

Leopold-Franzens-Universität Innsbruck

Institut für Geographie

# **Abschlussprojekt:**

## **Wien: Freizeitaktivitäten im Sommer wie im Winter**

Lehrveranstaltung: **716409 VU/3 Geoinformatik: Web mapping**  
Sommersemester 2019

Dozenten: Klaus Förster  
Bernd Oegg

Bearbeiter: Veit Badde (Matr. Nr.11841570)  
Andreas Summer (Matr. Nr. 01441912)  
Nadine Pook (Matr. Nr. 11805752)

Webadresse <https://hacknbike.github.io/>

Innsbruck, den 18.06.2019

# Inhaltsverzeichnis

1	Konzept.....	2
2	Änderung des Konzepts .....	3
3	Seitenaufbau und Styling.....	4
3.1	Startseite: Seitenaufbau & Stylesheet .....	<b>Fehler! Textmarke nicht definiert.</b>
4	Leaflet Karte.....	4
5	Karten Plug-Ins .....	11
5.1	Leaflet.markercluster.....	11
5.2	Leaflet Control Search.....	12
5.3	Leaflet.Sun.....	13
5.4	Leaflet.fullscreen .....	14
5.5	Leaflet Funktion Control.Scale .....	15
6	Literaturverzeichnis .....	15

## 1 Konzept

Unter dem Titel „**Wien: Freizeitaktivitäten im Sommer wie im Winter**“ möchten wir Einheimischen und Besuchern der Stadt Wien eine Orientierungsplattform für Freizeitaktivitäten bieten. Die Internetseite soll Interessierte dabei unterstützen, sich für eine Aktivität zu entscheiden bzw. unterschiedliche miteinander zu verknüpfen. Um dieses Ziel zu erreichen, haben wir eine Übersichtsseite erstellt, die in das Thema einführt und die Links zu den Winter- bzw. Sommeraktivitäten enthält. Zusätzlich werden weitere Links geliefert, die ebenfalls nützlich zur Orientierung sein können.

Auf der Winter- bzw. Sommerseite findet sich jeweils eine Karte, die einige mögliche Freizeitbeschäftigungen in und um Wien bereithält. Im Sommer sind dies Themenrad- und Stadtwanderwege, Badestellen, Schwimmbäder sowie Burgen und Schlösser. Im Winter

wiederum werden die lokalen Weihnachtsmärkte, ein Silvesterpfad und Sehenswürdigkeiten, sowie Museen als Indoormöglichkeiten, zum gemütlichen Aufwärmen nach einem kalten Tag in der Stadt, angezeigt. Diese können anhand einer Layercontrol beliebig angezeigt bzw. ausgeblendet werden und teilweise auch durch die Suchfunktion gefunden werden (mehr zu den Plugins in Kapitel 4). Die Aktivitäten und Besucherhighlights können angeklickt werden, was weitere Informationen zu den einzelnen Punkten erscheinen lässt, teilweise sind diese Informationen Beschreibungstexte, teilweise aber auch Links zu den zugehörigen Websites.

Zusätzliche Features sind Verweise auf die anderen beiden Seiten des Projekts (Home und Winter/Sommer), die Fullscreenfunktion für die implementierten Karten (bei denen der Kartenlayer ebenfalls über Layercontrol geändert werden kann), die Maßstabsleiste in der linken unteren Ecke und die Einteilung der Aktivitäten in Cluster, damit die Übersichtlichkeit auf der Karte erhalten bleibt. Als weitere Organisationshilfe bieten wir die Möglichkeit, mit dem Klick auf eine beliebige Stelle auf der Karte, den Zeitpunkt des Sonnenuntergangs angezeigt zu bekommen, um die Aktivitäten auch zeitlich sinnvoll planen zu können und nicht von der Dunkelheit überrascht zu werden.

Die benötigten Daten für diese Seiten erhalten wir im JSON-Format von der Seite des Open Source Österreich, auf der etliche Datensätze für Österreich bereitgestellt werden. Alles in allem stellen wir also drei HTML-Seiten zur Verfügung:

Eine Übersichtseite zu dem Thema,

eine Seite zu den Sommeraktivitäten sowie

eine Seite zu den Winteraktivitäten

Das Ganze dient dazu, eine Orientierungshilfe zu leisten und die Planungen für Freizeitaktivitäten in und um Wien ein wenig zu erleichtern.

