

VU Geoinformatik: Webmapping

LV-Leiter: Föster Klaus, BSc. & Mag. Öggl Bernd



## **Projektbericht - Visualisierung von eingelesenen GPX-Daten zu Mountainbikestrecken um Salzburg**

---

Jeff Reding, BSc. 01318844

Magnus Kramer, BSc. 01315194

Roman Fendt, BSc. 01320225

Abgabedatum: 13.06.2018

SS18

## Inhaltsverzeichnis

1. Konzept.....	3
2. Umsetzung.....	4
2.1 html - Dateien.....	4
2.1.1 index.html.....	4
2.1.2 team.html .....	5
2.1.3 touren.html .....	6
2.2 java - Dateien .....	7
2.2.1 startmap.js .....	7
2.2.2 toureninfo.js.....	8
2.2.3 salzBike.js.....	9
3. Fazit.....	11

## 1. Konzept

In unserem Webmapping-Projekt beschäftigten wir uns mit verschiedenen Mountainbikestrecken rund um die Stadt Salzburg. Das Salzburger Land zählt mit seinen 7156 Quadratkilometern zu einem der beliebtesten Erholungsgebiete Zentraleuropas und eignet sich daher sehr gut als Projektgrundlage. Da das Salzburger Land ein breit gefächertes Spektrum an Besuchern anzieht, von der Stadtfamilie bis zum Extremsportler, sollten die Radtouren zusätzliche Informationen über Schwierigkeitsgrad, Länge und Dauer beinhalten. Um dies zu ermöglichen, wird eine Karte der beliebtesten Mountainbiketouren Salzburgs erstellt. Unterhalb der Karte werden die verschiedenen Zusatzinformationen angezeigt. Höhenprofil und Steigung der Tourenabschnitte werden ebenfalls dargestellt.

Es sollen verschiedene Grundkarten genutzt werden können, welche jeweils differenzierte Informationen beinhalten und dementsprechend je nach Anspruch gewählt werden können. Über gpx- files sollen Koordinaten und Höhendaten zu den jeweiligen Bikestrecken eingespeist werden. Zusätzlich sollen über eine weitere js-file Informationen bezüglich Distanz, zurückzulegende Höhenmeter, Schwierigkeit, etc. zu den jeweiligen Strecken hinzugefügt werden.

Zentrale Herausforderung bei der Umsetzung des Projekts wird der automatisierte Einbezug der verschiedenen Datengrundlagen (Länge, Schwierigkeit, Dauer,...) in die Karte. Im Vordergrund stehen die Informationen über sämtliche Aspekte der Routen, anhand welcher sich für jeden Nutzer eine individuell auf seine Bedürfnisse zugeschnittene Route finden lässt.

Die Verknüpfung der html-Dateien mit den Javascript-Dateien, sowie die Einspeisung der verschiedenen Leaflet- Plugins wird auf Grundlage des im Unterricht gelernten angewendet und erweitert.

## 2. Umsetzung

Am Beginn der Umsetzung steht die Beschaffung der benötigten Daten. Da unter *data.gv.at* nur Zugriff auf die *shapefiles* und *geojson-Dateien* (ohne Höheninformation) der verschiedenen Touren möglich ist, mussten die *gpx-files* über <https://www.bergfex.at/sommer/salzburg/touren/bike/> bezogen werden. Unter dem Link befinden sich nebst den gpx-Tracks, Daten über diese Eigenschaften der Touren:

Startpunkt	Parkplatz	Landschaft
Kondition	Distanz in km	Dauer in Stunden
Schwierigkeit	Aufstieg in hm	

Im Folgenden soll auf die verschiedenen html- und Javascript-Dateien, inklusive deren Aufbau, eingebauter Plugins, sowie Herausforderungen beim Erstellen, eingegangen werden.

### 2.1 html-Dateien

Für die Internetseite wurden drei html-Dateien; *index.html*, *Team.html* und *Touren.html* angelegt. Die Rollen und Eigenschaften dieser wird nun im folgenden Teil des Berichtes aufgeführt und erörtert.

