

Institut für Geographie  
Leopold-Franzens Universität Innsbruck

716409 VU Geoinformatik: Web mapping SS21

Studienkennzahl: UC 066 855

LV Leiter:

Klaus Förster, BSc

Mag. Bernhard Öggl

# **Seen in Österreich und ihre Wasserqualität**

## **Projektbericht**

Rafael Langthaler (01416620)  
Rafael.langthaler@student.uibk.ac.at

Christian Wolf (11716539)  
Csav3937@student.uibk.ac.at

Tim Eastwood (01609222)  
Timothy.eastwood@student.uibk.ac.at

Innsbruck, 16. Juni 2021

## Inhaltsverzeichnis

<b>1</b>	<b>Kurzbeschreibung des Projekts.....</b>	<b>3</b>
<b>2</b>	<b>Darstellung der Implementierungsschritte und eingesetzten Techniken .....</b>	<b>3</b>
2.1	Startseite (index.html).....	3
2.2	Badegewässer im Monitoring der AGES (badegewaesser.html) .....	6
2.3	Badestellen und Grillplätze in Wien (wien.html).....	10
<b>3</b>	<b>Literaturverzeichnis.....</b>	<b>15</b>
3.1	Daten: .....	15
3.2	Plugins: .....	15

# 1 Kurzbeschreibung des Projekts

Das Projekt dient zur Zusammenschau von österreichischen Badegewässern bzw. Grillplätzen in Wien. Im weitesten Sinne soll unser Projekt zur Erholung und Entspannung aller Menschen, die sich in Österreich aufhalten, beitragen. Die Menschen haben einen langen, kalten Winter und mehrere pandemiebedingte Lockdowns hinter sich und wollen jetzt den Sommer richtig genießen! Wo ihnen dieses am besten gelingt, soll unsere Seite Hilfestellung leisten.

## 2 Darstellung der Implementierungsschritte und eingesetzten Techniken

### 2.1 Startseite (index.html)

Für die Erstellung der HTML-Seite wurde das Neuseeland Projekt als Vorlage genutzt und mit weiteren Features erweitert.

```
<!DOCTYPE html>
<html lang="en">

<head>

  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.3/css/all.min.css" />

  <link rel="preconnect" href="https://fonts.gstatic.com">

  <link
    href="https://fonts.googleapis.com/css2?family=Open+Sans:ital@0;1&family=Roboto+Condensed:ital@0;1&display=swap"
    rel="stylesheet">

  <link rel="stylesheet" href="main.css">

  <title> Die Badeseen Österreichs </title>
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/leaflet/1.7.1/leaflet.css">

  <script src="https://cdnjs.cloudflare.com/ajax/libs/leaflet/1.7.1/leaflet-src.min.js"></script>

  <link rel="stylesheet" href="main.css">

  <script defer src="main.js"></script>
```

Als Header wurde ein Bild vom Achensee benutzt was von pixabay.com stammt und frei benutzbar ist. Der Urheber ist genannt und man kann über einen Link gleich auf die Datei auf pixabay.com gelangen.

Angaben zum Ersteller der HTML-Page wurde ebenfalls aus dem Neuseeland Projekt übernommen.

```

<body>

  <header>
    
    <p class="white"><a href="https://pixabay.com/photos/lake-achensee-austria-tyrol-2057240/">mbc-2016</a></p>

    <h1></h1>
    <nav>
      <a href="https://pixabay.com/photos/lake-achensee-austria-tyrol-2057240/">
        
      </a>
      <address>
        <a href="mailto:timothy.eastwood@student.uibk.ac.at"><i class="fas fa-envelope-square"></i>Tim
          Eastwood</a> -
        <a href="https://github.com/timeastwood6020"><i class="fab fa-github"></i>github Repo</a>
      </address>
    </nav>
  </header>

