

Dashboard

Zwei JavaScript-Applikationen, „Geocoding“ und „Wetter“, gibt es bereits auf der Webseite „apisur.neocities.org“. Dashboard vereinigt beide Programme zu einer Anwendung.

Dashboard greift auf externes JavaScript zu. Weiterhin werden zwei API-Server kontaktiert. Die Bezeichnung „externes JavaScript“ ist nicht ganz richtig, denn diese Skripte werden während des Hochfahrens der Software nicht von einem externen Server eingebunden, sondern liegen als virengeprüfte Datei im apisur.neocities.org-Archiv.

Hier nun eine Auflistung der Bibliotheken und APIs:

- <https://canvas-gauges.com> ; JavaScript-Bibliothek für analoge Messinstrumente. Entwickelt von Mykhailo Stadnyk.
- <https://www.atomek.de/clock/index.html> ; JavaScript-Bibliothek für eine analoge Uhr. Entwickelt von Jan Schirrmacher.
- <https://leafletjs.com> ; JavaScript & CSS-Bibliothek für das Generieren einer Landkarte von OpenStreetMap. Entwickelt von Volodymyr Agafonkin.
- <https://nominatim.org> ; API (Application Programming Interface / Programmierschnittstelle) zur Auflösung einer Adresse in Längen- und Breitengrad und umgekehrt. Dieser Dienst fordert eine Identifizierung in Form einer E-Mail-Adresse. Weiterhin gibt es Regeln für die Anzahl der Aufrufe pro Zeiteinheit.
- <https://open-meteo.com/> ; API (Application Programming Interface / Programmierschnittstelle) zur Bereitstellung von Wetterdaten, im JSON-Format, in Abhängigkeit der Längen- Breitengrad Koordinaten. Auch hier gibt es Regeln für die Anzahl der Aufrufe pro Zeiteinheit.

Wie funktioniert das Programm? Es gibt zwei Möglichkeiten für die Bestimmung der Wetterparameter in Verbindung mit einer Landkarte. 1) Suchen in Abhängigkeit einer Adresse. 2) Suchen in Abhängigkeit der Längen- und Breitengrad Koordinaten. Zwei Radio-Button bestimmen die Auswahl. Mit Betätigung der Starttaste läuft folgender Prozess ab: Adresse, Breiten- und Längengrad werden vom HTML-Script zum JavaScript transferiert [[document.getElementById\("id"\)](#)]. JavaScript erstellt daraus einen URL-String

und kontaktiert den nominatim-Server [**fetch(url)**]. Der Server liefert eine JSON-Antwort, die alle Informationen für das Erstellen einer Landkarte enthält, zusätzlich Längen- und Breitengrad-Koordinaten. Diese Koordinaten werden in einen zweiten URL-String eingebunden und der open-meteo-Server wird damit gefüttert [erneut → **fetch(url)**]. Dieser Server liefert auch eine JSON-Antwort, die alle Wetterdaten beinhaltet, die für dieses Programm notwendig sind. Jetzt, wo alle Daten im „Kasten“ sind, kommt eine mächtige externe JavaScript-Bibliothek [**leaflet**] ins Spiel. Diese Bibliothek generiert zusammen mit OpenStreetMap eine Landkarte mit allem Pipapo. Jetzt fehlt nur noch die „Rückfahrkarte“ von JavaScript zur HTML-Seite! Acht Visualisierungs-Rechtecke [**canvas**] der HTML-Seite erwarten ein Ergebnis. Das übernimmt eine weitere externe Bibliothek mit dem Namen „**canvas-gauges**“.

Zum Schluss komme ich nochmals auf die externen Bibliotheken zurück. Im Internet gibt es ja viele Anbieter! Zwei Merkmale sind für mich entscheidend! 1) Gibt es eine gute Dokumentation mit Beispielen. 2) Verfügt die Bibliothek über eine große Community und ist sie im GitHub & Stack Overflow vertreten.

November 2025, Hans Busche