

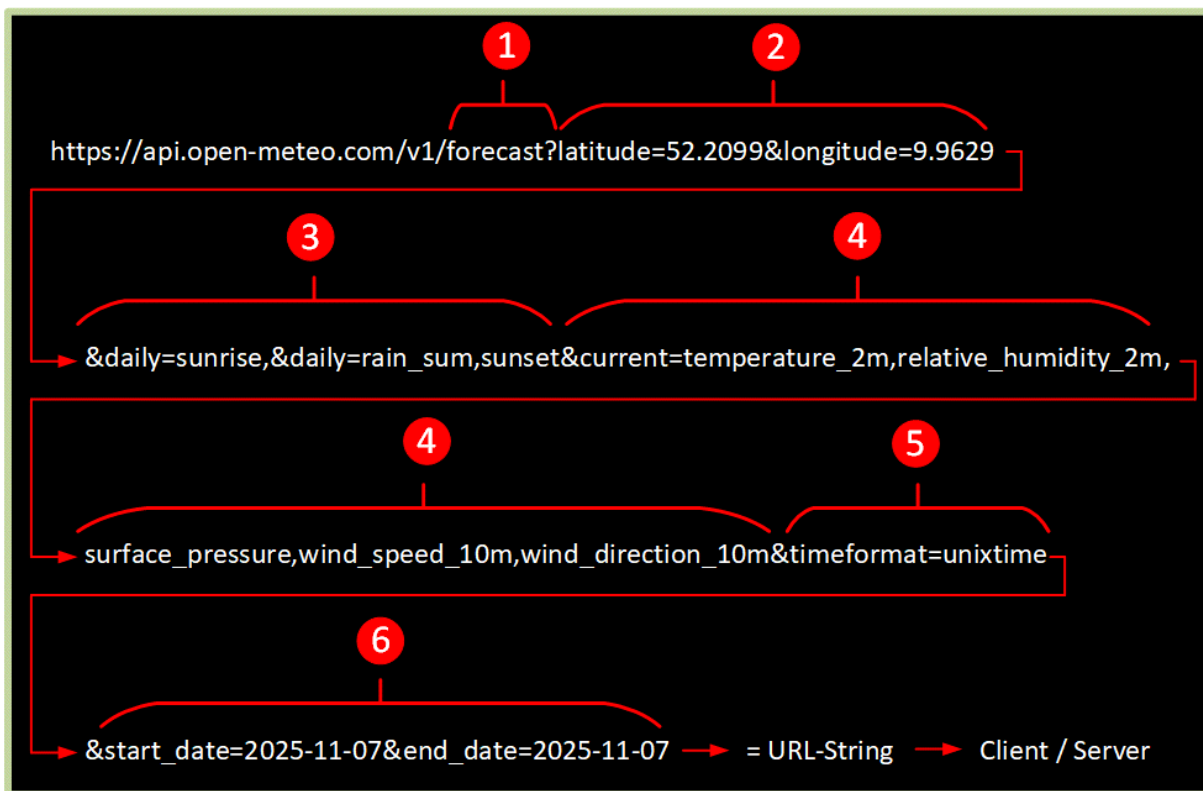
Wetter.

Diese JavaScript-Applikation ist mit dem Wetter-Server OpenMeteo.com verheiratet. OpenMeteo.com stellt eine kostenlose API (Application Programming Interface) zur Verfügung. Weiterhin ist keine vorherige Registrierung erforderlich. Anwender, die diesen Dienst nutzen müssen sich strikt an die Bedingungen von OpenMeteo.com halten, ansonsten wird die IP-Adresse gesperrt.

Diese Beschreibung behandelt zwei Dinge. 1) Eine kurze Einleitung der Bedienung der Software, verbunden mit ein paar Erfahrungen und Hinweise. 2) Die Besonderheit der Datenstruktur und die Kommunikation mit dem Wetter Server von OpenMeteo.com.