```

Als Intro wurde ein kleiner Paragraf zu allgemeinen Informationen zu Seen in Österreich von der Webseite des Bundesministeriums für Landwirtschaft, Regionen und Tourismus übernommen. Ein Link zur Webseite ist darunter eingebaut.

```

<main>
  <h1>Die Badeseen Österreichs</h1>
  <p>
    In Österreich gibt es mehr als 25.000 stehende Gewässer mit einer Fläche größer als 250 m². Hierzu zählen
    sowohl natürlich entstandene große Seen, Lacken, Klein- und Augewässer wie auch die künstlich entstandenen
    Baggerseen, Teiche, Speicherseen und Stauseen.

    Es gibt ca. 2.140 stehende Gewässer, deren Fläche größer als 1 ha ist. Die Gesamtfläche dieser Gewässer
    beträgt rund 613 km², was ca. 0,7% der Staatsfläche entspricht.

    62 Seen sind "große Seen" mit einer Fläche über 50 ha. Davon sind 43 Seen als "natürlich" und 19 als
    "künstlich" ausgewiesen.

    (Quelle: <a
      href="https://info.bmlrt.gv.at/themen/wasser/wasser-oesterreich/zahlen/fluesse_seen_zahlen.html">Bundesministerium:
      Landwirtschaft, Regionen und Tourismus</a>)
  </p>

```

Um den Besuchern der Webseite zu helfen sich für einen Badensee zu entscheiden, wurden 2 Inspirationsquellen eingefügt: Eine Webseite und ein direkt eingebautes Youtube video.

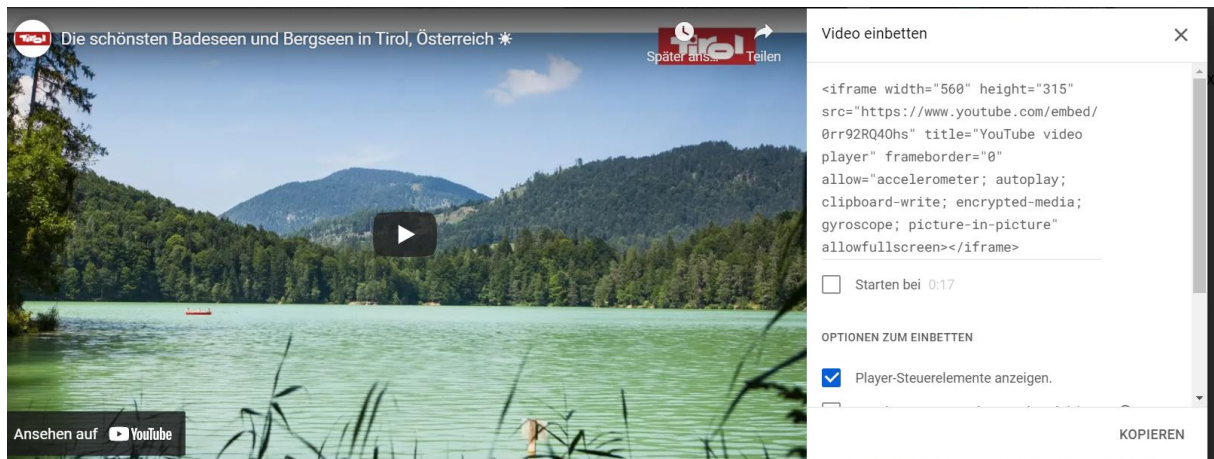
```

<h2>Die schönsten Seen Österreichs</h2>

<h2> Wohin gehts?!</h2>
<p>
  Inspiration zum nächsten Badeausflug gesucht? Schauen Sie sich doch mal hier um!
  <a href="https://www.iamstudent.at/blog/badeseen-oesterreich/">iamstudent: DIE SCHÖNSTEN BADESEEN
    ÖSTERREICH: ERFRISCHUNG NAHT!</a>
</p>

<p>Oder vielleicht hilft Ihnen dieses kurze Video auf die Sprünge ins Kühle Nass!</p>
<iframe width="896" height="504" src="https://www.youtube.com/embed/0rr92RQ40hs" title="YouTube video player"
  frameborder="0"
  allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture"
  allowfullscreen></iframe>
<a href="https://www.youtube.com/watch?v=0rr92RQ40hs">YouTube: Tirol</a>

```



Auf Youtube gibt es die Funktion das Video direkt auf eine Seite einzubetten. Der Code wurde in Visual Studio Code kopiert und die Größe des Videos auf die HTML-Seite angepasst.

```
<h2>Die Wasserqualität Österreichischer Seen</h2>
<p>
  <h3> Limnologie </h3>
  Die Limnologie ist die Wissenschaft von den Binnengewässern als Ökosystemen, deren Struktur, Stoff- und Energiehaushalt und biologisch-ökologische Struktur und Funktion sie erforscht und deren abiotische und biotische Prozesse sie zu quantifizieren sucht. <a href="https://de.wikipedia.org/wiki/Limnologie">Wikipedia</a>
</p>
```