## **2 Änderung des Konzepts**

In unserem ursprünglichen Konzept haben wir noch die Mountainbike-Trails im Salzburger Land als Datengrundlage gewählt. Wir wollten dabei, ähnlich wie in Kapitel 1 beschrieben, eine Übersichtsseite erstellen und von dort ausgehend, drei weitere HTML-Seiten verlinken. Diese sollten die Mountainbike-Trails in ihre unterschiedliche Schwierigkeitsstufen aufteilen und jeweils auf einer Karte darstellen.

Nach einigen misslungenen Versuchen die Trails auf unseren Karten anzeigen zu lassen und daraus folgenden Überlegungen, haben wir uns dazu entschieden von diesem Konzept Abstand zu nehmen und stattdessen mit den Daten der Stadt Wien zu arbeiten. Dies hatte

zum einen den Grund, dass die, vom Land Salzburg bereitgestellten Daten, ein schwer zu bearbeitendes Format besaßen, da hierbei jeder Trail als einzelne JSON-Datei dargestellt wird. Dieses Problem kann zwar durch verschiedene Maßnahmen umgangen werden (z.B. durch das importieren in GIS-Anwendungen und des anschließenden Exports als JSON-Datei), allerdings haben wir dabei weitere Schwierigkeiten erkannt und viel Zeit darin investiert. Zum anderen bietet die Stadt Wien eine breitere und verhältnismäßig einheitliche Datengrundlage an, was das Arbeiten mit diesen Daten erheblich vereinfacht und diese somit benutzerfreundlicher bedienen lässt. Schlussendlich haben wir uns somit zu dem Schritt entschieden, ein neues Konzept zu bearbeiten und damit versucht ein schlüssiges, interessantes und umsetzbares Gesamtpaket zu verwirklichen.

### 3 Seitenaufbau und Styling

#### Startseite: Seitenaufbau & Stylesheet

Die Übersichtsseite bietet einen Einblick in das Thema und beschreibt kurz Wien im Sommer und im Winter. Sie besteht aus einer *index.html* und einer *main.css* -Datei. Die *index.html* beschreibt die Hauptseite und wird über `<!DOCTYPE html>` `<html>` definiert. Im `<head>` (...) `</head>` sind ist zum einen die Verbindung zu unserem Stylesheet, dem *main.css* definiert, zum anderen die Links zu unseren Quellen für die Schriftarten ([fonts.googleapis.com](https://fonts.googleapis.com)) und Icons ([use.fontawesome.com](https://use.fontawesome.com)).

```
1  <!DOCTYPE html>
2  <html>
3
4  <head>
5      <meta charset="utf-8">
6      <meta http-equiv="X-UA-Compatible" content="IE=edge">
7      <title>Wien</title>
8      <meta name="viewport" content="width=device-width, initial-scale=1">
9      <link rel="stylesheet" type="text/css" media="screen" href="main.css">
10     <link href="https://fonts.googleapis.com/css?family=Playfair+Display|Lobster" rel="stylesheet">
11     <link href="https://fonts.googleapis.com/css?family=Open+Sans&display=swap" rel="stylesheet">
12     <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.7.2/css/all.css"
13         integrity="sha384-fnmOCqbTlWIlj8LyTjo7mOUStjsKC4p0Pqbqyi7RrhN7udi9RwhKkMHpvLbHG9Sr" crossorigin="anonymous">
14 </head>
```

*index.html*

Im `<body>` `</body>`, welcher im *index.html* zu finden ist, befinden sich sämtliche Inhalte der Seite. Im *main.css* werden für diesen über `body { ... }` Farbe von Schrift und Hintergrund durch