#### 2.1.1 index.html

Die *index.html* stellt die Startseite unserer Homepage dar. In der Datei wurden zudem die benötigten Plugins geladen. Die Startseite beinhaltet eine Übersichtskarte, anhand welcher sich der Nutzer einen groben Überblick über die verfügbaren Routen verschaffen kann. Für die Startseite nutzten wir zum einen das *fullscreen*-Plugin, welches dem Nutzer erlaubt, die Karte im Vollbildmodus betrachten zu können.

##### Including via CDN

Leaflet.fullscreen is available through the Mapbox Plugin CDN - just copy this include:

```
<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'></link>
```

Abbildung 1: Leaflet link des Fullscreen Plugins

Quelle: Screenshot aus leafletjs.com

Der Fullscreen Modus ist durch das Mapbox Plugin CDN erreichbar und muss deshalb nur als Link hinzugefügt werden. (Siehe *Abb. 1*). Anschließend wird es in der .js file durch: *fullscreenControl: true;* eingeschaltet.

Zum anderen wurde das *omnivore*- Plugin aus der leaflet- Bibliothek verwendet, durch welches das Einspeisen verschiedener Datentypen ermöglicht wird. Falls zukünftige Routen zur Karte hinzugefügt werden sollen, welche beispielsweise einer geojson-Datei, -Daten aus Google Earth oder einer csv- Datei entsprechen, ist das Einbinden dieser Datensätze problemlos machbar. Die Einbindung des *omnivore*- Plugins in die Seite ist in *Abb.2* dargestellt.

```
114
115     // GPX Track laden
116     trailGroup.clearLayers();
117     gpxTrack = omnivore.gpx('data/' + track).addTo(trailGroup)
118
```

*Abbildung 2.: Einbindung des omnivore- Plugins in die js-Datei*

*Quelle: Screenshot aus eigener Datei*

### 2.1.2 Team.html

In der Datei, sowie in allen anderen html-Dateien, wurde außerdem die *stylesheet.css* Datei eingefügt, durch welche das Design der Website vorgenommen wird. Des weiteren wurden unsere Bilder im Reiter „Die Boys“ stellvertretend für eventuelle Reiseleiter, Guides, etc. eingefügt, sowie unsere Kontakt-Informationen in Form der Email-Adressen (*Abb.3*) angegeben.

```

46 <table class="team">
47 <tr>
48 <td>
49 
50 </td>
51
52
53 <td>
54 
55 </td>
56
57
58 <td>
59 
60 </td>
61 </tr>
62 <tr>
63 <td>
64 <a href="mailto:jeff.reding@student.uibk.ac.at" class="link">Jeff Reding</a>
65 </td>
66 <td>
67 <a href="mailto:magnus.kramer@student.uibk.ac.at" class="link">Magnus Kramer</a>
68 </td>
69 <td>
70 <a href="mailto:roman.fendt@student.uibk.ac.at" class="link">Roman Fendt</a>
71 </td>
72 </tr>
73 </table>

```

Abbildung 3: Einbindung der Teamfotos und der Kontaktinformationen

Quelle: Screenshot aus eigener Datei

### 2.1.3 Touren.html

Die Touren.html stellt wichtige Informationen über die Karte bereit. In dieser ist durch das *elevation*- Plugin ein Werkzeug verwendet worden, mit welchem sich Daten aus dem gpx-Dateien als Polylinien in einem interaktiven Höhenprofil darstellen lassen. Durch dieses Plugin erhält der Nutzer genaue Informationen über Steigungen sowie Entfernungen und deren spezifische Verteilung. Somit kann eine Tour noch genauer geplant werden und auf das individuelle Können der Mountainbiker eingegangen werden. Zudem wird ein *container* angelegt. In diesem sind die gpx-Dateien der verschiedenen Touren enthalten. Durch einen *dropdown-content* wird die Liste für die individuelle Auswahl erstellt, durch welche der Nutzer eine Tour auswählen kann.

```

76 <table border="1">
77 <tr>
78 <td>
79 <h4>Start- und Endpunkt</h4>
80 <div id="Start"></div>
81 </td>
82 <td>
83 <h4>Parkplatz</h4>
84 <div id="Parkplatz"></div>
85 </td>
86
87 <td>
88 <h4>Kondition </h4>
89 <div id="Kondition"></div>
90 </td>
91
92 <td>
93 <h4>Länge in Km</h4>
94 <div id="Laenge"></div>
95 </td>
96 <td>
97 <h4>Dauer in Std. </h4>
98 <div id="Zeit"></div>
99 </td>
100 </tr>
101 <tr>
102 <td>
103 <h4>Landschaft</h4>
104 <div id="Landschaft"></div>
105 </td>
106 <td>
107 <h4>Schwierigkeit</h4>
108 <div id="Schwierigkeit"></div>
109 </td>
110 <td>