Paragraph erstellt mit Verlinkung zur Wikipedia Seite vom Begriff „Limnologie“.

```
Die Messstationen, sowie die aktuelle Wassertemperatur aller Seen in Österreich findet ihr hier:

<nav>
  <a class="back" href="https://lakesaustria.github.io/badegewaesser.html?fbclid=IwAR1og_Rl05c6jh4M1xjIpet9eTSzpya0VvqbD0jDmYDogw55a_yka" class="fas fa-arrow-left"><i>Aktuelle Wassertemperatur der Seen in Österreich</i></a>
</nav>
</p>
```

Erstellung einer Navigation um auf die zweite Seite zu wechseln. Hier ist eine Karte mit der aktuellen Wassertemperaturen der Seen zu sehen.

```
<h2> Grillen, Chillen & Bierchen killen</h2>


<p class="white"><a href="https://pixabay.com/de/photos/grillen-hotdogs-hamburger-burger-1081675/">Free-Photos</a></p>

<p>
  Zum perfekten Badetag am See gehört eine Grillerei unter Freunden und Familie einfach dazu!

  Wo genau die besten Spots zum Grillen sind findest du auf der nächsten Seite!

  <nav>
    <a class="back" href="https://lakesaustria.github.io/badegewaesser.html?fbclid=IwAR1og_Rl05c6jh4M1xjIpet9eTSzpya0VvqbD0jDmYDogw55a_yka" class="fas fa-arrow-left"><i>Zu den Grillplätzen</i></a>
  </nav>
</p>
</main>
```

Als Überleitung zur dritten Seite/zweite Karte unserer Webseite ist ein Bild von einem Grill. Die Verlinkung zum Urheber und Webseite des Bildes ist auch da.

Eine Navigationsleiste bringt die Besucher der Seite sofort zu der Karte mit den Grillplätzen!

## 2.2 Badegewässer im Monitoring der AGES ([badegewaesser.html](#))

### Datenbeschaffung:

Die Daten zu den österreichischen Badegewässern wurden über <https://www.data.gv.at/katalog/dataset/badegewaesser> beschafft. Dabei sind die Daten auf einer Array-Objekt Struktur aufgebaut, so besteht der Datensatz aus einem Array von Bundesland-Objekten, die wiederum aus einem Array von Badegewässer-Objekten bestehen. Letztlich bestehen auch die Messwerte aus einem Array von Messdaten-Objekten. Leider können die Daten nicht direkt von der AGES abgefragt werden, dabei kommt es zu einem CORS Fehler. Man muss daher die JSON-Daten herunterladen, weswegen die Daten nicht einer Echtzeitqualität entsprechen, wohl aber tagesaktuell sind, wenn man die JSON-Daten täglich neu herunterlädt.


# Katalog österreichische Badegewässer

Ergebnisse der regelmäßigen Messungen von E.coli, Enterokokken, Sichttiefe und Wassertemperatur der österreichischen Badegewässer. Diese Daten werden mindestens fünf mal pro Badesaison erhoben und tagaktuell zur Verfügung gestellt.

## Daten und Ressourcen

Badegewaesser.json **JSON**

 Mehr Information

 Zur Ressource

Datenverantwortliche Stelle	AGES - Institut für medizinische Mikrobiologie und Hygiene Wien
Kontaktseite der datenverantwortlichen Stelle	<a href="http://www.ages.at/ages/geschaeftsfelder/oeffentliche-gesundheit/imed-wien/">http://www.ages.at/ages/geschaeftsfelder/oeffentliche-gesundheit/imed-wien/</a>
Veröffentlichende Stelle	AGES
Lizenz	Creative Commons Namensnennung 3.0 Österreich
Link zur Lizenz	<a href="https://creativecommons.org/licenses/by/3.0/at/deed.de">https://creativecommons.org/licenses/by/3.0/at/deed.de</a>

```
{
  "VERSION": "122914",
  "BUNDESLAENDER": [{
    "BUNDESLAND": "Burgenland",
    "BADEGEWAESSER": [{
      "BADEGEWAESSERID": "AT1051051100150010",
      "BADEGEWAESSERNAME": "Naturbadesee K\u00f6fnigsdorf",
      "MESSWERTE": [{
        "D": "01.06.2021",
        "E": 15,
        "E_C": 21,
        "W": 9.800000000000007,
        "S": 3,
        "A": 1
      }]
    }]
  ]
}
```

