

Dokumentation

Projektaufgabe: Aufbau einer VM zur Datensammlung, Speicherung und Visualisierung

Datenbank:

Name: weather_data

Tabelle weather

Die Tabelle weather speichert verschiedene Wetterdaten. Im Folgenden werden die Struktur der Tabelle sowie die vorhandenen Datensätze erläutert.

Tabellenstruktur

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
city_name	varchar(255)	YES		NULL	
temperature	float	YES		NULL	
feels_like_temp	float	YES		NULL	
temp_min	float	YES		NULL	
temp_max	float	YES		NULL	
weather_description	varchar(255)	YES		NULL	
wind_speed	float	YES		NULL	
cloudiness	int(11)	YES		NULL	
timestamp	timestamp	NO		current_timestamp()	

Beschreibung der Felder

- **id:** Eindeutiger Identifikator für jeden Datensatz. Datentyp: int(11).
- **city_name:** Name der Stadt, für die die Wetterdaten gelten. Datentyp: varchar(255).
- **temperature:** Aktuelle Temperatur in Grad Celsius. Datentyp: float.
- **feels_like_temp:** Gefühlte Temperatur in Grad Celsius. Datentyp: float.
- **temp_min:** Minimale Temperatur in Grad Celsius. Datentyp: float.
- **temp_max:** Maximale Temperatur in Grad Celsius. Datentyp: float.
- **weather_description:** Beschreibung der Wetterlage (z.B. "Few clouds", "Light rain"). Datentyp: varchar(255).
- **wind_speed:** Windgeschwindigkeit in Meter pro Sekunde. Datentyp: float.
- **cloudiness:** Bewölkungsgrad in Prozent. Datentyp: int(11).
- **timestamp:** Zeitstempel der Wetteraufzeichnung. Datentyp: timestamp.

Indizes

Die Tabelle weather hat einen Primärschlüssel auf der Spalte id.

```
ALTER TABLE `weather`  
  ADD PRIMARY KEY (`id`);
```

AUTO_INCREMENT

Das Feld id ist ein AUTO_INCREMENT-Feld, welches automatisch einen einzigartigen Wert generiert, wenn ein neuer Datensatz eingefügt wird.

```
ALTER TABLE `weather`  
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=26;
```

