

Geocoding, OpenStreetMap

Viele Online-Händler präsentieren auf ihrer Webseite eine Landkarte, die man mit der Maus vergrößern oder verkleinern kann (Zoom, heranzoomen). Was steckt dahinter? Wie funktioniert das? Diese Applikation behandelt den Prozess, wie man eine Adresse oder ein Ortsname in geografische Koordinaten (Breiten- und Längengrad) umgewandelt, um einen bestimmten Punkt auf der Erdoberfläche zu identifizieren. Auch der umgekehrte Weg, geografische Koordinaten in eine Adresse wandeln, wird hier behandelt. Die Webseite „Geocoding“ unterliegt der nicht kommerziellen Nutzung. Der Anwender nutzt diese Software im privaten Bereich.

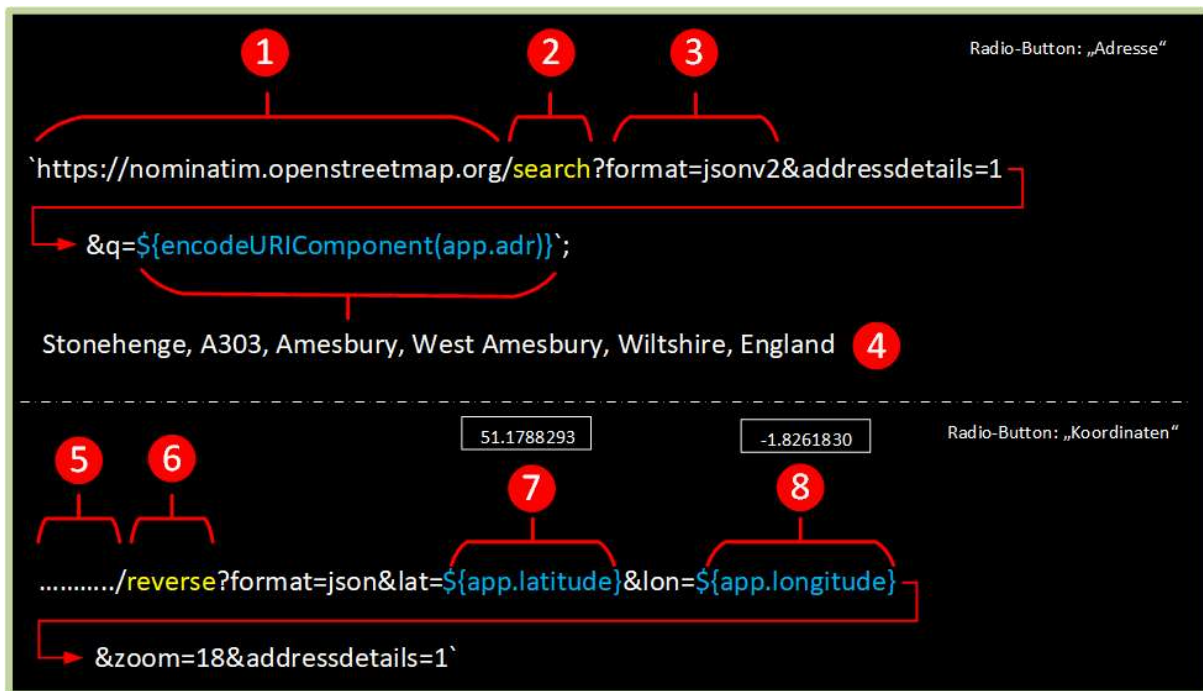
Dreh- und Angelpunkt dieser Applikation ist der Zugriff auf eine Datenbank von OpenStreetMap, die Geodaten bereitstellt. Details der Software, siehe weiter unten in dieser Beschreibung. OpenStreetMap (OSM) wird von freiwilligen Personen gepflegt. Die Daten können kostenlos heruntergeladen und für verschiedene Zwecke, z. B. in Apps und auf Webseiten, verwendet werden. Natürlich gibt es Regeln, die beachtet werden müssen.

Bedienung der Geocoding-Applikation: Für die Bereitstellung einer Landkarte gibt es zwei Möglichkeiten. Die Markierung des Radio-Buttons ist auf „Eingabe Adresse“ gesetzt, das ist die Standardeinstellung, dann erwartet die Applikation von dir eine Adresse. Oder durch Mausklick wird „Radio-Button-Eingabe geografische Koordinaten“ ausgewählt, dann sind Breiten- und Längengrad die Vorgaben für die Applikation. Weiterhin wird eine E-Mail-Adresse erwartet. Mit Betätigung des Schalters „Suche starten“ wird die Landkarte angezeigt. Unterhalb der Karte ist ein schwarzes Informationsfeld. Dort wird bei erfolgreicher Server-Kontaktierung ein Antwortsatz generiert. Kann OpenStreetMap die Adresse nicht auflösen erfolgt die Meldung „Keine Antwort erhalten“. In einer Konfliktsituation zwischen Client und Server wird eine Fehlermeldung angezeigt.