## Implementierungsschritte:

Wie angedeutet, wurden die JSON-Daten, wie etwa beim Beispiel Wien Tourismus, heruntergeladen und über eine Javascript Datei verlinkt. Hier wurde bereits eine aktuellere Version der Daten – 15.06.2021 – eingebunden:

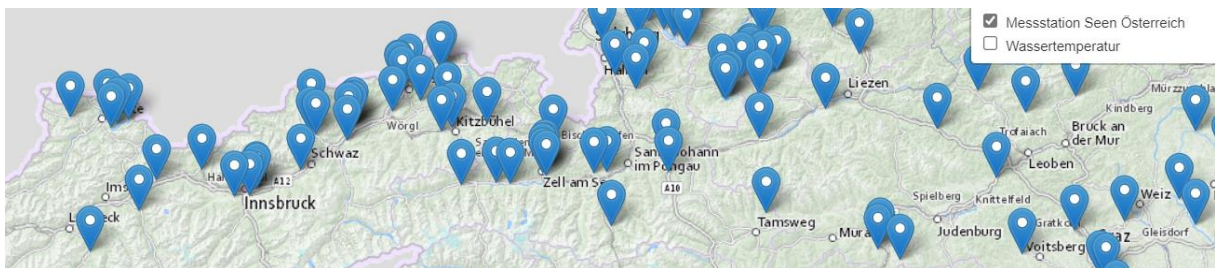
```
JS badegewaesser.js > [?] BADEGEWAESSER > data
1 // https://www.ages.at/typo3temp/badegewaesser_db.json
2 // https://www.data.gv.at/katalog/dataset/badegewaesser/resource/7a90ef57-8cb2-4d62-a474-aa2085733de9
3
4 const BADEGEWAESSER = [{
5   title: "Badegewaesser Oesterreich",
6   source: "https://www.data.gv.at/katalog/dataset/badegewaesser/resource/7a90ef57-8cb2-4d62-a474-aa2085733de9",
7   data: "data/badegewaesser_db_15_06.json"
8 }];
9
10
```

## Eingesetzte Techniken – Marker setzen

Der Aufbau der Daten benötigte eine andere Codierung als in unseren bisherigen Beispielen, um auf die einzelnen Geo- und Messdaten zuzugreifen. Im Detail wurde mittels zwei for-

Schleifen durch die Bundesländer bzw. Badegewässer geloopt, um die Marker-Koordinaten „LATITUDE“ und „LONGITUDE“ auszulesen. Der Gedanke dazu lautete: „Für jedes Bundesland, jedes Gewässer nach LongLat zu durchsuchen und einen Marker setzen.“

```
let dummyUrl = 'data/badegewaesser_db.json';
fetch(dummyUrl)
  .then(response => response.json())
  .then(json => {
    // console.log("Data: ", json.BUNDESLAENDER);
    for (station of json.BUNDESLAENDER) {
      //console.log("Badegewässer: ", station.BADEGEWAESSER);
      for (lakestation of station.BADEGEWAESSER) {
        //console.log("Gewässername: ", lakestation.BADEGEWAESSERNAME);
        let marker = L.marker([
          lakestation.LATITUDE,
          lakestation.LONGITUDE,
        ]);
```