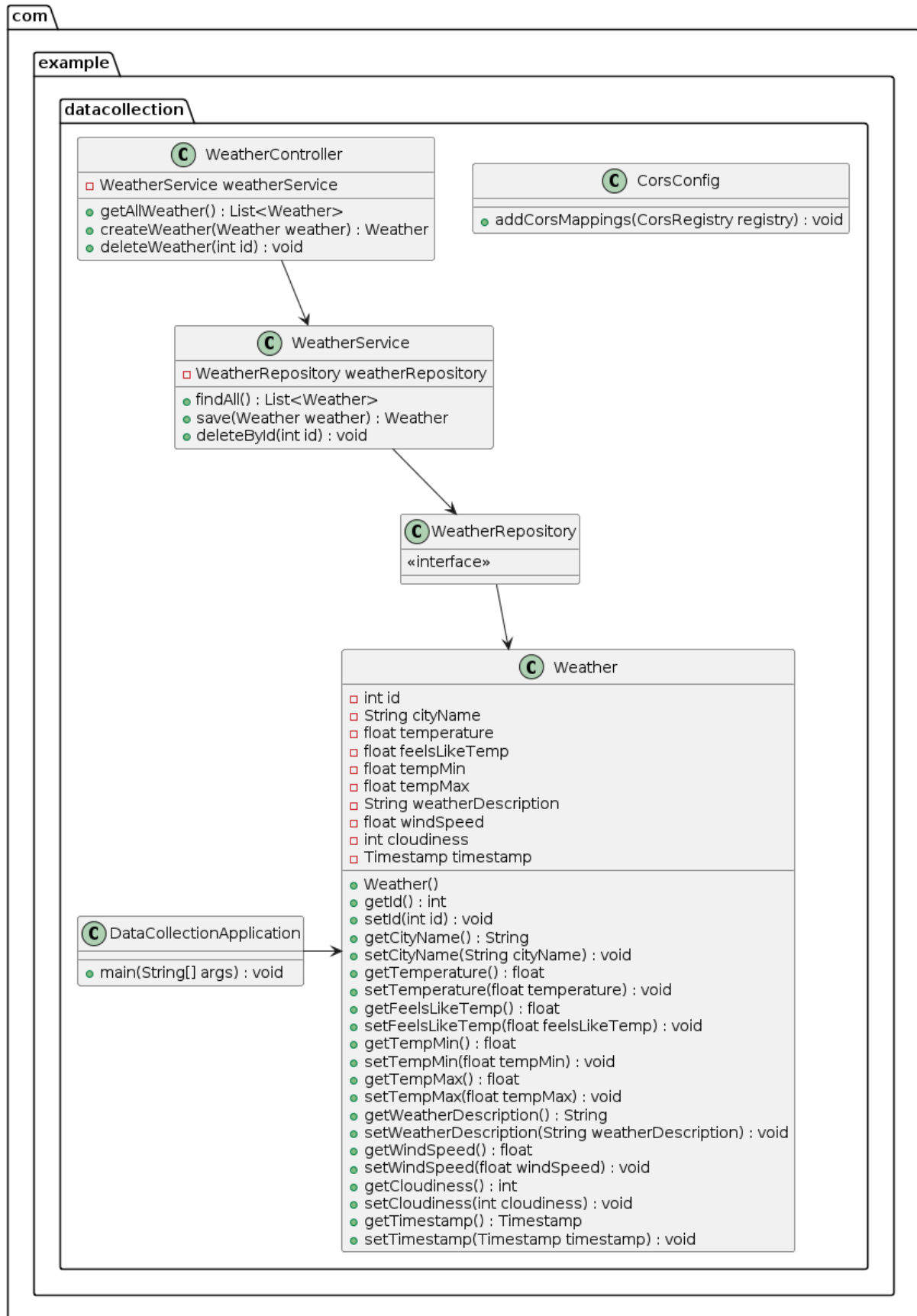
Beispielhafte Datensätze

Hier sind einige der Datensätze, die in der Tabelle weather gespeichert sind:

```
MariaDB [weather_data]> SELECT * FROM `weather`;
```

id	city_name	temperature	feels_like_temp	temp_min	temp_max	weather_description	wind_speed	cloudiness	timestamp
10	Tett nang Castle	24.1	24.42	22.51	26.04	Few clouds	3.09	20	2024-06-27 13:55:53
11	Tett nang Castle	24.1	24.42	22.51	26.04	Few clouds	3.09	20	2024-06-27 13:57:51
12	Tett nang Castle	24.1	24.42	22.51	26.04	Few clouds	3.09	20	2024-06-27 13:57:53
13	Tett nang Castle	24.1	24.42	22.51	26.04	Few clouds	3.09	20	2024-06-27 13:57:55
14	Tett nang Castle	24.1	24.42	22.51	26.04	Few clouds	3.09	20	2024-06-27 13:58:12
16	Tett nang Castle	23.91	24.26	22.51	26.04	Light rain	4.12	20	2024-06-27 14:19:19
17	Tett nang Castle	23.89	24.24	23.57	25.37	Light rain	4.12	20	2024-06-27 14:31:47
18	Tett nang Castle	23.89	24.24	23.57	25.37	Light rain	4.12	20	2024-06-27 14:32:11
19	Tett nang Castle	23.89	24.24	23.57	25.37	Light rain	4.12	20	2024-06-27 14:32:14
20	Tett nang Castle	23.89	24.24	23.57	25.37	Light rain	4.12	20	2024-06-27 14:32:15
21	Tett nang Castle	23.89	24.24	23.57	25.37	Light rain	4.12	20	2024-06-27 14:32:17
22	Tett nang Castle	23.1	23.37	21.57	28.26	Light rain	3.6	20	2024-06-27 14:49:21
23	Tett nang Castle	23.62	23.92	21.57	26.25	Few clouds	3.6	20	2024-06-27 15:18:46
24	Tett nang Castle	24.19	24.47	22.51	25.37	Few clouds	4.12	20	2024-06-27 15:36:11
25	Tett nang Castle	25.04	25.35	23.57	27.04	Few clouds	4.12	20	2024-06-27 16:02:54
26	Tett nang Castle	25.52	25.8	24.17	26.04	Broken clouds	4.12	61	2024-06-27 17:06:14
27	Tett nang Castle	21.07	21.5	20.51	23.47	Clear sky	1.54	0	2024-06-28 09:58:24
28	Tett nang Castle	22.43	22.82	22.15	24.72	Scattered clouds	0.45	30	2024-06-28 10:02:53
29	Tett nang Castle	22.32	22.72	21.57	25.43	Clear sky	2.06	0	2024-06-28 10:11:01
30	Tett nang Castle	22.55	23	21.57	25.43	Clear sky	2.06	0	2024-06-28 10:21:01