Nach dem Hochfahren der JavaScript-Applikation erwartet die Software von dir zwei Eingaben und eine Start-Button Aktivierung. Zuerst musst du für den Ort, dessen Wetterinformationen du erfahren willst, mit zwei Punkten - den Breitengrad (Latitude) und Längengrad (Longitude), selektieren. Bei mir sind das Latitude 52.210335 und Longitude 10.007097 für Borsum bei Hildesheim. Mit Betätigung der Starttaste wird die grafische Webseite aufgefrischt. Im Fehlerfall erscheint eine Textmeldung unterhalb des JavaScript Canvas Bereiches. Unterhalb des Textbereiches wird eine Tabelle angezeigt, die Informationen über Datum & Zeit, Sonnenaufgang und Sonnenuntergang beinhaltet. Ich bitte dich eindringlich um einen besonnenen Umgang mit den Server-Anfragen. Setze hier kein „Funkfeuer“ ab, so dass, das ganze System blockiert wird! In der Vergangenheit habe ich einige Anwendungen, besonders für meine Wetterstation „Hirschgeweih“ mit der Programmiersprache Python geschrieben. Dienstleister wie „Dark Sky Weather“, „Wunderground Weather“ und jetzt auch „OpenWeatherMap“ haben ihre kostenlose Wetter-API Bereitstellung eingestellt. Meine Software funktioniert nicht mehr, es hat viel Arbeit gekostet. Lange Zeit hatte ich die Schnauze voll! In den Foren heißt es schnell: „Erstmal mit der „Wurst“ locken und dann ab in den Abo-Dschungel“! Ich sehe das etwas anders! Millionen von Menschen halten sich an die Regeln, nur eine Handvoll Personen veranstalten hier Böses und machen alles kaputt. Das ist so! Das war so! Es wird so bleiben, der Mensch!

Nach dieser Einleitung, verbunden mit einer persönlichen Meinung, geht es nun weiter zu den Besonderheiten von OpenMeteo.com.



Der Client setzt eine Nachricht zum Server ab. Diese Nachricht besteht aus einer URL, diese startet mit einer Zieladresse (der Server) gefolgt von vielen Anweisungen, die Wetterparameter beinhalten. Dieses Verfahren ist eine Besonderheit von OpenMeteo.com. Ich habe das vorher noch nie gesehen. Man kann sich so sein eigenes Menü zusammenstellen. Die Webseite von OpenMeteo.com erklärt diese Verfahren bis ins kleinste Detail. Informiere dich. Zu Position **-1-**: Gewünscht ist eine Wettervorhersage, für den Ort, der an Position **-2-** spezifiziert ist (Latitude / Longitude). Zu Position **-3-**: Sonnenaufgang und Sonnenuntergang soll mit dabei sein. Von der Vielzahl der Parameter benötige ich nur die, die unter Position **-4-** aufgelistet sind. Man kann verschieden Zeitformate auswählen, ich habe mich für das UNIX-Format entschieden, Position **-5-**. Zum Schluss noch der Zeitstempel, hier für einen Tag, Position **-6-**.

Auf der nächsten Seite zeige ich dir den JavaScript-Algorithmus, für das Absetzen der obigen -sehr langen- URL.

```

304 //*****
305 // Wetterdaten vom OpenWeatherMap - Server lesen
306 //*****
307 async function getWeatherData() {
308     let s;
309     document.getElementById("info").innerText = " ";
310     app.latitude = document.getElementById("lat").value;
311     app.longitude = document.getElementById("lon").value;
312     getDateToday();
313     app.url = `https://api.open-meteo.com/v1/forecast?latitude=${app.latitude}
314     &longitude=${app.longitude}&daily=sunrise,&daily=rain_sum,sunset
315     &current=temperature_2m,relative_humidity_2m,surface_pressure,wind_speed_10m,
316     wind_direction_10m&timeformat=unixtime
317     &start_date=${app.DateToday}&end_date=${app.DateToday}`;
318     try {
319         const response = await fetch(app.url);
320         if (!response.ok) {
321             throw new Error(`HTTP-Fehler! Status: ${response.status}`);
322         }
323         app.weatherData = await response.json();
324         showWeather();
325         s = "Erfolgreiche Netzwerkanfrage zum Server: open-meteo.com!";
326         document.getElementById("info").innerText = s;
327     } catch (error) {
328         let s1 = `${error.message}`;
329         s = "Bei der Netzwerkanfrage zum Server open-meteo.com ist etwas ";
330         s = s + "schiefgelaufen! Die Response / Promise Einrichtung meldet ";
331         s = s + "folgendes: " + s1;
332         document.getElementById("info").innerText = s;
333     }
334 }

```

Achtung! Zeile 313 bis 316! Der URL-String ist hier auf fünf Zeilen verteilt! Das ist nur für die Übersichtlichkeit dieser Beschreibung so gemacht. In der Realität darf das nur eine Zeile sein!!

