.gov API
    - Optional: Use Test Containers (<https://testcontainers.com/>) to wire up to the data base and caching (if used)

Provide the code in a public repository.