UML-Diagramm:



Endpoints inkl. JSON-Format Beschreibung

1. Endpoint zum Abrufen aller Wetterdaten

URL: /weather

HTTP-Methode: GET

Beschreibung: Dieser Endpoint gibt alle gespeicherten Wetterdaten zurück.

The screenshot shows a web browser interface for testing HTTP requests. The address bar displays `http://localhost:8080/weather`. The request method is set to `GET`. The response body is displayed in JSON format, showing two weather records for 'Tett nang Castle'.

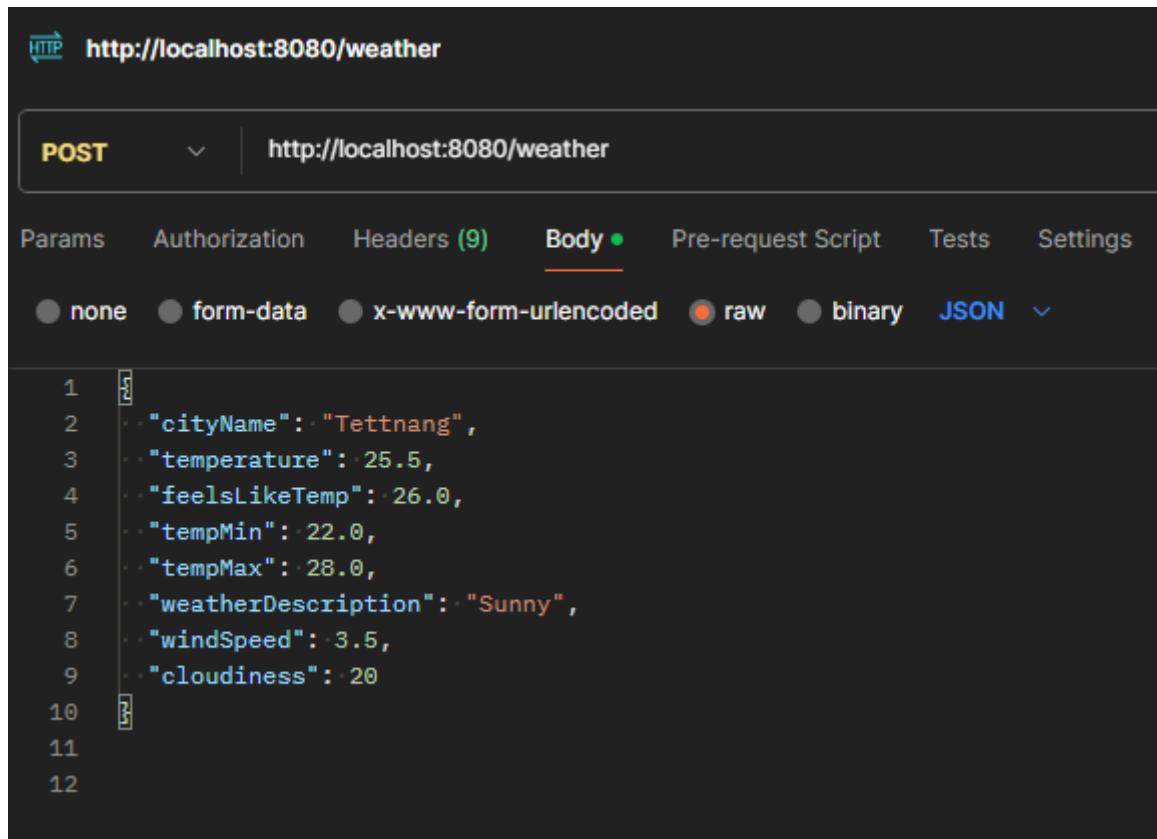
```
[
  {
    "id": 10,
    "cityName": "Tett nang Castle",
    "temperature": 24.1,
    "feelsLikeTemp": 24.42,
    "tempMin": 22.51,
    "tempMax": 26.04,
    "weatherDescription": "Few clouds",
    "windSpeed": 3.09,
    "cloudiness": 20,
    "timestamp": "2024-06-27T11:55:53.000+00:00"
  },
  {
    "id": 11,
    "cityName": "Tett nang Castle",
    "temperature": 24.1,
    "feelsLikeTemp": 24.42,
    "tempMin": 22.51,
    "tempMax": 26.04,
    "weatherDescription": "Few clouds",
    "windSpeed": 3.09,
    "cloudiness": 20,
    "timestamp": "2024-06-27T11:57:51.000+00:00"
  },
]
```

