

# Vergleich von OpenStreetMap-API vs. Google Maps-API

Marina Schwab      Daria Kern

Sommersemester 2015



Hochschule Aalen  
Fakultät Elektronik und Informatik  
Seminararbeit  
Prof. Dr. Detlef Küpper

# Inhaltsverzeichnis

<b>1</b>	<b>APIs</b>	<b>1</b>
1.1	Allgemein . . . . .	1
1.2	Google Maps . . . . .	2
1.2.1	Google Maps JavaScript API v3 . . . . .	2
1.2.2	Zusätzliche Google Maps APIs . . . . .	3
1.3	OpenStreetMap . . . . .	4
1.3.1	OpenStreetMap OpenLayer API . . . . .	5
1.3.2	Zusätzliche OpenStreetMap APIs . . . . .	5
<b>2</b>	<b>Vergleich von OpenStreetMap-API vs. Google Maps-API</b>	<b>7</b>
2.1	Codevergleich am Biespiel Marker setzen . . . . .	7
2.1.1	Google Maps Codebeispiel . . . . .	7
2.1.2	OpenStreetMaps Codebeispiel . . . . .	8
2.1.3	Auswertung des Codevergleichs . . . . .	8
2.2	Vergleich der Darstellungsmöglichkeiten . . . . .	11
2.2.1	Google Kartenstile . . . . .	11
2.2.2	Google Overlays . . . . .	11
2.2.3	Google Layers . . . . .	12
2.2.4	OpenStreetMaps Overlays . . . . .	12
2.2.5	OpenStreetMap Layers . . . . .	12
2.2.6	Auswertung der Darstellungsmöglichkeiten . . . . .	16
2.3	Vergleich und Auswertung der Lizenztypen . . . . .	17
<b>3</b>	<b>Persönliches Fazit</b>	<b>18</b>
<b>4</b>	<b>Eidesstaatliche Erklärung</b>	<b>19</b>
	<b>Abbildungsverzeichnis</b>	<b>20</b>
	<b>Literaturverzeichnis</b>	<b>25</b>

## Vorwort

“Die größte Sehenswürdigkeit, die es gibt,  
ist die Welt - sieh sie dir an.” – Kurt Tucholsky [Tuc75]

Es wird Vermutet, dass im frühen Stadium der Menschheit schon die ersten Versuche gemacht wurden den Lebensraum darzustellen, indem man Zeichnungen in den Sand malte. [Wikc]

Jetzt, in einer Zeit in der als Hauptinformationsquelle das Internet dient, darf eins nicht fehlen, die Darstellung der Welt Selbst in digitaler Form. Die Erde mit ihrer Fläche von 510 Millionen km<sup>2</sup> sorgt für eine extrem große Menge von Daten, welche verarbeitet und dargestellt werden können. [Wikb] Doch mittlerweile ist die virtuelle Welt zum Greifen nah, auf zahlreichen Webseiten stellen Anbieter ihre Online Maps zur Verfügung. Die bekanntesten Anbieter sind Google Maps, OpenStreetMap, Bing Maps und ArcGIS Online.

Möchte man die Services einer solchen Online Karte in Anspruch nehmen, geschieht das über sogenannte *APIs*.

In unserer Ausarbeitung möchten wir einen Vergleich durchführen, zwischen den *APIs* der Anbieter *OpenStreetMap* und *Google Maps*, welche zur Einbettung einer Weltkarte in einer Website dienen. Wir werden dazu das Setzen eines Markers, die unterschiedlichen Darstellungsweisen durch die Schichten der beiden APIs und die Lizenztypen gegenüberstellen. Abgeschlossen wird jede Gegenüberstellung durch eine separate Auswertung der zusammengetragenen Fakten. Zum Ende unserer Ausarbeitung erfolgt ein persönliches Fazit unsererseits auf Basis der zuvor genannten Auswertungen.

Zur Einführung in den Vergleich erfolgt im weiteren Verlauf eine ausführlichere Erklärung des Begriffs *API* und eine kurze Vorstellung der verwendeten und zusätzlichen *OpenStreetMap-APIs* beziehungsweise *Google Maps APIs*.