Für die Eingabe der Adresse gibt es nur einen einzeiligen Editor. Es können mehrere Suchbegriffe eingegeben werden, diese müssen durch ein Komma getrennt werden. Beispiel: **Straße und Hausnummer**, **Postleitzahl und Stadt**.

E-Mail-Adresse wird erwartet! Der Server, hier <https://nominatim.org/>, schreibt das in seiner „**Usage Policy**“ vor!!

Die Säulen dieser Applikation sind: 1) Nominatim API ("*Open Source Geocoding With OpenStreetMap Data*"). 2) Eine externe JavaScript Bibliothek (leaflet.js). Diese Bibliothek „leaflet“ wird nicht während des Hochlaufs der Software durch von extern Drittanbieter eingelinkt, sondern liegt, Vieren geprüft, im Ordner des Providers „neocities.org“. Das sind zwar 150 KByte Speicherplatz mehr, aber ich bevorzuge diesen Weg.



Die erste Säule, Nominatim, eine API.

Ist das Radio-Button-Element „Adresse“ ausgewählt, dann wird deine eingegebene Adresse in die Variable „app.adr“ gespeichert, Markierung 4. Radio-Button „Adresse“ sorgt auch dafür, dass die erste URL (Uniform Resource Locator) mit der Kennung „/search“, Markierung 2, zum Nominatim Server gesendet wird. Wählst du Radio-Button-Element „Koordinaten“ aus, dann werden deine Eingaben für Latitude und Longitude in die zwei Variablen „app.latitude & app.longitude“ hinterlegt, Markierung 7 und 8. Weiterhin wird dann die URL mit der Kennung „/reverse“, Markierung 6, zum Server geschickt. Man kann nicht einfach die Zeichenkette (String) für Adresse oder Koordinaten in die URL einpflanzen, man muss dafür die blau markierte Syntax verwenden, siehe Markierung 4 & 7 & 8. Schaust du dir den tatsächlichen generierten URL-String im Debugger an, dann wird dir schwindelig! Wir haken das jetzt ab, ist einfach so! Der URL-Vorspann Markierung 1 und 5 sind identisch und beinhalten das Ziel: Nominatim und OpenStreetMap. Wir erwarten vom Server eine Antwort, diese haben wir als JSON-Notation festgelegt, Markierung 3.

```

17 //*****
18 // Wenn mit Nominatim alle geklappt hat, dann von openstreetmap die Karte holen
19 //*****
20 function get_OSM_map() {
21     const mapContainer = document.getElementById("map");
22     if (app.mymap !== null && mapContainer._leaflet_id) {
23         app.mymap.remove();
24         app.mymap = null;
25     }
26     var s = "<b>" + app.ort.display_name + "<br>";
27     s = s + "Breitengrad: " + app.ort.lat + "<br>";
28     s = s + "Längengrad: " + app.ort.lon + "</b><br>";
29     app.mymap = L.map("map").setView([app.ort.lat, app.ort.lon], app.initialZoom);
30     L.tileLayer("https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png", {
31         attribution:
32         'Map data &copy; <a href="https://www.openstreetmap.org/copyright">OpenStreetMap</a> contributors',
33     }).addTo(app.mymap);
34
35     L.marker([app.ort.lat, app.ort.lon], { icon: app.blueIcon })
36     .addTo(app.mymap)
37     .bindPopup(s)
38     .openPopup();
39 }

```

Die zweite Säule, Leaflet, eine mächtige JavaScript Bibliothek und CSS.

Die Zeilen 29 bis 38 sind Anweisungen aus der Dokumentation von <https://leafletjs.com/>. Hat der Server Nominatim die Antwort geliefert erfolgt der Aufbau der Landkarte. Hier schreibt OpenStreetMap und Leaflet eine Attribution „**Usage Policy**“ vor! Dass passiert in den Zeilen 31 bis 33. In den Zeilen 26 bis 28 wird die Antwort von Nominatim in einen String verpackt und in der Zeile 37 visualisiert. In Zeile 35 wird auf der Landkarte ein blauer Pfeil (Pin) angezeigt. Bei der Entwicklung dieser Software stellte sich heraus, dass man immer nur einmal eine Karte zur Anzeige bringen konnte. Ein erneuter Klick auf den Button „Start“ erzeugte eine Fehlermeldung! Abhilfe schaffen hier die Zeilen 21 bis 25.

Im apicur.neocities.org-Archiv gibt es eine zweite JavaScript-Applikation mit der Bezeichnung „Wetter“. Sie ist mit der Geocoding-App vergleichbar und nutzt ebenfalls eine API (Application Programming Interface). Ein sehr schönes und interessantes Projekt wäre die Zusammenführung der beiden Programme. Wenn du Lust hast!? Wie man im Browser (Google Chrome) die Skripte HTML, CSS und JavaScript kopieren kann wird in der „Wetter-App“ erklärt (PDF).

November 2025, Hans Busche