*color:* und *background-color:*, die Schriftart über *font-family:*, die maximale Breite der eingefügten Bilder mit *max-width:* und die Anpassung der Ränder über *margin:* festgelegt.

```
1  body {
2      max-width: 1180px;
3      margin: auto;
4      /* Ränder passen sich automatisch auf das Format des Bildschirms an */
5      font-family: 'Open Sans', sans-serif;
6      /* Ausgewählte Schriftart aus der freien Quelle fonts.googleapis.com */
7      color: black;
8      /* Schriftfarbe */
9      background-color: white;
10     /* Hintergrundfarbe */
11 }
```

body in main.css

Im `<body>` ist zunächst die Überschrift im `<header>` mit den zwei verschiedenen Überschriftarten `<h3>` und `<h4>` benannt. Im *main.css* sind die Attribute für das Überschriftfeld selbst festgelegt und das Hintergrundbild für die Überschrift eingebunden. Außerdem sind die Attribute für die Schrift hier definiert.

```

14  header {
15      text-align: center;
16      /* Text in der Überschrift ist mittig gesetzt */
17      background-image: url("images/vienna2.jpg");
18      /* Hintergrundbild der Überschrift aus dem Ordner Images */
19      background-position: 0 50%;
20      /*Position des Bildes auf Überschrift angepasst */
21      font-size: 3em;
22      /*Schriftgröße */
23      font-family: 'Open Sans', sans-serif;
24
25      box-shadow: 4px 4px 20px #gray;
26  }
27
28
29  header h3 {
30      font-size: 2em;
31      margin: 0;
32      color: #black;
33      font-family: 'Open Sans', sans-serif;
34  }
35
36  header h4 {
37      font-size: 0.75em;
38      color: #white;
39      font-family: 'Open Sans', sans-serif;
40  }

```

*header im main.css*

Das Bild hinter der Überschrift ist abgespeichert im Ordner „images“ und kann so direkt eingebunden werden. Die ursprüngliche Quelle für das Bild ist die freie Plattform für Bilder *pixabay.com*.

Unter der Überschrift ist jeweils eine *diff*-Klasse für die Autoren *.author*, mit einem direkten Link zu den Email Adressen und für den Link zum Repository *.repo*, definiert. Die Attribute für beide sind wieder im *main.css* zu finden. Die verwendete Schriftart *Open Sans* ist aus der Quelle *fonts.googleapis.com* entnommen.

Durch eine Linie `<hr>` werden die Überschrift und der Abschnitt zu den Autoren und dem Repository vom Hauptteil im `<main>` innerhalb des `<body>` getrennt. Nach einem Einführungsteil zum Thema „Wien“, folgen die Beschreibungen zu „Wien im Sommer“ und „Wien im Winter“. Biede enthalten ein Bild von *pixabay.com*, auf welches wieder über den Ordner *images* zugegriffen wird. Der Name des Fotos beinhaltet außerdem ein Kamera-Icon, welches frei auf *leaflet.com* verfügbar ist. Die Verbindung wird über den dazugehörigen Link im `<header>` hergestellt.

```
<figcaption><em><i class="fas fa-camera"></i> Foto: Wien im <br>Winter</em></figcaption>
 <br>
```

*Foto mit Icon*

Unter den Bildern befindet sich jeweils ein Einführungstext zu „Wien im Sommer“ und „Wien im Winter“ mit den dazu als Link eingebundenen Quellen, einer Aufzählung der Freizeitaktivitäten, welche auf den Karten Abgebildet sind und der eingebundene Link zu den *index.html* „Sommer“ und „Winter“, welche die Übersichtskarten enthalten.

```
<p class="blocksatz">Hier wird der Sommer in Wien noch ein wenig sommerlicher: Theater, Kino und Musik genießen
oder einfach mal in der Sonne entspannen zwischen Grün, Sand und Wasser, bei erfrischenden Drinks im Glas
und Köstlichkeiten auf dem Teller. Wien bietet ein abwechslungsreiches Programm im Sommer
<br><br><em>Quelle: <a href="https://www.stadt-wien.at/freizeit/sommer-in-wien.html">stadt-wien.at</a></em>
<br><br>
<u>Zusätzlich bieten wir euch eine Übersichtskarte mit folgenden Freizeitaktivitäten:</u> <br> Bäder <br>
Themenradwege <br> Wanderwege <br> Burgen und Schlösser
<br><br>

<nav>
  <a class="sommer" href="https://hacknbike.github.io/wien-freizeit/sommer/index.html"><i
  |   class="fas fa-sun"></i> Freizeitaktivitäten im Sommer </a></nav> <br><br><br>
```

*Wien im Sommer – Text und Link*

Neben dem Link zu den Übersichtskarten ist jeweils ein Icon von *fontawesome.com* eingefügt, welches die Jahreszeiten beschreibt.

