Drache-o-metre

See also the PDF version.

Overview

Drache-o-metre provides accurate and up-to-date weather forecasts based on the user's current GPS location. Built with Android Studio, it uses the OpenWeather API to deliver detailed weather information and customizable settings. Drache-o-metre is a project within our mobile computing minor of our 3rd year of informatic licence.

Activities

1. **Current Weather (MainActivity)**

    - Displays the current weather for the user's location.
    - Includes daily and hourly forecasts for the current week.

2. **Detailed Weather**

    - Provides detailed weather predictions for the upcoming days.
    - Displays minimum and maximum temperatures, humidity percentage, and other essential data.

3. **Settings**

    - Manage notification preferences.
    - Access additional information about the application.

Technologies Used

- **Android Studio**: Primary development environment.
- **OpenWeather API**: Fetches real-time weather data.
- **GPS Integration**: Retrieves the user's current location for accurate forecasts.

App Intents

- **Navigation Intents**:
    - Navigate between the three main activities: Current Weather, Detailed Weather, and Settings.
- **Broadcast Intents**:
    - Handle location updates and weather data refresh in the background.

Background Services and Threads

- **GPS Background Service**:
    - Continuously tracks the user's location even when the app is minimized, ensuring weather data remains relevant.
- **Worker Threads**:

    - Fetch weather data asynchronously from the OpenWeather API to ensure a smooth user experience without blocking the main thread.

Sensors Used

- **GPS Sensor**:
    - Utilized to determine the user's current location for accurate weather forecasts.

Installation

1. Clone the repository:

    ```
    git clone https://github.com/h4ggstrom/drache-o-metre
    ```

2. Open the project in Android Studio.

3. Add your OpenWeather API key:

    - Navigate to `app\src\main\java\com\example\drache_o_metre\data\interact\responses`.

    - Create `ApiKey.java`

    - Add theses information, replacing `"YOUR APIKEY"`:

        ```
        package com.example.drache_o_metre.data.interact.responses;
        public class ApiKey {
            private String apiKey = "YOUR APIKEY";

            public String getApiKey() {
               return apiKey;
            }
         }
        ```

4. Build and run the project on an Android device or emulator.

How It Works

- The app uses GPS to determine the user's location.
- The OpenWeather API fetches weather data based on the coordinates.

Dependencies

- OpenWeather API
- Android SDK
- Java
- Retrofit (for API calls)

- GSON-converter (for API responses)

License

Authors

- Robin de Angelis, L3 Informatique : robin.de-angelis@etu.cyu.fr
- Killian Treuil, L3 Informatique : killian.treuil@etu.cyu.fr

Additional Notes

- Ensure your device has GPS enabled for accurate location-based forecasts.
- For support or feature requests, contact us via GitHub or email.

Check out the GitHub repository.

Realized for Paris Cergy University, 2024.