## 1 APIs

### 1.1 Allgemein

API ist in der Informatik die Abkürzung für “*Application Programming Interface*” (dt.: Schnittstelle zur Anwendungsprogrammierung).

Eine solche Programmierschnittstelle bietet, meist durch eine Sammlung von Methoden, Zugriff auf ein Softwaresystem und ermöglicht so die Erstellung von Software, ohne genauere Kenntnisse über das Softwaresystem selbst zu benötigen. Der Anwender spricht das Softwaresystem nicht direkt an, sondern interagiert über die API mit ihm.

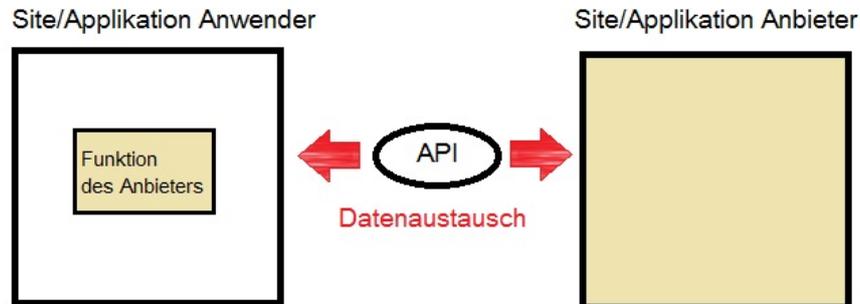


Abbildung 1: Funktionen des Anbieters über API in Website einbinden.

Im Speziellen wird hier auf sogenannte WEB APIs eingegangen. Das sind “[...]Schnittstellen für und von Websites und Webapplikationen.” – [Grü09].

Die API stellt hier die Schnittstelle zwischen zwei Websites/Webapplikationen dar. Sie ermöglicht den Datenaustausch und die Datenverarbeitung zwischen Anbieter und Anwender. So können, über eine API, Funktionen des Anbieters in die eigene Website/Webapplikation mit eingebunden werden. Dies ist in Abbildung 1 dargestellt.

Im folgenden Verlauf dieser Arbeit soll explizit auf die JavaScript basierenden APIs der Anbieter Google Maps und OpenStreetMaps eingegangen werden. Als Zusatz werden noch die Ergänzenden APIs angesprochen, diese werden im Vergleich jedoch nicht berücksichtigt

## 1.2 Google Maps

Google Maps ist seit dem 8. Februar 2005 ein bestehender internetbasierter Kartendienst des Unternehmens Google Inc. Er ist kostenlos und öffentlich zugänglich. Zu Google Maps gibt es eine Reihe unterschiedlicher APIs, die alle unter den Begriff Google Maps API fallen. Im Folgenden soll kurz auf die einzelnen Schnittstellen eingegangen werden. [Wik15b, Rit13, Cona]

### 1.2.1 Google Maps JavaScript API v3

Für die Einbindung benutzerdefinierter und interaktiver Karten in die eine eigene Website ist besonders *Google Maps JavaScript API v3* interessant. Es sind für die Verwendung dieser API JavaScript Kenntnisse von Nöten. Jedoch eignet sie sich besonders gut für den Vergleich mit *OpenStreetMap JavaScript API*. [Conj]