Zusätzlich findet sich eine Liste mit Links zu nützlichen Seiten in Bezug auf Wien und die genannten Freizeitaktivitäten.

```

<h3>Links</h3>

<ul>
  <li><a href="https://www.wien.info/de">Online Reiseführer</a>
</li>
  <li><a href="https://de.wikipedia.org/wiki/Wien">Wikipedia</a></li>
  <li><a href="https://www.wintermarkt.at/">Wintermarkt</a>
</li>
  <li><a href="https://www.wien.gv.at/freizeit/baeder/">Wiener Bäder</a>
</li>
  <li><a href="https://wienersilvesterpfad.at/">Silvesterpfad</a>
</li>
  <li><a href="https://www.wienmuseum.at/">Museen</a>
</li>
</ul>

```

Links

Abschließend findet man im `<footer>` noch einmal die Buttons „Sommer“ und „Winter“ mit den dazugehörigen Icons um auf die Seiten mit den Übersichtskarten zu gelangen.

```

<footer>
  <!-- links zu Sommer- und Winterkarte mit Icons -->
  <nav>
    <a class="sommer" href="https://hacknbike.github.io/wien-freizeit/sommer/index.html">
      <i class="fas fa-sun"></i> Sommer </a> <br>
    <a class="winter" href="https://hacknbike.github.io/wien-freizeit/winter/index.html">
      <i class="far fa-snowflake"></i> Winter</a>
  </nav>
</footer>

</body>

```

Footer

Im letzten Teil des `main.css` werden mit der `@media`- Regel die Eigenschaften der Seite angepasst auf den Monitor auf dem die Seite angezeigt wird, definiert. So ist es möglich die Seite unverrückt und vollständig auf Smartphone, wie auch auf großen PC-Bildschirmen zu betrachten.



```

@media only screen and (max-width: 900px) {
  main figure figcaption {
    float: none;
    margin-left: 0;
    width: auto;
  }
}

```

*@media-Rule*

## 4 Leaflet Karte

Die Leaflet Kartenapplikation wird mit folgendem Code in die HTML Seite eingebunden:

```

<link rel="stylesheet"
href="https://unpkg.com/leaflet@1.5.1/dist/leaflet.css" />
<script src="https://unpkg.com/leaflet@1.5.1/dist/leaflet.js"></script>

```

Nachfolgender Code ist für die Initialisierung der Karte zuständig. Die Karte in unserer Webseite verfügt über sieben mögliche Kartenlayer, wobei hier nur der erste dargestellt ist.

```

let karte = L.map("map");

const kartenLayer = {
  osm: L.tileLayer("https://{s}.tile.osm.org/{z}/{x}/{y}.png", {
    subdomains: ["a", "b", "c"],
    attribution: 'Map data &copy; <a
href="https://www.openstreetmap.org/">OpenStreetMap</a> contributors, <a
href="https://creativecommons.org/licenses/by-sa/2.0/">CC-BY-SA</a>'
  }),
};

```

Dies Anweisung fügt die Kartenlayer einer LayerControl hinzu:

```

const layerControl = L.control.layers({
  "Geoland Basemap": kartenLayer.geolandbasemap,
  "Geoland Basemap Grau": kartenLayer.bmapgrau,
  "Geoland Basemap High DPI": kartenLayer.bmaphidpi,
  "Geoland Basemap Orthofoto": kartenLayer.bmaporthofoto30cm,
  "Geoland Basemap Gelände": kartenLayer.bmapgelaende,
  "OpenStreetMap": kartenLayer.osm,
  "Stamen Terrain": kartenLayer.stamen_terrain,
}).addTo(karte);

```

Die erste Zeile im folgenden Code-Abschnitt definiert den beim Öffnen der Webseite geladenen Kartenlayer, die zweite setzt das dabei geladene Blickfeld fest.

```
kartenLayer.geolandbasemap.addTo(karte);  
karte.setView([48.208333, 16.373056], 11);
```