### Pop-up hinzufügen

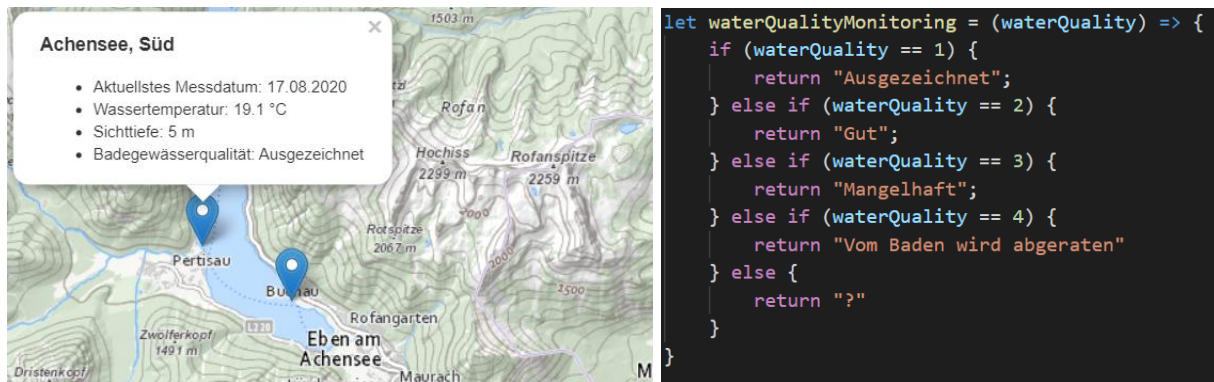
Ähnlich wie beim AWS-Tirol Beispiel wurde ein Pop-up mit ausgewählten Messdaten hinzugefügt. Die Variable „lakestation“ entspricht dabei der Messstation bzw. des Zugangs zu einem Badegewässer, etwa „Pertisau, Achensee“. Die Messwerte bestehen bekanntlich aus einem Array aus Objekten, das erste Objekt [0] enthält dabei immer die aktuellen Messwerte (A, D, W, S). Für die Badegewässerqualität wurde eine Funktion eingebunden, die die Skala (1-4) zu der entsprechenden Wortbeschreibung („Ausgezeichnet“ – „Vom Baden wird abgeraten“) umschreibt.

```
marker.bindPopup(`
<h3>${lakestation.BADEGEWAESSERNAME}</h3>
<ul>
  <li>Aktuellstes Messdatum: ${lakestation.MESSWERTE[0]['D']}</li>
  <li>Wassertemperatur: ${lakestation.MESSWERTE[0]['W']} °C</li>
  <li>Sichttiefe: ${lakestation.MESSWERTE[0]['S']} m</li>
  <li>Badegewässerqualität: ${waterQualityMonitoring(waterbodyQuality)}</li>
</ul>`)
```



## Funktion waterQualityMonitoring()

Die Funktion waterQualityMonitoring() ist simpel. Es würde sich anbieten die Badegwasserqualität in einen Layer zu verpacken und mit einer Bildbeschreibung, etwa Smileys, darzustellen.



## WassertemperaturLayer mit Farbdiversifizierung

Es wurde ein Layer zu der Wassertemperatur erstellt, der die Temperatur von kalt nach warm in Blau- zu Rottönen darstellt. Die Temperaturintervalle betragen dabei ab 15 ° bis 30 ° C durchgehend 2,5 °, um die Skala zu differenzieren. Da die Temperatur der österreichischen Seen im Sommer meist um die 24 ° C liegt.



### 2.3 Badestellen und Grillplätze in Wien (wien.html)

In dem folgenden Teil der Projektarbeit wird beschrieben, wie die Seite zu den Badestellen und Grillplätzen in Wien ausgearbeitet wurde. Dazu wurde zuerst die Grundstruktur erstellt. Hierfür wurden die beiden Files „wien.html“ und „wien.css“ benötigt. Die Überschrift der Seite wurde formuliert und die Parameter für die Karte wurden festgelegt. Um eine Darstellung von Wien in der Karte zu erhalten, wurde die Leaflet Bibliothek hinzugefügt. Zudem wurde Leaflet-providers in das wien.html File integriert. In dem Javascript File wien.js wurde eingestellt welcher Punkt im Zentrum stehen soll und in welchem Maß in der Karte gezoomt ist.

Die Daten zur Darstellung der Badestellen und Grillplätze stammen von data.gv.at. Beides sind JSON-Dateien. Um die Datensätze einzubauen wurde eine neues Javascript File „ogdwien.js“ erstellt und in „wien.html“ integriert.

```
1 + const OGDWIEN = [{
2 +   title: "Badestellen Standorte Wien",
3 +   source: "https://www.data.gv.at/katalog/dataset/7c6e9dbf-487b-4146-a379-19e46642c392",
4 +   data: "data/BADESTELLENOGD.json"
5 + }, {
6 +   title: "Grillplätze Standorte Wien",
7 +   source: "https://www.data.gv.at/katalog/dataset/c6fc4688-1aca-4363-bd1a-8d14ea08d2f8",
8 +   data: "data/GRILLPLATZOGD.json"
9 + }];
```

Dieses Javascript File beinhaltet den Titel, eine URL-Quelle von wo die Daten stammen und den Ordner in dem sich die Daten befinden. In dem Fall ist es der Unterordner „data“, welcher sich im Repository befindet. In einem weiteren Schritt wurde mit dem fetch-Befehl ein Pin erstellt, wo sich die Badestellen befinden. Zudem wurde die Möglichkeit für ein Popup erstellt.

```

17 + fetch("data/BADESTELLEN0GD.json")
18 + .then(response => response.json())
19 + .then(stations => {
20 +     L.geoJson(stations, {
21 +         onEachFeature: (feature, layer) => {
22 +             layer.bindPopup(feature.properties.BEZEICHNUNG)
23 +         }
24 +     }).addTo(map);
25 + }) ⊖