### 1.2.2 Zusätzliche Google Maps APIs

- **Google Maps Road API** : Mit der Hilfe dieser API können Pfade der Straßengeometrie besser angepasst werden. Außerdem lassen sich Metadaten über Straßenabschnitte abfragen. [Dev15, Conn]
- **Google Maps SDK (iOS)/Google Maps Android API v2** : Ermöglicht die Einbettung einer Karte mit Hilfe einer Betriebssystem spezifischen Bibliothek in eine iOS/Android App. [Conh, Coni]
- **Google Places API Web Service** : Diese gibt Informationen über eine Einrichtung, einen Standort oder eine Sehenswürdigkeit zurück. [Conl]
- **Google Places API für iOS / Android** : Hierbei werden Informationen über eine Einrichtung, einen Standort oder eine Sehenswürdigkeit an iOS/Android Geräte zurück gegeben. [Conk, Conm]
- **Static Maps API** : Die Static Maps API sorgt dafür, dass eine Goggle Maps Karte als Bild in die eigene Website eingebettet werden kann, ohne dass JavaScript benötigt wird oder ein dynamischer Seitenaufbau stattfindet. [Cono]
- **Street View Image API** : Diese API erzeugt ein Street View Panorama oder Vorschaubild als statisches Bild in die eigene Website eingebettet werden kann, ohne dass JavaScript benötigt wird. [Conp]
- **Geocoding API** : Mit der Geocoding API können Adressen in geographische Koordinaten umgewandelt werden und auch umgekehrt, sodass aus geographische Koordinaten eine Adresse ausgegeben wird. [Conf]
- **Directions API** : Der Nutzen dieser API ist die Erstellung von Wegbeschreibungen, die Ausgangspunkt, Zielort und Zwischenstationen in Form von Textfolgen oder Koordinaten beinhalten. [Conb]
- **Distance Matrix API** : Stellt Entfernungen und Zeit einer Reise in einer Matrix aus Ausgangspunkten und Zielorten dar. [Conc]
- **Google Maps Geolocation API** : Anhand dieser API kann der ungefähre Standort des Anwenders festgestellt werden. [Cong]
- **Elevation API** : Stellt Daten mit den Informationen über die Höhe der Erdoberfläche und des Meeresbodens zur Verfügung. Für den Fall das keine exakte Höhenmessung für den abgefragten Standort verfügbar ist, wird ein Schätzwert zurückgegeben, der sich aus den vier nächsten Standorten ergibt. [Cond]

- **Time Zone API** : Anhand der Time Zone API kann die Zeitzone eines Standortes abgefragt werden, sowie die Zeitverschiebung zur koordinierten Weltzeit. [Conq]
- **Google Maps Embed API** : Ermöglicht die Einbindung von Google Maps in die eigene Website mittels eines HTML Fragments. [Cone]

### 1.3 OpenStreetMap

Die Gründung des Projekts *OpenStreetMap* (kurz: OSM) im Jahr 2004 von Steve Coast basiert auf dem Ziel eine vollständige Weltkarte zu erstellen und frei verfügbar zu machen. Die dazu erhobenen Geodaten werden von der OSM-Community zusammengetragen, diese Mitglieder werden auch *Mapper* genannt. Der Nachteil sind Qualitätsmängel, welche durch die unterschiedlichen Datenquellen entstehen. Von Vorteil ist dennoch das OSM alle Rechte behält und diese lizenzkostenfrei der Öffentlichkeit zur Verfügung stellt. [Sch] Der Verwender darf die Informationen kopieren, weiterverarbeiten, übermitteln und veröffentlichen, dies ist in der *Open Data Commons Open Database Lizenz* (kurz: ODbL) der *OpenStreetMap Foundation* (kurz: OSMF) festgelegt.

Unter Geodaten werden Datendimensionen verstanden, welche in Punkte, Linien, Flächen, Oberflächen und Körper unterteilt werden. Diese müssen in internationalen Normen und Standards modelliert werden. [Wik15a]

Finanziert wird das Projekt durch freiwillige Helfer und gesponserte Server. Die Kosten für Strom, Datenverkehr und ähnlichen Betriebsmitteln für die Server werden von einer englischen Universität übernommen. Ebenfalls erhält die deutsche OSM-Community Spendengelder durch FOSSGIS e.V., welcher das Ziel verfolgt freie geographische Informationssysteme zu fördern und zu verbreiten. [Opeg] [Fos]

Im folgenden Kapitel wird auf die APIs von OpenStreetMap eingegangen. Der Fokus liegt OpenStreetMap OpenLayer API, welche zur Einbettung einer OSM-Karte in einer Webseite dient. Mit dieser API wird auch der spätere Vergleich durchgeführt wird. Auf die weiteren OSM-APIs wird kurz eingegangen.