Die Daten unserer Webseite liegen im JSON Format vor und werden wie im nachfolgenden Code dargestellt in die Karte eingebunden und mit einem definierten Icon in der Karte Angezeigt. Zusätzlich wird mit dem letzten Absatz auch ein Popup bei einem Klick auf das Icon hinzugefügt.

```
const adv = '  
https://data.wien.gv.at/daten/geo?service=WFS&request=GetFeature&version=1.1.0  
&typeName=ogdwien:ADVENTMARKTOGD&srsName=EPSG:4326&outputFormat=json';  
  
function makeMarker(feature, latlng) {  
    const advIcon = L.icon({  
        iconUrl: 'http://www.data.wien.gv.at/icons/adventmarktogd.png',  
        //anderer Marker  
        iconSize: [16, 16]  
    });  
    const advMarker = L.marker(latlng, {  
        icon: advIcon  
    });  
    advMarker.bindPopup(`  
        <h3>${feature.properties.BEZEICHNUNG}</h3>  
        <p>Adresse: ${feature.properties.ADRESSE}</p>  
        <p>Geöffnet von-bis: ${feature.properties.DATUM}</p>  
        <p>Öffnungszeiten: ${feature.properties.OEFFNUNGSZEIT}</p>  
        <hr>  
        <footer><a target="blank"  
href="${feature.properties.WEBLINK1}">Weblink</a></footer>  
        `); //Name, Beschreibung, Weblink (neuer Tab)  
    return advMarker;  
}
```