2. Endpoint zum Erstellen neuer Wetterdaten

URL: /weather

HTTP-Methode: POST

Beschreibung: Dieser Endpoint erstellt neue Wetterdaten.

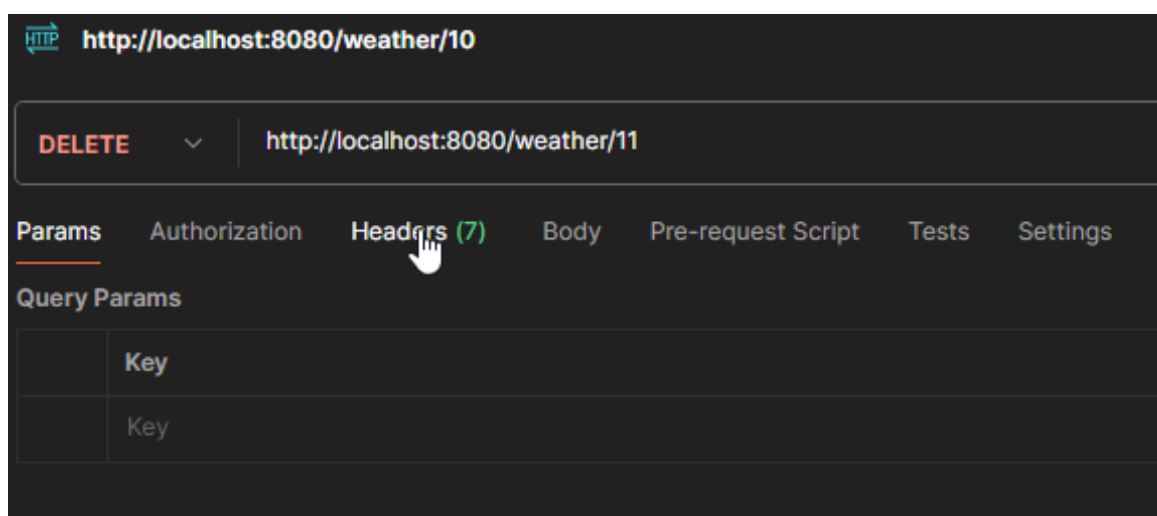


3. Endpoint zum Löschen von Wetterdaten

URL: /weather/{id}

HTTP-Methode: DELETE

Beschreibung: Dieser Endpoint löscht die Wetterdaten mit der angegebenen ID.



