

Wetter-App – Dokumentation

Diese Dokumentation beschreibt die im Rahmen des Berufsschulunterrichts entwickelte Wetter-App. Sie enthält alle notwendigen Informationen zur Installation, Bedienung und Konfiguration für zukünftige Anwenderinnen und Anwender.

Einleitung

Die Wetter-App ist eine einfache Anwendung zur Anzeige der aktuellen Temperaturdaten, basierend auf der **OpenMeteo API**. Ziel des Projekts war es, ein praxisnahes Programmierbeispiel für die Klasse **FA 11.1** zu entwickeln.

Die App wurde mit **Python 3.13** entwickelt und getestet. Ältere Versionen sind möglicherweise nicht kompatibel und werden nicht unterstützt. Die Entwicklung und Tests erfolgten unter **Linux**, konkret mit **Pop!_OS (Ubuntu 22.04 LTS / Debian)**. Zusätzlich wurden Funktionstests unter **Windows** durchgeführt, ohne dabei auf Probleme zu stoßen.

Benötigte Bibliotheken:

- `matplotlib==3.10.3`
- `openmeteo-requests==1.4.0`
- `pandas==2.2.3`
- `plotly==5.24.1`
- `requests-cache==1.2.1`
- `retry-requests==2.0.0`
- `seaborn==0.13.2`
- `streamlit==1.37.1`

Andere Betriebssysteme werden derzeit nicht offiziell unterstützt.

Zum Anzeigen der Benutzeroberfläche wird ein Webbrowser benötigt. Getestet wurden **Firefox**, **Microsoft Edge** und **Google Chrome**. Bei anderen Browsern kann es zu Kompatibilitätsproblemen kommen.

Installation

1. ZIP-Datei entpacken
2. Abhängigkeiten installieren (z. B. via `pip install -r requirements.txt`)
3. Im Root-Verzeichnis folgenden Befehl im Terminal ausführen:
`PYTHONPATH=src streamlit run src/application/ui/webapp_ui.py`

Benutzeroberfläche

Die Oberfläche ist zweigeteilt:

Linke Seite – Eingabe

- **Country:** Land
- **Postal Code:** Postleitzahl
- **City:** Stadt
- **Time Interval:** Auswahl zwischen **Days** (tagesweise) und **Hours** (stundenweise)
- **Time Span (Days):** Anzahl der Tage, für die Daten abgefragt werden sollen
- **Go-Button:** Startet die Suche
- **Download CSV:** Exportiert die Daten als CSV-Datei

Rechte Seite – Anzeige

Man kann zwischen **Summary View** (Zusammenfassung) und **Detailed Analysis** (Detailanalyse) wechseln.

Summary View / Day

- Anzeige der Wetterdaten des aktuellen Tages
- Übersicht über Temperatur, gefühlte Temperatur und Windverhältnisse

Detailed Analysis / Day

Die Daten werden tabbasiert dargestellt:

- **Temperature:** Maximal-, Minimal-, Mittelwert- sowie gefühlte Temperaturen
- **Precipitation:** Regenmenge (`rain_sum`), Schauer (`showers_sum`) und Schneefall (`snowfall_sum`)
- **Wind:** Windgeschwindigkeiten
- **Visibility & Clouds:** Sichtweite (in Metern), Wolkenbedeckung in Prozent
- **Wind Gusts:** Windböen

Detailed Analysis / Hour

- **Temperature:** Stündliche Temperatur mit Vergleich zur gefühlten Temperatur
- **Precipitation:** Stündliche Niederschlagswerte
- **Wind:** Stündliche Windgeschwindigkeiten
- **Visibility & Clouds:**
 - **visibility:** Sichtweite in Metern
 - **cloud_high:** Wolken über 8 km
 - **cloud_mid:** Wolken zwischen 3–8 km
 - **cloud_low:** Wolken unter 3 km
- **Wind Gusts:** Stündliche Windböen

FAQ (Häufig gestellte Fragen)

F: Wie ändere ich die Standardwerte für Stadt, Land und Postleitzahl?

A: In der Datei `settings.py` können die Variablen `default_city`, `default_country` und `default_postal_code` angepasst werden.

F: Die Diagramme sind zu überladen. Wie kann ich einen bestimmten Zeitraum auswählen?

A: Mit der Maus einen Bereich im Diagramm markieren – dieser wird vergrößert dargestellt. Mit einem Doppelklick wird die Originalansicht wiederhergestellt.

F: Ich finde den Download-Button nicht.

A: Der Button erscheint erst, nachdem Daten erfolgreich geladen wurden. Fülle die Eingabefelder aus und klicke auf „Go“.

F: Wie kann ich einen Plot als PNG speichern?

A: Fahre mit der Maus über den Plot und klicke auf das Kamera-Symbol.