```

Abbildung 4: Tabelle über die Toureninformationen

Quelle: Screenshot aus eigener Datei

Unterhalb der Karte wird eine Tabelle erstellt, in welcher sich Informationen über Startpunkt, Parkmöglichkeiten, Schwierigkeit, Distanz, Dauer, Landschaft, benötigte Kondition, Anspruch an Technik, Aufstieg, höchster Punkt und dem Ziel der jeweiligen Tour befinden. Dies wird in *Abb.4* dargestellt. Auch wird unter der Karte eine Beschreibung der jeweiligen Charakteristika der Tour angezeigt.

## 2.2 Javascript Dateien

Im Zuge des Projekts wurden drei Javascripts geschrieben. Hierbei handelt es sich um die Dateien *salzBike.js*, *startmap.js* und *Toureninfo.js*. Im folgenden Teil des Berichts werden diese einzeln aufgeführt und ihr Aufbau eingehend erklärt.

### 2.2.1 startmap.js

In der *startmap.js* Datei werden zu Beginn die drei Hintergrundkarten eingespeist. Die OpenStreetMap-Karte wurde unter *openstreetmap.org* bezogen, wohingegen die Geolandbasemap und das Orthophoto mit einer Rastergröße von 30 Zentimeter auf Grundlage der Daten der Adresse *wien.gv.at* hinzugefügt wurde. Diese wurden allesamt unter *mylayers* in die Karte integriert. Des Weiteren wurde die Steuerungsfunktion zur Auswahl der Karten mithilfe von *mymapcontrol* hinzugefügt, angepasst und links unten positioniert. Durch *set.View* wurden manuell die Ausgangskordinaten zur Ausrichtung des Startausschnitts der Karte hinzugefügt. Wie in *Abb.5* dargestellt, werden im letzten Teil des Scripts mithilfe des *omnivore* - Plugins sämtliche gpx-Tracks in die *startmap.js* Datei eingefügt.

```

53
54 //let gpxTrack = new L.GPX("data/Gaisberg_Gipfel.gpx",{async:true}).addTo(myMap);
55
56 var gpxGaisberg_Zistelam = omnivore.gpx("data/Gaisberg_Zistelam.gpx").addTo(myMap);
57 var gpxGaisberg_Gipfel = omnivore.gpx("data/Gaisberg_Gipfel.gpx").addTo(myMap);
58 var gpxHoeglrunde = omnivore.gpx("data/Hoeglrunde.gpx").addTo(myMap);
59 var gpxKleine_Dax_Lueg_Ausfahrt = omnivore.gpx("data/Kleine_Dax_Lueg_Ausfahrt.gpx").addTo(myMap);
60 var gpxMuehlsteinrunde = omnivore.gpx("data/Muehlsteinrunde.gpx").addTo(myMap);
61 var gpxRupertiwinkel = omnivore.gpx("data/Rupertiwinkel.gpx").addTo(myMap);
62 var gpxStaufen_Umrundung = omnivore.gpx("data/Staufen_Umrundung.gpx").addTo(myMap);
63 var gpxWallersee_Rundfahrt = omnivore.gpx("data/Wallersee_Rundfahrt.gpx").addTo(myMap);
64