Client-Skript zur zyklischen Anfrage

Die Codebeschreibung kann den Kommentaren entnommen werden, das Skript wird über crontab zyklisch ausgeführt.

```
import requests
import json

# OpenWeatherMap API Konfiguration
api_key = "c24d32d7ec1cac6df361f081baf588d8"
base_url = "http://api.openweathermap.org/data/2.5/weather?"
city_name = "Tetttnang"

# Komplette URL für die API-Anfrage zusammenstellen
complete_url = f"{base_url}q={city_name}&appid={api_key}&units=metric"

# API-Anfrage an OpenWeatherMap senden
response = requests.get(complete_url)

# Daten aus der API-Antwort extrahieren
data = response.json()

# Erforderliche Daten extrahieren
city_name = data['name']
temperature = data['main']['temp']
feels_like_temp = data['main']['feels_like']
temp_min = data['main']['temp_min']
temp_max = data['main']['temp_max']
weather_description = data['weather'][0]['description']
wind_speed = data['wind']['speed']
cloudiness = data['clouds']['all']

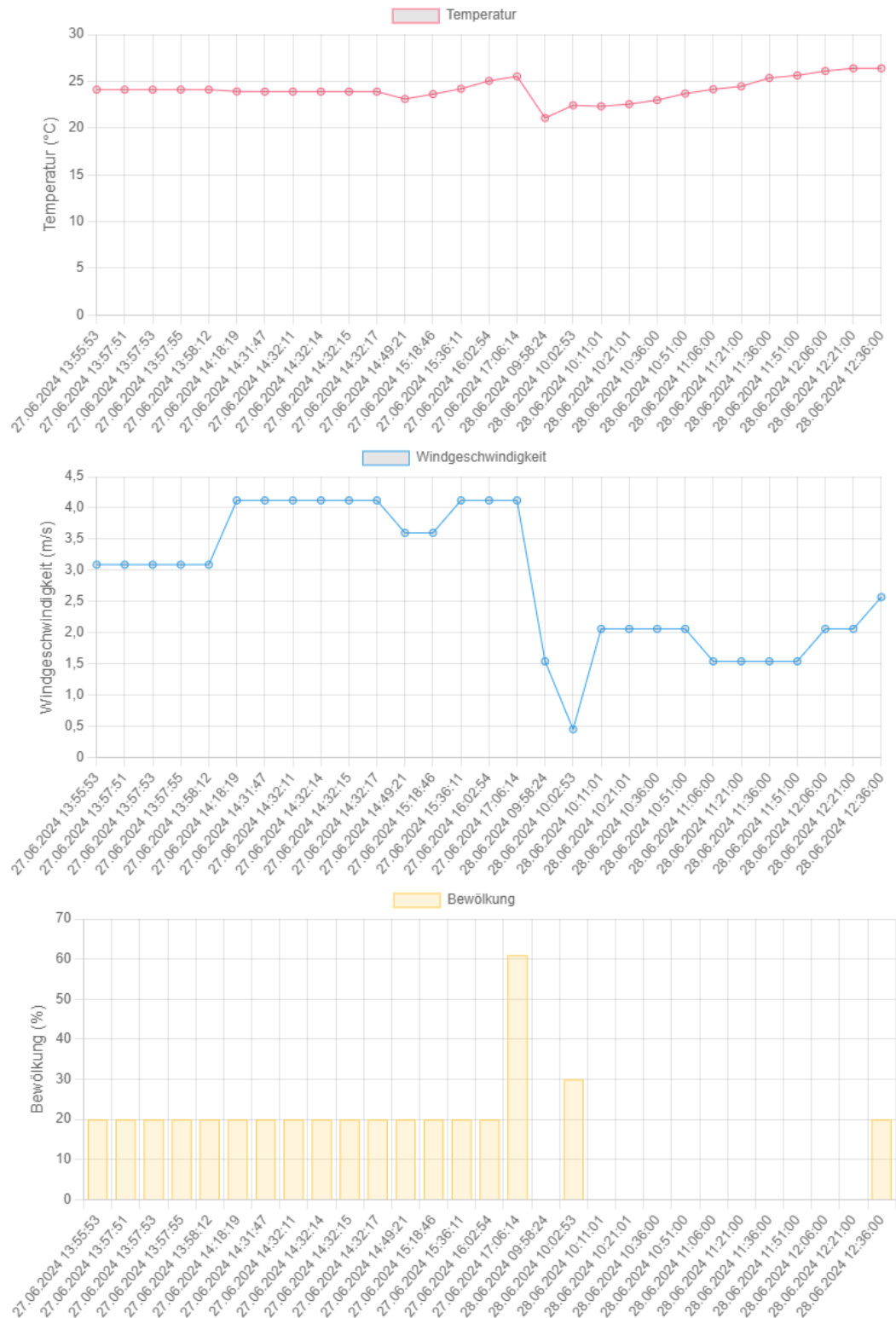
# Daten im gewünschten Format für die POST-Anfrage vorbereiten
weather_data = {
    "cityName": city_name,
    "temperature": temperature,
    "feelsLikeTemp": feels_like_temp,
    "tempMin": temp_min,
    "tempMax": temp_max,
    "weatherDescription": weather_description.capitalize(), # Erste Buchstabe groß schreiben
    "windSpeed": wind_speed,
    "cloudiness": cloudiness
}

# Ziel-URL für die POST-Anfrage definieren
post_url = "http://localhost:8080/weather"

# POST-Anfrage senden
try:
    response = requests.post(post_url, json=weather_data)
    if response.status_code == 200:
        print("Daten erfolgreich an die lokale URL gesendet.")
    else:
        print(f"Fehler beim Senden der Daten. Status Code: {response.status_code}")
except requests.exceptions.RequestException as e:
    print(f"Fehler bei der Anfrage: {e}")
```

