

# Protokoll für das Abschlussprojekt Web mapping Kurs 18S



## Mitglieder:

Lisa Helminger, MatNr.: 01215781

Stefanie Mössler, MatNr.: 01140252

Werner Klaus, MatNr.: 01019112

## Betreuer:

Klaus Förster, BSc

Mag. Bernd Öggl

**Web-Adresse :** <https://teamturbo.github.io/>

**Thema:** INNSBRUCK NIGHTLIFE

# 1. Projektanforderungen und -beschreibung

Im Rahmen des Methodenkurses “2018S716409 VU Geoinformatik: Web mapping” sollten die Studierenden sich in Gruppen zusammenfinden um gemeinsam ein Abschlussprojekt durchzuführen. Dafür gab es folgende Vorgaben der Kursleitung:

- jedes Team besteht aus zwei bis drei TeilnehmerInnen
- es sind mindestens drei HTML Seiten zu erstellen und zu verlinken
- mindestens eine dieser Seiten beinhaltet eine Kartenapplikation
- die Kartenapplikation soll mindestens drei Leaflet Plugins verwenden
- CSS Stylesheets und Skripts sind in eigene Dateien auszulagern
- Datenquellen (auch Bilder) sollen frei verfügbar sein (Open data)
- Geodaten sollen im Format GeoJSON oder GPX vorliegen
- die Entwicklung der Applikation erfolgt fortlaufend unter Verwendung von GIT
  - zum gemeinsamen Arbeiten wird ein Organizations account pro Team erstellt
  - alle Team Mitglieder sind als Mitglieder dieser Organization einzutragen
  - der Name der Organization ist frei wählbar
  - innerhalb der Organization wird im Repo ORGNAME.github.io gearbeitet
- damit lautet die öffentlich erreichbare Web-Adresse für alle Ergebnisse des Projekts <https://ORGNAME.github.io/index.html>
- Alle Projektseiten werden hier gelinkt: <https://webmapping18s.github.io/index.html>

Zur Umsetzung des Projektes entschied sich das Team Turbo das innsbrucker Nachtleben mit hilfreichen Tipps und Tricks in eine Karte zu verpacken. Dafür wurden den Anforderungen entsprechend drei HTML Seiten zu den Themen Late Shopping & Late Snacking, Bars & Clubs und Frühstücken am Tag danach erstellt. Weiters wurde eine gemeinsame Karte mit den unterschiedlichen Markern und Darstellungsmöglichkeiten ausgearbeitet welche in einer gemeinsamen index.html Seite eingebettet wurde. Die folgenden Kapitel bieten einen tieferen Einblick in die unterschiedlichen Implementierungsschritte, Datenquellen, verwendete Techniken und Plugins.

## 2. Implementierungsschritte

Die Basis unseres Projektes waren wie bei vielen Open Source Web Karten, die Bibliotheken und Funktionen rund um Leaflet (<https://leafletjs.com/>). Damit lassen sich freie Kartendienste, sowie etliche freie Tools und Plugins nutzen, um eine ansprechende Web Karte zu erstellen.

Als Vorlage haben wir beschlossen, die bereits in der Vorlesung erarbeiteten Grundlagen weiter zu verwenden. So orientiert sich das grundlegende Aussehen der HTML Seiten an der Workload 9 zur Biketour durch Tirol.

## a) Hintergrundkarten und Beschriftungs-Overlay

Als Erstes haben wir die Kartengrundlage in eine neue .js Datei übertragen und Hintergrundkarten (OSM Basemap, Basemap.at und Orthophotohintergrund der basemap.at) hinzugefügt.

```
//Hintergrundkarten
let myLayers = {
  osmlayer: L.tileLayer(
    "https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png", {
      subdomains: ["a", "b", "c"],
      attribution: "Datenquelle: <a href = 'https://www.openstreetmap.org'>openstreetmap.org</a>"
    }),
  geolandbasemap: L.tileLayer("https://{s}.wien.gv.at/basemap/geolandbasemap/normal/google3857/{z}/{y}/{x}.png", {
    subdomains: ["maps", "maps1", "maps2", "maps3", "maps4"],
    attribution: "Datenquelle: <a href = 'https://www.basemap.at'>basemap.at</a>"
  }),
  bmapoverlay: L.tileLayer("https://{s}.wien.gv.at/basemap/bmapoverlay/normal/google3857/{z}/{y}/{x}.png", {
    subdomains: ["maps", "maps1", "maps2", "maps3", "maps4"],
    attribution: "Datenquelle: <a href = 'https://www.basemap.at'>basemap.at</a>"
  }),
  bmaporthofoto30cm: L.tileLayer("https://{s}.wien.gv.at/basemap/bmaporthofoto30cm/normal/google3857/{z}/{y}/{x}.jpg", {
    subdomains: ["maps", "maps1", "maps2", "maps3", "maps4"],
    attribution: "Datenquelle: <a href = 'https://www.basemap.at'>basemap.at</a>"
  })
};
```