```

Damit die Pins der Badestellen später besser von denen der Grillplätze unterschieden werden können, wurde ein passendes Icon von Mapicons ausgewählt und wie in der folgenden Abbildung dargestellt eingebunden. Ein weiterer Unterordner icons wurde im Repository erstellt. So konnte das Icon über die „iconUrl“ abgefragt werden und die Größe mit „iconSize“ vergeben werden.

```

20 20      L.geoJson(stations, {
21 21          onEachFeature: (feature, layer) => {
22 22              layer.bindPopup(feature.properties.BEZEICHNUNG)
23 +      },
24 +      pointToLayer: (geoJsonPoint, latlng) => {
25 +          return L.marker(latlng, {
26 +              icon: L.icon({
27 +                  iconUrl: 'icons/badestelle.png',
28 +                  iconSize: [39, 39]
29 +              })
30 +          })
23 31      }
24 32      }).addTo(map);
25 33  }) ⊖

```

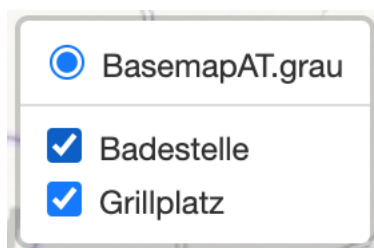
Um mehrere Datensätze gleichermaßen einzubinden wurde eine For-Schleife programmiert, die nach dem Titel der Daten in „ogdwien.js“ sucht. So wurde für die Badeplätze zum Beispiel die Funktion drawBathingSpot geschrieben welche ebenso durch die For-Schleife ausgeführt wird.

```

17 - fetch("data/BADESTELLEN0GD.json")
18 - .then(response => response.json())
19 - .then(stations => {
20 -   L.geoJson(stations, {
21   17 + let drawBathingSpot = (geojsonData) => {
22   18 +   L.geoJson(geojsonData, {
23   19     onEachFeature: (feature, layer) => {
24   20       layer.bindPopup(feature.properties.BEZEICHNUNG)
25   21     },
26   22     @@ -30,4 +28,14 @@ fetch("data/BADESTELLEN0GD.json")
27   23   }
28   24   })
29   25   }
30   26   }).addTo(map);
31   27   }
32   28   }
33 - }) ⊖
34   31 + }
35   32 +
36   33 + for (let config of OGDWIEN) {
37   34 +   fetch(config.data)
38   35 +     .then(response => response.json())
39   36 +     .then(geojsonData => {
40   37 +       if (config.title == "Badestellen Standorte Wien") {
41   38 +         drawBathingSpot(geojsonData);
42   39 +       }
43   40 +     })
44   41 +   } ⊖

```

Mit attribution wurde noch ein Zitat in die Karte eingefügt von wo die Datensätze der Badestellen und Grillplätze sowie die Icons stammen. Damit die Badestellen ein und ausgeblendet werden können, wurde mit „overlays“ gearbeitet. Diese wurden der „layerControl“ hinzugefügt.



Um die Grillplätze darzustellen wurde ebenfalls eine derartige Funktion wie „drawBathingSpot“ erstellt. Für diese Funktion wurde der Name „drawbbqArea“ ausgewählt. Diese wurde anschließend der For-Schleife hinzugefügt.

```

42    61    for (let config of OGDWIEN) {
43    62        fetch(config.data)
44    63            .then(response => response.json())
45    64            .then(geojsonData => {
46    65                if (config.title == "Badestellen Standorte Wien") {
47    66                    drawBathingSpot(geojsonData);
48    67                } else if (config.title == "Grillplätze Standorte Wien") {
49    68                    drawbbqArea(geojsonData);
50    69                }
51    70            })
52    71        }

```

In den Popups der Grillplätze wird die Lage beschrieben und die Grillplatz ID vergeben. Außerdem wird angegeben ob für den jeweiligen Grillplatz eine Reservierung notwendig ist und ein Weblink mit einer weiteren Beschreibung bereitgestellt. Hierüber soll es auch gleich möglich sein die Reservierungspflichtigen Grillplätze zu reservieren.

```

47    48        onEachFeature: (feature, layer) => {
48    49            layer.bindPopup(feature.properties.LAGE)
49    50            layer.bindPopup(`<strong>Lage: ${feature.properties.LAGE}</strong>
50    51            <hr>
51    52            Grillplatz ID: ${feature.properties.GRILLPLATZ_ID} <br>
52    53            Reservierung notwendig: ${feature.properties.RESERVIERUNG} <br>
53    54            <a href="${feature.properties.WEBLINK1}">Weitere Info</a>`)
49    54        },