Homepage

Wetter in Tett nang



Beschreibung Homepage

Der Code kann in Github eingesehen werden: [kibeleol/Wetterwebsite \(github.com\)](https://github.com/kibeleol/Wetterwebsite)

Die Homepage ist darauf ausgelegt, das aktuelle Wetter für Tettang anhand von interaktiven Diagrammen darzustellen. Hier ist eine Zusammenfassung dessen, was die Homepage tut:

1. DOM-Eventlistener: Sobald das DOM vollständig geladen ist (DOMContentLoaded), wird die Funktion fetchDataAndDrawCharts() aufgerufen, um die Wetterdaten abzurufen und die Diagramme zu zeichnen.
2. Datenabruf und Verarbeitung:
 - Die Funktion fetchDataAndDrawCharts() ruft über die fetch()-API die Wetterdaten von `http://localhost:8080/weather` ab.
 - Die erhaltenen Daten werden in JSON-Format umgewandelt und anschließend gefiltert, um nur Daten für Tettang zu extrahieren.
3. Diagrammerstellung:
 - Basierend auf den gefilterten Daten werden drei Arten von Diagrammen erstellt:
 - Temperaturdiagramm: Zeigt die Temperatur über einen bestimmten Zeitraum an.
 - Windgeschwindigkeitsdiagramm: Zeigt die Windgeschwindigkeit über den gleichen Zeitraum an.
 - Bewölkungsdiagramm: Zeigt die Bewölkung in Prozent über den gleichen Zeitraum an.
4. Diagrammoptionen:
 - Jedes Diagramm ist interaktiv und bietet Tooltips mit detaillierten Informationen beim Überfahren mit der Maus.
 - Die Y-Achsen der Diagramme sind beschriftet, um die Einheiten der jeweiligen Messgrößen (Temperatur in °C, Windgeschwindigkeit in m/s, Bewölkung in %) anzuzeigen.
5. Fehlerbehandlung:
 - Bei einem Fehler während des Datenabrufs oder der Verarbeitung wird eine Fehlermeldung angezeigt, die besagt, dass die Daten nicht erfolgreich abgerufen werden konnten.
6. Styling und Layout:
 - Das Layout der Homepage ist responsiv und nutzt ein Container-Modell mit einem maximalen Inhalt von 800px Breite.
 - Die Seite verwendet eine klare und leserliche Schriftart (Arial, sans-serif) mit einem hellen Hintergrund (#f0f0f0).