Abbildung 1: Implementierung der Hintergrundkarten (Eigene Darstellung)

Zusätzlich zu den Hintergrundkarten wurde ein Beschriftungs-Overlay des Orthophoto-Layers im Karten-Controlbereich zum ein- und ausblenden eingefügt.

```
//Default Hintergrundkarte
myMap.addLayer(myLayers.bmaporthofoto30cm);
//Default Overlaykarte
myMap.addLayer(myLayers.bmapoverlay);

//Kartenansichten schalten und zu Karte hinzufügen(aktivieren)
let myMapControl = L.control.layers({
  "OSM BaseMap": myLayers.osmlayer,
  "BaseMap.at": myLayers.geolandbasemap,
  "Orthofoto": myLayers.bmaporthofoto30cm,
}, {
  "Overlay Beschriftung": myLayers.bmapoverlay,
  "Shops": preshopsMarker,
  "Bars": barMarker,
  "Clubs": clubMarker,
  "LateNight Snacks": latesnacksMarker,
  "Frühstücken": breakfastMarker,
  "Alle Marker": clustergrp,
}, {
  collapsed: true
}).addTo(myMap);
```

Abbildung 2: Implementierung des Beschriftungslayers (Eigene Darstellung)

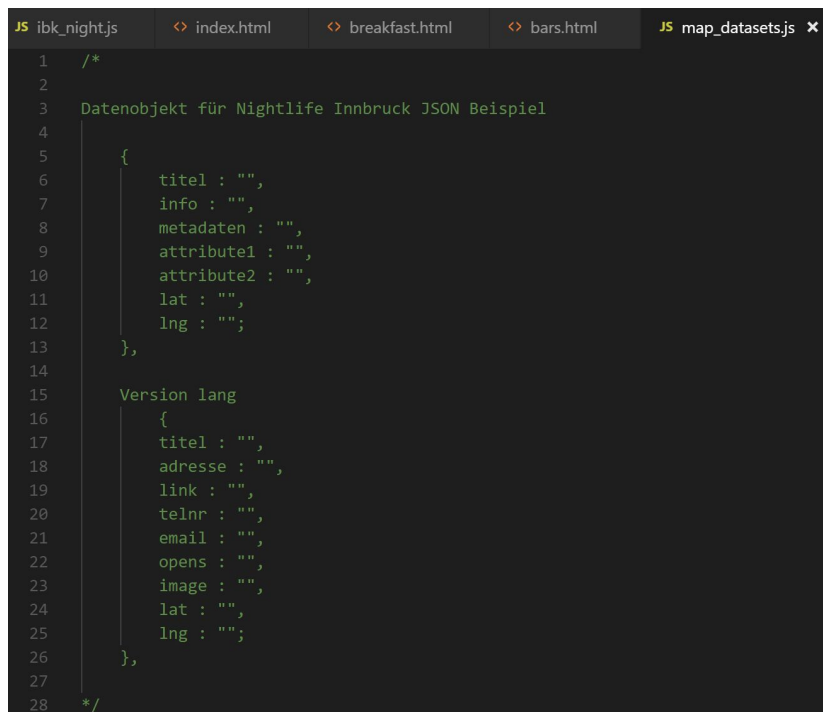
## b) Daten Objekt

Im nächsten Arbeitsschritt haben wir anhand der Zuteilungen der Gruppenmitglieder die darzustellenden Daten erhoben.

Da wir mit Hilfe der frei zur Verfügung stehenden Daten kein passendes Thema finden konnten, haben wir einem freieren Thema entsprechend, Daten selbst erhoben.

Dazu haben wir ein Datenobjekt in einer eigenen Daten.js erstellt und in der index.html eingebunden.