```

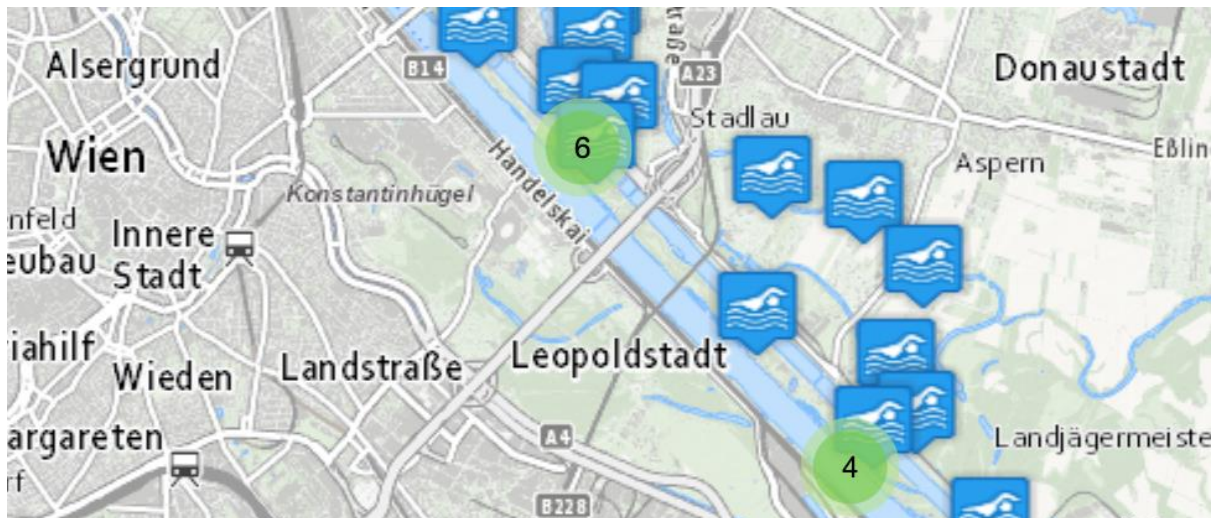
Ebenso wie für die Badestellen, werden die Grillplätze als „overlays“ hinzugefügt und mit Hilfe der „layerControl“ dargestellt.

In den Popups der Badestellen wird die Bezeichnung und der Bezirk angegeben.

Anschließend wurden noch einige nützliche Leaflet Plugins in die Karte hinzugefügt. Einige Darunter waren das Leaflet-hash Plugin und das Leaflet.fullscreen Plugin welche schnell implementiert wurden konnten. CDNJS bot hierfür die notwendigen Downloads an, welche „wien.html“ hinzugefügt wurden.

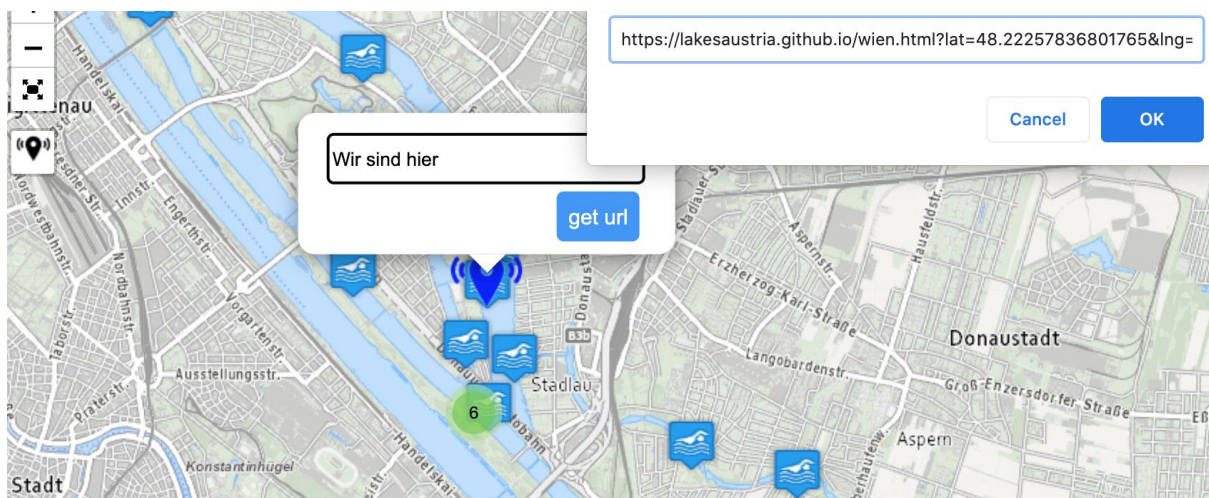
Ein weiteres Plugin welches implementiert wurde, ist Leaflet.markercluster. Hiermit sollen die Grillplätze zusammengeführt werden und sind in der folgenden Abbildung grün dargestellt.