## 5 Karten Plug-Ins

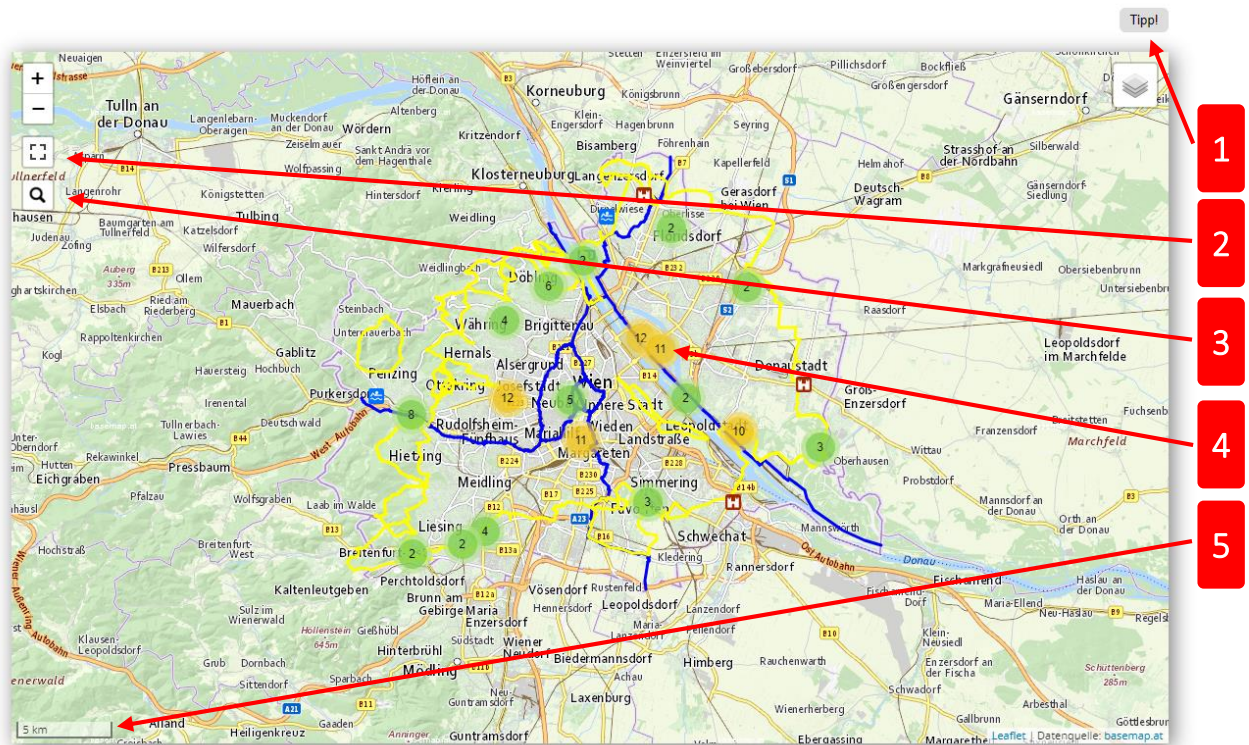


Abbildung 1: Kartendarstellung

### 5.1 Leaflet.markercluster

Github Dokumentation: <https://github.com/Leaflet/Leaflet.markercluster>

Dieses Leaflet Plug-In fasst mehrere Marker, wenn diese sehr nahe bei einander liegen zu einem übersichtlicheren Zahlenhinweis zusammen. Die in einem Kreis dargestellte Zahl entspricht dabei denn zusammengekommen einzelnen Markern. Dieses Tool passt sich dem jeweiligen Zoomlevel des Users an. Nummer 1 in Abbildung 1 illustriert die Funktion dieses Plug-Ins.

Dieses Plug-In wurde von uns mittels eines Content Delivery Networks eingebunden. Der Link dazu lautet wie folgt: <https://unpkg.com/leaflet.markercluster@1.4.1/dist/>

Der Code im Header für die Verknüpfung mit der HTML Seite lautet folgendermaßen:

```
<script
src="https://unpkg.com/leaflet.markercluster@1.4.1/dist/leaflet.markercluster.
js"></script>
```

```

<link rel="stylesheet"
href="https://unpkg.com/leaflet.markercluster@1.4.1/dist/MarkerCluster.css" />
<link rel="stylesheet"
href="https://unpkg.com/leaflet.markercluster@1.4.1/dist/MarkerCluster.Default
.css" />

```

Initialisiert wird das Plug-In in der JavaScript-Datei so:

```

async function loadSights(url) {
  const clusterGruppe = L.markerClusterGroup();
  const response = await fetch(url);
  const sightsData = await response.json();
  const geoJson = L.geoJson(sightsData, {
    pointToLayer: makeMarker
  });

  //Clustergruppe
  clusterGruppe.addLayer(geoJson);
  karte.addLayer(clusterGruppe);
  layerControl.addOverlay(clusterGruppe, "Burgen und Schlösser");
}

```

Im ersten Absatz werden die Daten eine Lade Funktion hinzugefügt und eine Cluster-Gruppe erzeugt. Der zweite Absatz fügt diese dann zur Karte hinzu und erstellt eine Layercontrol für die Daten-Gruppe.

(Github (2019): Marker Clustering plugin for Leaflet)

## 5.2 Leaflet Control Search

Github Dokumentation: <https://github.com/stefanocudini/leaflet-search>

Mit dem Leaflet-Plug-In Control Search kann ein Suchfeld in der Karte erzeugt werden. Damit kann eine Quelle auf Übereinstimmungen untersucht werden. Wird ein Ergebnis gefunden und ausgewählt wird, beinhaltet das Plug-In eine Funktion die automatisch zum gewählten Objekt zoomt und dieses rot einkreist. Nummer 3 in Abbildung 1 zeigt den Zugehörigen Button zu dieser Such-Funktion. Dieses Bezieht sich allerdings immer nur auf eine Datenquelle, da es in den jeweiligen Layer eingebunden wird.

Da es für dieses Plug-In keine CDN gibt müssen die Dateien heruntergeladen und im Projekt gespeichert werden. Dann werden sie wie folgt ins HTML eingebunden:

```

<script src="plugins/leaflet-search.min.js"></script>
<link rel="stylesheet" href="plugins/leaflet-search.min.css" />

```

Der Code für das Suchfeld wird in die Funktion, welche die jeweiligen Daten beinhaltet eingebunden:

```
//Suchfeld einfügen
const suchFeld = new L.Control.Search({
  layer: clusterGruppe,
  propertyName: "NAME",
  zoom: 17,
  initial: false,
});
karte.addControl(suchFeld);
```

Die verwendeten Options stehen dabei für:

- Layer: Daten-Layer welche durchsucht werden soll.
- propertyName: Element im Daten-Layer welches für die Suche herangezogen werden soll.
- zoom: Größe des Zooms mit welchem ein gefundenes Objekt herangefahren wird.
- initial: Einstellung ob Suche mit dem Anfang des Suchelements übereinstimmen muss oder auf Begriffe im Wortinneren erkannt werden sollen.