Weiters haben wir uns entschieden, die verschiedenen Themenbereiche zu gliedern, damit wir in der Karte einzeln darauf zugreifen können. Somit sind folgenden Gruppierungen entstanden: Late Night Shops, Late Night Snacks, Bars, Clubs und Breakfast. Zur Vereinfachung haben wir die fünf Objekt Arrays gemeinsam in einer .js erfasst. Mit einer Objektstrukturs Vorlage konnte jedes Teammitglied eigenständig und einfach alle Daten erfassen. Auf diese Datensammlung konnten wir dann ohne großen Aufwand zugreifen um daraus die Marker zu erstellen.



```
JS ibk_nightjs  <> index.html  <> breakfast.html  <> bars.html  JS map_datasets.js x
1  /*
2
3  Datenobjekt für Nightlife Innsbruck JSON Beispiel
4
5  {
6      titel : "",
7      info : "",
8      metadaten : "",
9      attribute1 : "",
10     attribute2 : "",
11     lat : "",
12     lng : "";
13 },
14
15 Version lang
16 {
17     titel : "",
18     adresse : "",
19     link : "",
20     telnr : "",
21     email : "",
22     opens : "",
23     image : "",
24     lat : "",
25     lng : "";
26 },
27
28 */
```

Abbildung 3: Vorlage der Objektstruktur (Eigene Darstellung).

```
//----- BREAKFAST -----
const nightlife_bf_ibkDaten = [{
  titel: "BreakfastClub",
  adresse: "Maria-Theresien-Straße 49",
  link: "https://www.breakfast-club.at/",
  telnr: "+43 512 307959",
  email: "kontakt@breakfast-club.at",
  opens: `Essenszeiten:
um 7:30 kommen wir für EUCH,
ab 7:45 bewirten wir EUCH,
ab 8:00 kochen wir für EUCH,
bis 15:00 kochen wir für EUCH,
bis 15:30 bewirten wir EUCH,
ab 16:00 brauchen WIR Pause`,
  image: "https://www.breakfast-club.at/mediathek/homepage/logo.png",
  lat: 47.2638761,
  lng: 11.3949641,
},
```

Abbildung 4: Beispiel Datenobjekt Frühstück (Eigene Darstellung).

```
//----- PRESHOPS -----
const nightlife_shops_ibkDaten = [{
  titel: "JET Stop & Shop",
  adresse: "Griesauweg 20",
  link: "https://www.jet-tankstellen.at/de/tankstellen/details.php",
  telnr: "+43 512 397729",
  email: "austria@jet-tankstellen.at",
  opens: "Öffnungszeiten: täglich 06:00 - 23:00 Uhr",
  image: "images/jet.jpg",
  lat: 47.1552,
  lng: 11.2615,
},
```

Abbildung 5: Beispiel Datenobjekt Shops (Eigene Darstellung).

```
//----- BARS -----
const nightlife_bars_ibkDaten = [{
  titel: "John Montagu",
  adresse: "Höttinger Gasse 7",
  link: "https://www.facebook.com/Johnmontaguibk/",
  telnr: "+43 699 17296655",
  email: "joshua@johnmontagu.biz",
  opens: `Mo-Sa: 17:30 - 02:00 Uhr,
So: geschlossen`,
  image: "images/JM.jpg",
  lat: 47.2689176,
  lng: 11.3889535,
},
```

Abbildung 6: Beispiel Datenobjekt Bars (Eigene Darstellung).

```
//----- CLUBS -----
const nightlife_clubs_ibkDaten= [{
  titel: "Abyss",
  adresse: "Amraser-See-Str. 5",
  link: "http://www.abyssbar.com/",
  telnr: "+43 512 890143",
  email: "info@abyssbar.com",
  opens: `Mi-Do: 19:30 - 03:00 Uhr,
        Fr-Sa: 19:30 - 06:00 Uhr`,
  image: "images/abyss.jpg",
  lat: 47.261363,
  lng: 11.422251,
},
```

Abbildung 7: Beispiel Datenobjekt Club (Eigene Darstellung).

```
//----- LATENIGHT SNACKS -----
const nightlife_latesnack_ibkDaten = [{
  titel: "Kaiser Stube",
  adresse: "Museumstrasse 31",
  link: "http://kaiserstube.com/",
  telnr: "+43 512 589889",
  email: "office@kaiserstube.com",
  opens: "Öffnungszeiten: täglich 19:00 - 04:00 Uhr",
  image: "images/kaiserstube.jpg",
  lat: 47.1600,
  lng: 11.2404,
},
```

Abbildung 8: Beispiel Datenobjekt Latenight Snacks (Eigene Darstellung).