In den Zeilen 307 bis 334 erfolgt die Kommunikation mit dem Server! Zeile 310 und 311: Die vom Anwender eingegebenen Koordinaten (Breitengrad & Längengrad = Ort) werden in eine Variable gepackt. JavaScript verfügt über einen Mechanismus um einen URL-String mit Variablen zu steuern, beeinflussen. Das ist die Syntax `${app.latitude}`; `${app.longitude}` und `${app.DateToday}`. Der Inhalt der Variable „`app.DateToday`“ wird von der Funktion in Zeile 312 bereitgestellt. In Zeile 319 wird der Krieg gewonnen oder verloren. Informiere dich über die JavaScript Funktion „`fetch()`“. Das ist der General in meiner Wetter-Software.

Auf der nächsten Seite zeige ich dir die Antwort des Servers von OpenMeteo.com.

```
const weatherJsonString = `
{
  "latitude": 52.2,
  "longitude": 9.959999,
  "generationtime_ms": 0.180721282958984,
  "utc_offset_seconds": 0,
  "timezone": "GMT",
  "timezone_abbreviation": "GMT",
  "elevation": 80,
  "current_units": {
    "time": "unixtime",
    "interval": "seconds",
    "temperature_2m": "°C",
    "relative_humidity_2m": "%",
    "surface_pressure": "hPa",
    "wind_speed_10m": "km/h",
    "wind_direction_10m": "°"
  },
  "current": {
    "time": 1762533900,
    "interval": 900,
    "temperature_2m": 6.5,
    "relative_humidity_2m": 96,
    "surface_pressure": 1002.5,
    "wind_speed_10m": 8.7,
    "wind_direction_10m": 163
  },
  "daily_units": {
    "time": "unixtime",
    "sunrise": "unixtime",
    "rain_sum": "mm",
    "sunset": "unixtime"
  },
  "daily": {
    "time": [1762473600],
    "sunrise": [1762496822],
    "rain_sum": [0],
    "sunset": [1762530044]
  }
}
`;
```

War die Kommunikation zum Open Meteo.com-Server eine Punktlandung, reagiert der Server mit dem Senden der Wetterparameter in Form eines JSON-Formats (JavaScript Objekt Notation). JSON und JavaScript verstehen sich bestens. Angenommen du willst die Windgeschwindigkeit selektieren (Zeile 25 von oben) dann sieht die Syntax folgendermaßen aus:

```
app.WeatherData = JSON.parse(
  weatherJsonString
)
Windgeschwindigkeit =
app.WeatherData.current.wind_speed_10m
```

In der Variablen „Windgeschwindigkeit“ steht dann der Wert „8.7“

In dem JSON-String sind nicht nur die Wetterparameter gespeichert, sondern auch wichtige Informationen um eine ansprechende Webseite zu bauen. Du wirst mit Einheiten und Zeitstempel zusätzlich versorgt.

Mir geht es ja nur darum, dir eine Hilfe zu geben damit du dir eine eigene Webseite bauen kannst. Keine Frage! Meine Applikation ist da sehr minimalistisch.

Wenn du dir eine eigene Wetter-Webseite bauen willst, dann ist es ratsam vorher so einen String, wie oben gezeigt, zu erstellen. Das geschieht, wenn man die URL direkt in den Browser eingibt und das JSON-Resultat in eine Variable kopiert. Warum? Während der Software-Entwicklung müsste man sehr viele Anfragen zum Server absetzen und das ist nicht im Sinne des API-Anbieters. Unterschätze nie, wie oft man sich bei abstrakten Daten vertippt.

Du willst meine Software, HTML, CSS und JavaScript, kopieren? Kein Problem. Jedoch kann ich hier nur die Vorgehensweise für Google Chrome beschreiben. Einen anderen Browser benutze ich nicht! Rufe die Wetter-Applikation im Google Chrome Browser auf. Betätige die Taste „F12“. Es wird ein zusätzliches Fenster „DevTool 143“ geöffnet. Dieses Werkzeug dient auch zum Debuggen der JavaScript Datei. Mit diesem Tool bist du in der Lage den Quellcode zu kopieren. **HTML:** Betätige den Reiter „Quellcode“ selektiere die HTML-Datei, markieren und kopieren. **CSS:** Betätige den Reiter „Quellcode“ selektiere die CSS-Datei, markieren und kopieren. **.JS:** Betätige den Reiter „Quellcode“ selektiere die .JS-Datei, markieren und kopieren.

Die in der Applikation visualisierte Grafik, wie Barometer, Thermometer, Hygrometer, Windrichtung, Regenmesser und Windgeschwindigkeit, wurden im Jahr 2013 von mir gekauft. Sie sind bereits Bestandteil meiner lokalen Wetterstation „Hirschgeweih“. Lieferant: „Digital River GmbH, Jürgen Schaetzke“.

November 2025, Hans Busche