```

*Abbildung 5:* Einbindung der gpx-Dateien

*Quelle:* Screenshot aus eigener Datei

### 2.2.2 toureninfo.js

In dieser Datei sind sämtliche relevanten Daten über die einzelnen Touren eingespeichert. Der Aufruf erfolgt durch den Befehl `window.TOURENINFO`. Für zusammenfassende Informationen wurde ein Kurztext, sowie eine Beschreibung der Tour angelegt. Ergänzend werden nun die Tourdaten eingespeist, auf welche in der Tabelle unter der Karte zugegriffen wird, um diese anzeigen zu können. Dies wird für alle acht Touren vorgenommen. Anhand von *Abb.6* kann am Beispiel der Höglrunde der benötigte Code für eine einzelne Tour entnommen werden.

```
48 "Hoeglrunde.gpx": {
49   "Titel": "Hoeglrunde",
50   "Kurztext": "Die Högl – Runde bietet das perfekte Fahrerlebnis für jung und alt. Sie zeichnet sich durch einen mäßigen Schwi
51   "Tourenbeschreibung": "Vom Ortszentrum Piding geht's auf Asphalt aufwärts zum Berggasthof Johannishögl und der daneben lieg
52   "KoordinatenStart": "47.770111, 12.91143",
53   "Start": "Pidinger Hauptbahnhof",
54   "KoordinatenZiel": "47.77002, 12.91151",
55   "Ziel": "Ortszentrum Piding",
56   "auf": 540,
57   "ab": 540,
58   "HoechstPunkt": 768,
59   "NiedrigsterPunkt": 453,
60   "Schwierigkeit": "leicht",
61   "Kondition": "2/6",
62   "Technik": "2/6",
63   "Landschaft": "5/6",
64   "Laenge": 23.87,
65   "Zeit": 2.5,
66   "Parkplatz": "Ortszentrum – Haus der Vereine",
67   "Quelle": "https://www.bergfex.de/sommer/bayern/touren/mountainbike/10577,mtb-24--hoegl-runde-von-sueden/",
68   "Foto": "",
69 },
--
```

*Abbildung 6:* Tourdaten nach der Einbindung in die *toureninfo.js* - Datei

*Quelle:* Screenshot aus eigener Datei



### 2.2.3 **salzBike.js**

Im oberen Teil der `salzBike.js` Datei werden sämtliche als *trailGroup* gpx definierten Dateien als *featureGroup* in die Karte eingebunden. Ebenso wird durch *eleGroup* eine weitere *featureGroup* hinzugefügt. Diese ist für die korrekte Darstellung des Höhenprofils der Strecken verantwortlich. Durch die dritte und letzte Gruppe, der *markerGroup* werden die Start- und Endpunkte der Touren definiert und in die Karte eingefügt. Anschließend werden die benötigten Hintergrundkarten als *myLayers* definiert.

Durch *L.control.elevation* wird das Höhenprofil eingefügt und durch entsprechende Attribute bearbeitet und positioniert. Um die Darstellung der tourenspezifischen Daten in der Tabelle zu gewährleisten, wird die Funktion *loadTrack* benutzt um auf diese zuzugreifen. Dies wird unter *Abb.7* veranschaulicht.

```

72 //Etappeninfo anzeigen
73 document.getElementById("Titel").innerHTML = window.TOURENINFO[track].Titel;
74 document.getElementById("Kurztext").innerHTML = window.TOURENINFO[track].Kurztext;
75 document.getElementById("Tourenbeschreibung").innerHTML = window.TOURENINFO[track].Tourenbeschreibung;
76 //document.getElementById("KoordinatenStart").innerHTML = window.TOURENINFO[track].KoordinatenStart;
77 document.getElementById("Start").innerHTML = window.TOURENINFO[track].Start;
78 //document.getElementById("KoordinatenZiel").innerHTML = window.TOURENINFO[track].KoordinatenZiel;
79 //document.getElementById("Ziel").innerHTML = window.TOURENINFO[track].Ziel;
80 document.getElementById("auf").innerHTML = window.TOURENINFO[track].auf;
81 //document.getElementById("ab").innerHTML = window.TOURENINFO[track].ab;
82 document.getElementById("HoechsterPunkt").innerHTML = window.TOURENINFO[track].HoechsterPunkt;
83 //document.getElementById("NiedrigsterPunkt").innerHTML = window.TOURENINFO[track].NiedrigsterPunkt;
84 document.getElementById("Schwierigkeit").innerHTML = window.TOURENINFO[track].Schwierigkeit;
85 document.getElementById("Kondition").innerHTML = window.TOURENINFO[track].Kondition;
86 document.getElementById("Technik").innerHTML = window.TOURENINFO[track].Technik;
87 document.getElementById("Landschaft").innerHTML = window.TOURENINFO[track].Landschaft;
88 document.getElementById("Laenge").innerHTML = window.TOURENINFO[track].Laenge;
89 document.getElementById("Zeit").innerHTML = window.TOURENINFO[track].Zeit;
90 document.getElementById("Parkplatz").innerHTML = window.TOURENINFO[track].Parkplatz;
91 //document.getElementById("Quelle").innerHTML = window.TOURENINFO[track].Quelle;
92 //document.getElementById("Foto").innerHTML = window.TOURENINFO[track].Foto;