### c) For-Schleife

Über eine “for-Schleife” für jedes Datenobjekt wurden die einzelnen Marker erstellt und mit einem PopUp versehen. Durch diese Schleife wird die Karte, wie auch die Popups, im Falle einer Erweiterung der Datengrundlage ständig neu erstellt und angezeigt.

```
//Marker Bars und Clubs
for (i = 0; i < nightlife_bars_ibkDaten.length; i++){
  const bars = nightlife_bars_ibkDaten[i];
  L.marker([bars.lat,bars.lng], barsmarkerOptions).addTo(barMarker).bindPopup(`<p><img src=${bars.image}></img> </p>`);
  L.marker([bars.lat,bars.lng], barsmarkerOptions).addTo(barclmarkers).bindPopup(`<p><img src=${bars.image}></img> </p>`);
}
```

Abbildung 9: Erstellung For - Schleife für die Marker und PopUp (Eigene Darstellung).



#### d) Marker Optionen

Über die Marker Optionen hat jedes Daten-Thema ein eigenes Icon zugewiesen bekommen:

```
//Bars und Clubs
const myIconbar = L.icon({
  imageUrl: 'images/icons/bar.png',
  iconAnchor: [16, 37]
});
const myIconclub = L.icon({
  imageUrl: 'images/icons/bar_coktail.png',
  iconAnchor: [16, 37]
});

const barsmarkerOptions = {
  title: "Bars",
  //draggable: true,
  opacity: 0.95,
  icon: myIconbar,
}
const clubsmarkerOptions = {
  title: "Clubs",
  //draggable: true,
  opacity: 0.95
}
```

Abbildung 10 und 11: Icon Objekt und Marker Optionen mit den erstellten Icon Objekten (Eigene Darstellung).

Die Karte wird durch die Funktion `getBounds()` standardmäßig auf alle Marker der ClusterMarker Gruppe beziehend dargestellt.

*(Als Standard Anzeige der Karte verwenden wir die `getBounds()` Funktion, bezogen auf alle Marker, die in einer ClusterMarker Gruppe liegen.)*

```
//Start auf MarkerGruppe
myMap.fitBounds(clustergrp.getBounds());
```

Abbildung 12: Start Kartendarstellung auf eine ClusterGruppe mit allen Markern (Eigene Darstellung).

### 3. Verwendete Techniken

Die wichtigste Technik war unserer Meinung nach, die eigene Erhebung relevanter Daten. Mit Hilfe dieser Sammlung haben wir in weiterer Folge eine Objektsammlung erstellt, auf welche wir einfach und schnell zugreifen konnten. Die Idee dazu haben wir in einer Vorlesung bekommen, bei der uns dieses Vorgehen kurz vorgestellt wurde (siehe Abbildung 3).

Als zweite wichtige Technik haben wir eine Reihe von `for`-Schleifen erstellt, um einzelne Datenobjekte getrennt voneinander behandeln zu können.

Diese Schleifen iterieren das Objekt Array durch alle Einträge durch und erstellen bei jedem Durchlauf den Marker für den jeweiligen Eintrag. Dies hat einerseits den Vorteil einer Vereinheitlichung aller Marker und PopUps und andererseits wird jeder neue Eintrag sofort übernommen und in die Karte integriert.

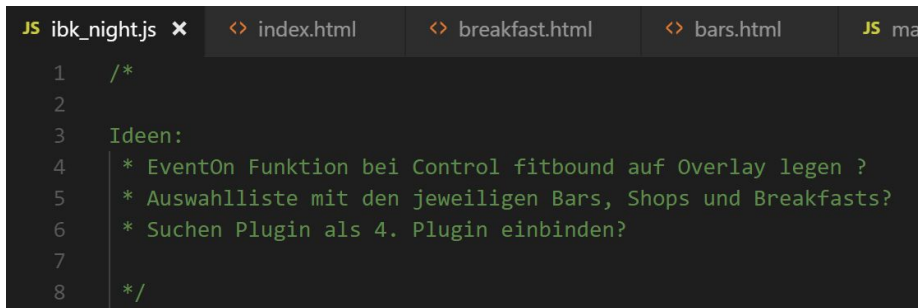
Weiters mussten keine wichtigen oder erwähnenswerten Techniken verwendet werden um die gewünschte Struktur zu erreichen.