In der Plug-In Dokumentation sind einige weitere Optionen, welche der Funktion mitgegeben werden könnten, erläutert.

(Github (2019): Search stuff in a Leaflet map)

### 5.3 Leaflet.Sun

Github Dokumentation: <https://github.com/dj0001/Leaflet.Sun>

Das Plug-In Leaflet Sun erlaubt in Zusammenhang mit einem Popup die Anzeige der Sonnenuntergangszeit am jeweiligen Ort in der Karte. Dies ist in nachfolgender Abbildung dargestellt.

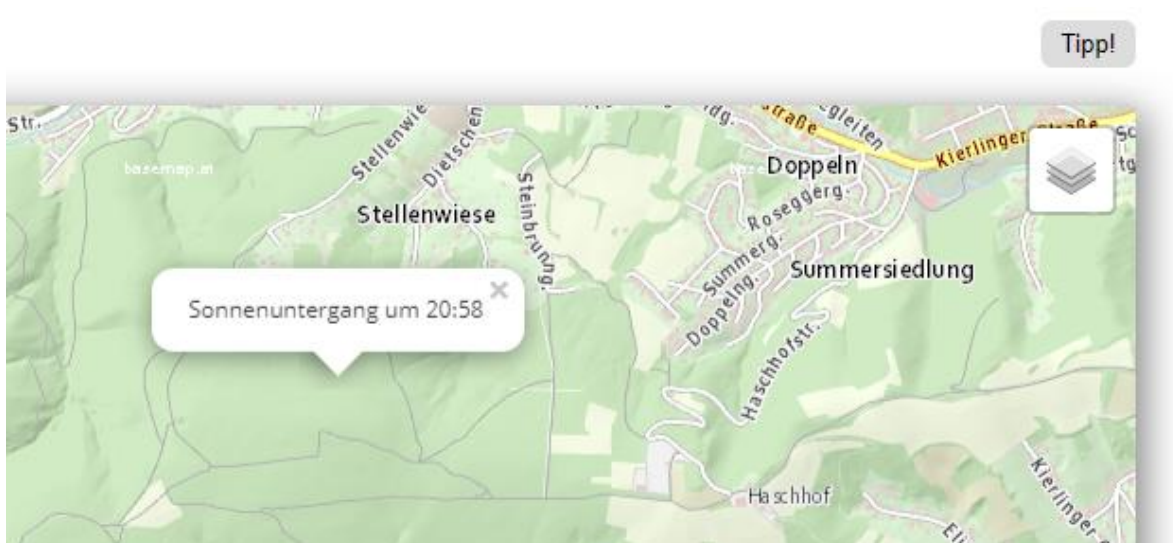


Abbildung 1: Plug-In Leaflet.Sun



Da es für dieses Plug-In keine CDN gibt müssen die Dateien heruntergeladen und im Projekt gespeichert werden. Dann werden sie wie folgt ins HTML eingebunden:

```
<script src="plugins\L.Sun.js"></script>
```

Der Code für den Popup mit der Funktion in JavaScript lautet wie folgt:

```
var popup = L.popup();

function onMapClick(e) {
    popup
        .setLatLng(e.latlng)
        //.setContent("You clicked the map at " + e.latlng.toString())
        //this.getLatLng() don't work
        .setContent("Sonnenuntergang um " + L.sun.sunset(e.latlng))
        //this.getLatLng() don't work
        .openOn(karte);
}
karte.on('click', onMapClick);
```

Um auf die Funktion hinzuweisen wurde zusätzlich ein Button mit einem Hinweis darauf rechts über der Karte hinzugefügt. Der HTML Code dafür lautet:

```
<button onclick="Tipp()" style="float: right;">Tipp!</button>
```

Im JavaScript wird dazu nachfolgend ein Popup-Alarm mit einem Hinweistext hinzugefügt:

```
function Tipp() {
    alert("Bei einem Klick in die Karte wird Ihnen die Uhrzeit des heutigen Sonnenuntergangs am jeweiligen Ort angezeigt.");
}
```

(Github (2018): Get sunset or sunrise at map click)