```

*Abbildung 7:*      *Einspeisung der tourspezifischen Informationen*

*Quelle:*            *Screenshot aus eigener Datei*

Der Befehl *hoeHENprofil.addData* sorgt dafür, dass das Höhenprofil als *layer.feature* hinzugefügt wird. Folgend addieren wir über *iconURL* die Start- und Ziel-Icons der Karte, welche zuvor heruntergeladen wurden. Um nun das Höhenprofil gut erkennbar darzustellen, ist es notwendig den verschiedenen Steigungen entsprechende Farbstufen zuzuweisen. Dazu werden die Werte in Gruppen von 5 Grad unterteilt. Mithilfe dieser kann ein genauer Eindruck über die Steigungsverhältnisse gewonnen werden. Der entsprechende Code-Ausschnitt um dies zu erreichen wird in *Abb.8* dargestellt.

```

149 var farbe;
150 switch (true) {
151     case (deg >= 20):
152         farbe = "#ff0000";
153         break;
154     case (deg >= 15):
155         farbe = "#ff3300";
156         break;
157     case (deg >= 10):
158         farbe = "#ff6600";
159         break;
160     case (deg >= 5):
161         farbe = "#ff9933";
162         break;
163     case (deg >= 1):
164         farbe = "#ffcc00";
165         break;
166     case (deg >= -1):
167         farbe = "#ccff99";
168         break;
169     case (deg >= -5):
170         farbe = "#b3ff66";
171         break;
172     case (deg >= -10):
173         farbe = "#99ff33";
174         break;
175     case (deg >= -15):
176         farbe = "#80ff00";
177         break;
178     case (deg >= -20):
179         farbe = "#66cc00";
180         break;
181     case (deg < -20):
182         farbe = "#4d9900";
183         break;
184 }
```

Abbildung 8: Farbzuzuweisung der Steigungswerte

Quelle: Screenshot aus eigener Datei

Um eine ausgewählte Tour und deren unterschiedliche Steigungen optisch darzustellen, wird durch den folgenden Befehl (*Abb.9*) eine Einfärbung der jeweiligen Streckenabschnitte im Sinne der vorherig definierten Farbstufen erzeugt.

```
var firstpolyline = new L.Polyline(pointList, {  
    color: farbe,  
    weight: 5,  
    opacity: 0.7,  
    smoothFactor: 1
```

*Abbildung 9:*      *Darstellung der Steigungen anhand von Farbdifferenzierungen*

*Quelle:*            *Screenshot aus eigener Datei*

Im letzten Teil der Datei wurde mithilfe einer *if*- Schleife die Anzeige der Touren koordiniert. Hiermit können die verschiedenen Touren selektiert werden. Falls keine Strecke ausgewählt wurde wird die Tour: Gaisberg mithilfe von *else()* ausgewählt und angezeigt.

### 3. Fazit

Das Abschlussprojekt stellte eine ausgezeichnete Übung dar, um im Unterricht Behandeltes mit neuem, selbst angeeigneten Wissen zu verknüpfen. Durch ausgedehnte Recherchen in Onlinequellen über Plugins, Layout und verschiedenen Styles, konnten weitere Kenntnisse über die Materie erlangt werden und auf die Basis der Gelernten gebaut werden. Probleme ergaben sich bei der Nutzung des Plugins *omnivore*, da es vorerst nicht ersichtlich war ob es zwingend nötig war es zu laden, da alle Tracks als gpx-Dateien gespeichert wurden. Dies wurde durch mehrmaliges Probieren unter Einbezug der *omnivore*- Beschreibungen (<https://github.com/mapbox/leaflet-omnivore>) gelöst. Beim Designen der Seite über das *stylesheet* waren der Kreativität nahezu keine Grenzen gesetzt. Die fertige Website kann unter folgendem Link gefunden werden:

<https://hackerboysibk.github.io>