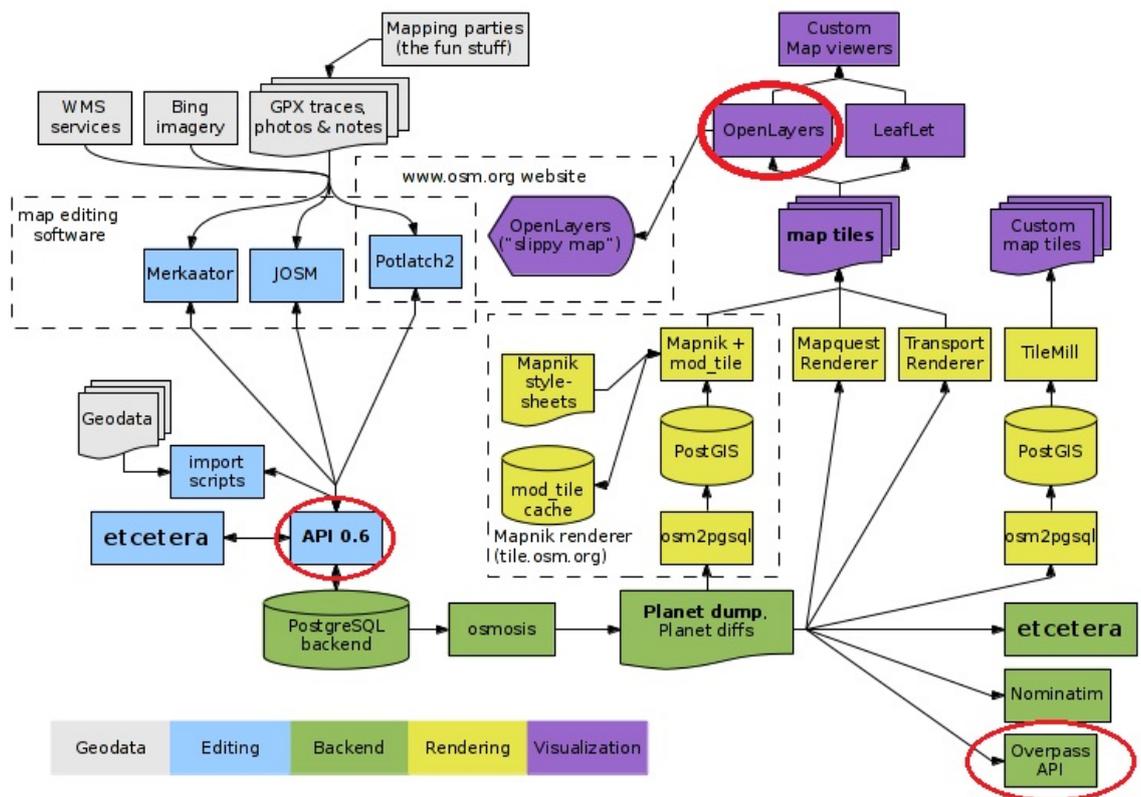
### 1.3.1 OpenStreetMap OpenLayer API

Die OpenStreetMap OpenLayer API ist das gebräuchlichste Werkzeug zur Einbettung einer OSM-Karte in eine Webseite, man spricht dabei auch von einer *Webmap*. Weitere Webmap-APIs mit ergänzenden Funktionen und unterschiedlichen Abhängigkeiten sind vorhanden, welche von ihren Entwicklern frei zur Verfügung gestellt werden. [Opeh]

OSM OpenLayer ist eine JavaScript-Bibliothek und verwendet die *Slippy Map*, diese ist Hauptdarstellungsform der OSM-Karte. Wie der Name OpenLayers schon verrät wird ein Schichtenmodell angewendet, dabei unterscheidet man in sogenannte *Base-Layers* und *Overlay Layers*. Diese zwei Begriffe werden im Kapitel 2.2 näher erläutert.