Zudem wurde eine Minimap erstellt, welche einklappbar ist und beim Aufruf der Karte eingebildet ist. In der linken unteren Ecke wurde als weiteres Kartenelement noch ein Maßstab hinzugefügt.

Ein letztes Plugin soll dem Nutzer der Karte die Möglichkeit geben, einer anderen Person einen Standort zu senden. Um dieses zu implementieren, wurde ein vorgefertigtes Javascript File mit dem Namen „Leaflet.LocationShare.js“ dem Projekt hinzugefügt und in „wien.html“ das script erstellt. Wie in der folgenden Abbildung dargestellt, wird eine Url erzeugt welche zum Beispiel als Textnachricht versendet werden kann. Der Empfänger erhält dann den Standort auf der Karte.



In weiterer Folge wurde das Layout mit Hilfe des „wien.html“ Files und des „wien.css“ Files angepasst. Unter anderem wurde von Pixabay ein Headerbild heruntergeladen und mit dem

Image Resizer von „redketchup.io“ die Größe angepasst. Das Bild, sowie ein Link zu der Person die es veröffentlicht hat, wurden im „header“ erzeugt. Der „main“ Teil beinhaltet die Überschrift sowie einen kleinen Text und die Karte. Im Text befindet sich noch ein Link zu wien.gv.at Seite. Mit Hilfe von „wien.css“ konnte dann noch Formatierungen am Text und dem Link im Headerbild vorgenommen werden. Hierfür wurden zudem andere Schriftarten mit „Google Fonts“ definiert.

### 3 Literaturverzeichnis

#### 3.1 Daten:

Bundesministerium für Digitalisierung und Wirtschaftsstandort: Badestellen Standorte Wien.

URL: <https://www.data.gv.at/katalog/dataset/7c6e9dbf-487b-4146-a379-19e46642c392>

Bundesministerium für Digitalisierung und Wirtschaftsstandort: Grillplätze Standorte Wien.

URL: <https://www.data.gv.at/katalog/dataset/c6fc4688-1aca-4363-bd1a-8d14ea08d2f8>

Mapicons: URL: <https://mapicons.mapsmarker.com/markers/restaurants-bars/restaurants/barbecue/>

Mapicons: URL: <https://mapicons.mapsmarker.com/markers/sports/relaxing-sports/swimming/>

Pixabay: Baden. URL: <https://pixabay.com/photos/swimming-jump-fun-people-young-388910/>

Google Fonts: URL: <https://fonts.google.com/>

#### 3.2 Plugins:

CDNJS: Leaflet. URL: <https://cdnjs.com/libraries/leaflet>

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

CDNJS: URL: <https://cdnjs.com/libraries/leaflet.fullscreen>

Leaflet-hash: URL: <https://github.com/mlevans/leaflet-hash>

CDNJS: URL: <https://cdnjs.com/libraries/leaflet-hash>

Leaflet.markercluster: URL: <https://github.com/Leaflet/Leaflet.markercluster>

CDNJS: URL: <https://cdnjs.com/libraries/leaflet.markercluster>

Leaflet.MiniMap: URL: <https://github.com/Norkart/Leaflet-MiniMap>

CDNJS: URL: <https://cdnjs.com/libraries/leaflet-minimap>

Leaflet-providers: URL: <https://github.com/leaflet-extras/leaflet-providers>

CDNJS: URL: <https://cdnjs.com/libraries/leaflet-providers>

L.LocationShare: URL: <https://github.com/CliffCloud/Leaflet.LocationShare>