Weather Dashboard Project Setup Guide

Get Organized, Get Started, Get Coding!

Michael Borck

Introduction

Welcome to the Weather Dashboard project! This guide will walk you through the initial setup of your project folder, creating a new Python environment, and setting up your GitHub repository. By following these steps, you'll establish a strong foundation for developing your Weather Dashboard app, ensuring your project is well-organised and version-controlled.

Project Overview

In this project, you'll be creating a Weather Dashboard application that retrieves and displays weather data using the `requests` package and the openweathermap api. You'll learn how to structure your project, manage dependencies, and use Git and GitHub for version control. This setup guide is designed to help you get started quickly and efficiently.

Objectives

1. **Create and structure your project folder:** Organise your project files into a logical structure with subfolders for notebooks, scripts, documentation, and tests.
2. **Set up a Python environment:** Ensure you have an isolated environment for your project with the necessary dependencies installed.
3. **Initialise a GitHub repository:** Learn how to create a remote repository, clone it locally, and synchronise your work with GitHub.

By the end of this guide, you will have a fully functional project setup that is ready for development. Let's get started!

Step-by-Step Instructions

Follow these detailed steps to set up your Weather Dashboard project:

Step 1: Create a GitHub repository

- Go to the GitHub website and create a new repository.
- Name the repository something like "weather-dashboard".
- Initialise the repository with a README file and a LICENSE file.

Step 2: Clone the GitHub repository

- Open a terminal or command prompt and navigate to the directory where you want to clone your repository.
- Clone the GitHub repository to your local machine by running the following command:

```
git clone https://github.com/your-username/weather-dashboard.git
```

- Navigate to the cloned repository by running the following command:

```
cd weather-dashboard
```

Step 3: Create the project folder structure

- Inside the cloned repository, create the following subfolders:

```
mkdir notebooks scripts docs tests
```

Step 4: Create a new Python environment

- Create a new Python environment by running the following command:

```
conda create --name weather-dashboard-env python=3.12
```

- Activate the new environment by running the following command:

```
conda activate weather-dashboard-env
```

Step 5: Create a requirements.txt file

- Inside the project folder, create a new file called `requirements.txt`.
- Open the file and add the following line:

```
requests==2.25.1
```

  (or the latest version you need)

- Save the file.

Step 6: Install packages

- Activate the new environment if not already activated by running the following command:

```
conda activate weather-dashboard-env
```

- Install the packages listed in the `requirements.txt` file by running the following command:

```
pip install -r requirements.txt
```

Step 7: Set up the workflow

- Add all the files in the project folder to the Git repository by running the following command:

```
git add .
```

- Commit the changes by running the following command:

```
git commit -m "Initial commit"
```

- Push the changes to the GitHub repository by running the following command:

```
git push -u origin master
```

- From now on, you can work locally, make changes, and commit them to the GitHub repository by running the following commands:

```
git add .
git commit -m "commit message"
```

Step 8: Create a README and LICENSE file

- Inside the project folder, create a new file called `README.md` and add a brief description of your project.
- Inside the project folder, create a new file called `LICENSE` and add the license terms for your project.

By following these steps, students should have a basic setup for their Weather Dashboard app project, including a new Python environment, a requirements.txt file, and a GitHub repository.