### 1.3.2 Zusätzliche OpenStreetMap APIs

- **OpenStreetMap API v0.6** : das Lesen, Schreiben und Löschen der Daten der OSM-Karte ist über diese Web-Service-Schnittstelle möglich. Man spricht daher im allgemeinen von der *OSM Editing-API*, beziehungsweise der *API v0.6*, welche seit April 2009 die aktuelle Version ist. [Opec]
- **OpenStreetMap Overpass API** : Die OSM-Overpass API ist eine Read-Only API, welche den Verwendungszweck zum Herunterladen von selektiven Daten aus der OSM-Datenbank hat, die Selektion legt der Anwender explizit fest. Die Mapper legen bei Ihrer Dateneingabe bestimmte Informationen über die bearbeitenden Kartenbereich ein und diese Informationen (Tags, Keys, Objekttypen) können direkt über die Overpass-API angesprochen werden. [Opef]



Quelle: [http://wiki.openstreetmap.org/wiki/Component\\_overview](http://wiki.openstreetmap.org/wiki/Component_overview)

Abbildung 2: Überblick der OpenStreetMap-Komponenten und die Verknüpfung mit den APIs.

## 2 Vergleich von OpenStreetMap-API vs. Google Maps-API

### 2.1 Codevergleich am Biespiel Marker setzen

Im Folgenden werden die Codes von *Google Maps JavaScript API v3* und *OpenStreetMap JavaScript API* hinsichtlich der Marker-Setzung verglichen. Ziel war es jeweils eine Karte zu erzeugen deren Mittelpunkt das Hauptgebäude der Hochschule Aalen ist. Außerdem war ein Marker auf den Campusteil Burren zu setzen. Dazu sind zwei Codeauszüge der verschiedenen API Programmierungen und das jeweilige Ergebnis bereitgestellt.

Im Vergleich fällt als Erstes auf, dass zur Programmierung der OpenStreetMap JavaScript API vier Dateien, eine HTML-, zwei CSS- und eine JavaScript-Datei, notwendig sind, wohingegen bei Google Maps JavaScript API v3 nur eine Datei benötigt wird. Zuerst soll auf den Code für das Einbinden von Google Maps eingegangen werden.

#### 2.1.1 Google Maps Codebeispiel

Bei Google Maps wurde dazu eine HTML Datei erstellt, siehe Abbildung 3. In Zeile 18 und 19 sind die gewünschten Koordinaten als die Variablen *Burren* und *Hauptgeb* deklariert. In der Variable *mapOptions* wird der Kartenzentrum in Zeile 22 auf das Hauptgebäude gesetzt. Der Marker wird in Zeile 26 erstellt und in Zeile 27 wird er auf den Campusteil Burren gesetzt. Anschließend wird er in Zeile 28 zur Karte hinzugefügt.

Erwähnenswert ist, dass sich zusätzlich noch die Möglichkeit bietet, über die *Google Maps Embed API* fast das gleiche Ergebnis ohne JavaScript Kenntnisse zu erzielen. Wie in der Code Abbildung 5 dargestellt, wird dazu ein entsprechend modifizierter in ein HTML-Dokument eingefügt. Hierbei verkürzt sich der Code noch einmal deutlich.

### 2.1.2 OpenStreetMaps Codebeispiel

Für das Setzen eines expliziten Markers in der OpenStreetMap-Variante wird dazu die verwendete HTML-Datei angepasst. Wie in der Abbildung 4 zu sehen ist, wird in Zeile 60 mit der vordefinierten Funktion *addMarker* und den entsprechenden Koordinaten des Campusteils Burren der Marker gesetzt. Der Kartenmittelpunkt wird durch die Eingabe der Koordinaten des Hauptgebäudes in den Zeilen 32 und 33 festgelegt und durch die vordefinierte Funktion *jumpTo* in Zeile 57 erzeugt.