## 5.4 Leaflet.fullscreen

Dokumentation: <https://github.com/Leaflet/Leaflet.fullscreen>

Mit diesem Plug-In wird in der Karte ein Button Hinzugefügt, mit welchem die Karte auf eine Fullscreen ansicht geschaltet werden kann. Nummer 2 in Abbildung 1 zeigt den Zugehörigen Button zu dieser Such-Funktion Für dieses Plug-In gibt es ein CDN welches mit den folgenden Zeilen ins HTML eingebunden werden kann:

```
<script src="https://api.mapbox.com/mapbox.js/plugins/leaflet-
fullscreen/v1.0.1/Leaflet.fullscreen.min.js"></script>
<link href="https://api.mapbox.com/mapbox.js/plugins/leaflet-
fullscreen/v1.0.1/leaflet.fullscreen.css"
rel="stylesheet" />
```

Initialisiert wird das Plug-In in der JavaScript-Datei mit der nachfolgenden Zeile Code:

```
karte.addControl(new L.Control.Fullscreen());
```

(Github (2018): A fullscreen control for Leaflet)

## 5.5 Leaflet Funktion Control.Scale

Github Dokumentation: <https://leafletjs.com/reference-1.5.0.html>

Die unten links in den Karten der Projektseiten angezeigte Maßstabsleiste würde mittels einer in Leaflet vorgesehen Funktion implementiert. Der dafür verwendete Code in der JavaScript Datei der jeweiligen Seiten lautet:

```
//Maßstab einfügen
const scale = L.control.scale({
  imperial: false,
  metric: true
});
karte.addControl(scale);
```

(Leaflet (2019): Control.Scale)

## 6 Literaturverzeichnis

Fontawesome (2018): Icons. URL: <https://fontawesome.com/icons/paper-plane?style=solid>  
<https://fontawesome.com/icons?d=gallery> [Zugriff am 09.06.2019].

Github (2018): A fullscreen control for Leaflet. URL: <https://github.com/Leaflet/Leaflet.fullscreen>  
[Zugriff am 11.06.2019].

Github (2018): Get sunset or sunrise at map click. URL: <https://github.com/dj0001/Leaflet.Sun>  
[Zugriff am 16.06.2019].

Github (2019): Marker Clustering plugin for Leaflet. URL:  
<https://github.com/Leaflet/Leaflet.markercluster> [Zugriff am 11.06.2019].

Github (2019): Search stuff in a Leaflet map. URL: <https://github.com/stefanocudini/leaflet-search>  
[Zugriff am 11.06.2019].

Google Fonts (2019): Open & Sans. URL:  
<https://fonts.googleapis.com/css?family=Open+Sans&display=swap> [Zugriff am 12.06.2019].

Leaflet (2019): Control.Scale. URL: <https://leafletjs.com/reference-1.5.0.html> [Zugriff am  
11.06.2019].

Open Data Österreich (k.A.): Stadt Wien. Sport und Freizeit. JSON-Format. URL:  
<https://www.data.gv.at/suche/?katFilter%5B2%5D=sportundfreizeit&formatFilter%5B0%5D=JSON&publisherFilter%5B0%5D=Stadt+Wien> [Zugriff am 11.06.2019].

- Open Data Österreich (k.A.): Stadt Wien. Kunst und Kultur. JSON-Format. URL:  
<https://www.data.gv.at/suche/?katFilter%5B1%5D=kunstundkultur&formatFilter%5B0%5D=JSON&publisherFilter%5B0%5D=Stadt+Wien> [Zugriff am 11.06.2019].
- Pixabay (2018): Wiener Riesenrad. URL: <https://pixabay.com/de/photos/wiener-riesenrad-riesenrad-prater-3651215/> [Zugriff am 11.06.2019].
- Pixabay (2018): Stephansdom. URL: <https://pixabay.com/de/photos/wien-vienna-stephansdom-3880488/> [Zugriff am 11.06.2019].
- Pixabay (2016): Der Palast. URL: <https://pixabay.com/de/photos/vienna-%C3%B6sterreich-der-palast-1750493/> [Zugriff am 11.06.2019].
- Unpkg CDN (2018): Leaflet Markercluster. URL: <https://unpkg.com/leaflet.markercluster@1.4.1/dist/> [Zugriff am 11.06.2019].