### Potentielle Ausbaumöglichkeiten

Im Verlauf des Projektes sind uns noch weitere Ideen gekommen, die wir in einem größeren Projektkontext, als klassisches “Nice-to-have”, bei einer kartenzentrierten Darstellung implementieren hätten können:

- Eine EventOn Funktion beim Legen der Karten-Control, um bei Umschalten der Overlays immer den jeweiligen Ausschnitt zentriert zu haben.
- Eine Auswahlliste mit den jeweiligen Bars, Shops und Breakfasts als alternative Navigation.
- Eine Suchfunktion (Leaflet Search) als Plugin einbinden, um auch innerhalb der Marker nach z.B. Namen oder Adressen suchen zu können.

Diese Ideen wurden von uns am Beginn des Kartenfiles (ibk\_night.js) in einer Kommentarsektion gesammelt.



```
1  /*
2
3  Ideen:
4  * EventOn Funktion bei Control fitbound auf Overlay legen ?
5  * Auswahlliste mit den jeweiligen Bars, Shops und Breakfasts?
6  * Suchen Plugin als 4. Plugin einbinden?
7
8  */
```

Abbildung 13: Gemeinsame Ideensammlung (Eigene Darstellung).



## 4. Datenquelle

Die Daten wurden von uns mittels Internet Recherche und Google Maps selbst erhoben. Diese Daten wurden wie schon in der Abbildung 3 dargestellt, in eine festgelegte Form gebracht.

## 5. Verwendete Plugins

Durch die Anwendung von Open Source Plugins wurden verschiedenen erwünschten Funktionen und Features ermöglicht. Hierfür wurden im Speziellen Plugins der Seite Leaflet.js verwendet. Im folgenden Bereich werden die verwendeten Plugins kurz aufgelistet und beschrieben.

### a) Leaflet MarkerCluster

```
//Cluster Marker für Plugin
const clustergrp = L.featureGroup().addTo(myMap);

//für Plugin MarkerCluster (1)
const barclmarkers = L.markerClusterGroup().addTo(clustergrp);
const clubclmarkers = L.markerClusterGroup().addTo(clustergrp);
const breakfastclmarkers = L.markerClusterGroup().addTo(clustergrp);
const latesnackscmarker = L.markerClusterGroup().addTo(clustergrp);
const shopsclmarker = L.markerClusterGroup().addTo(clustergrp);
```

Abbildung 14: Plugin Leaflet Markercluster (Eigene Darstellung).

Durch die Einbindung dieses Plugins werden je nach Zoomlevel verschiedenen Cluster von Markern vorgenommen.

Verfügbar unter: <https://github.com/Leaflet/Leaflet.markercluster>

### b) Leaflet Fullscreen

```
//Plugin Fullscreen (2)
myMap.addControl(new L.Control.Fullscreen());
```

Abbildung 15: Plugin Leaflet Fullscreen (Eigene Darstellung).

Dieses Plugin beinhaltet eine Fullscreen funktion bei der sich die Karte beim Anklicken des Symbols auf die gesamte Seite ausbreitet.

Verfügbar unter: <https://github.com/Leaflet/Leaflet.fullscreen>

### c) Leaflet Hash

```
//Leaflet Hash Plugin (3)
const hash = new L.Hash(myMap);
```

Abbildung 16: Plugin Leaflet Hash (Eigene Darstellung).

Das Hash Plugin fügt die Koordinaten in die URL ein und sorgt so für eine verbesserte Userfreundlichkeit, da man so den Bildausschnitt als Lesezeichen im Browser speichern kann.

Verfügbar unter: <https://github.com/mlevans/leaflet-hash>

### d) Leaflet Search

```
//Leaflet Search als Control hinzufügen (4)
myMap.addControl(new L.Control.Search({
  layer: clustergrp,
  propertyName: "titel"
}));
```

Abbildung 17: Plugin Leaflet Search (Eigene Darstellung).

Konnte aus vermutlich technischen Gründen trotz mehrfachen Versuchen leider nicht eingebaut werden.

Eine weitere Steigerung der Userfreundlichkeit wurde mit dem Search Plugin, welches eine Suchfunktion im Markerverzeichnis ermöglicht, erreicht. Bei einer großen Anzahl von Lokaltäten erleichtert dies die Suche nach Bestimmten unheimlich.

Verfügbar unter: <https://github.com/stefanocudini/leaflet-search>