Auch bei der OpenStreetMap OpenLayer API ist es möglich eine *embedded-  
Version* zu generieren wie in der Abbildung 6 zu sehen ist. Der dazu benötigte iFrame wird über die *Teilen-Funktion* auf der Webseite <http://www.openstreetmap.org/> automatisch generiert. Unter dem dort abgebildeten Reiter *Export* kann man den gewünschten Karten-Ausschnitt auswählen, welcher automatisch in den iFrame übernommen wird. [Wro10]

### 2.1.3 Auswertung des Codevergleichs

Bei beiden Anbietern ist es möglich einen Marker in einer Karte zu integrieren, sowie deren Mittelpunkt zu bestimmen. Außerdem bieten beide eine vereinfachte Embed-Version zur Einbindung einer Karte mit Hilfe eines iFrames.

Es wird jeweils ein vorgefertigter Beispielcode von den Anbietern zur Verfügung gestellt, der entsprechend angepasst werden kann. Allerdings ist bei beiden Möglichkeiten der Kodierungsaufwand unter Verwendung der OpenStreetMap im Vergleich zu Google Maps deutlich erhöht. Das liegt unter anderem daran, dass bei der Benutzung der OSM OpenLayer API mehrere Dateien benötigt werden. Ein besteht ein Zusammenspiel innerhalb der HTML-Datei, in welcher die zwei CSS-Dateien und eine JavaScript-Datei verwendet werden.

```

16 var map;
17 function initialize() {
18     var Burren = new google.maps.LatLng(48.84104,10.06706); <!--Variable Burren Koordinaten-->
19     var Hauptgeb = new google.maps.LatLng(48.837771,10.073615); <!--Variable Hauptgebäude Koordinaten-->
20     var mapOptions = {
21         zoom: 16, <!--Zoomstufe 16-->
22         center: Hauptgeb <!--Mitte der Karte beim Hauptgebäude-->
23     };
24     map = new google.maps.Map(document.getElementById('map-hsAalen'),mapOptions);
25
26     var marker = new google.maps.Marker({
27         position: Burren, <!--Marker auf Burren positionieren-->
28         map: map, <!--Marker zur Karte hinzufügen-->
29         title: 'meine Hochschule' <!--Mouseover Effekt-->
30     });
31 }
32
33 google.maps.event.addDomListener(window, 'load', initialize);
34

```

Abbildung 3: Auszug aus dem Code der Google Maps JavaScript API v3.

```

26 function drawmap() {
27
28
29     OpenLayers.Lang.setCode('de');
30
31     // Position und Zoomstufe der Karte
32     var lon = 10.073615; <!--Eingabe des Längengrads vom Hauptgebäude -->
33     var lat = 48.837771; <!--Eingabe des Breitengrads vom Hauptgebäude -->
34     var zoom = 16; <!--Zoomstufe 16-->
35
36     map = new OpenLayers.Map('map', {
37         projection: new OpenLayers.Projection("EPSG:900913"),
38         displayProjection: new OpenLayers.Projection("EPSG:4326"),
39         controls: [
40             new OpenLayers.Control.Navigation(),
41             new OpenLayers.Control.LayerSwitcher(),
42             new OpenLayers.Control.PanZoomBar(),
43         ],
44         maxExtent:
45             new OpenLayers.Bounds(-20037508.34,-20037508.34,
46                                     20037508.34, 20037508.34),
47         numZoomLevels: 16,
48         maxResolution: 156543,
49         units: 'meters'
50     });
51
52     layer_mapnik = new OpenLayers.Layer.OSM.Mapnik("Mapnik");
53     layer_markers = new OpenLayers.Layer.Markers("Address",
54     { projection: new OpenLayers.Projection("EPSG:4326"),
55     visibility: true, displayInLayerSwitcher: false });
56
57     map.addLayers([layer_mapnik, layer_markers]);
58     jumpTo(lon, lat, zoom);
59
60     // Position des Markers
61     addMarker(layer_markers, 10.06706, 48.84104); <!--Marker auf Burren positionieren-->
62 }

```

Abbildung 4: Auszug aus dem Code der OpenStreetMap JavaScript API.

```

1  <!DOCTYPE html>
2  <html>
3  <head>
4  </head>
5  <body>
6  <iframe src="https://www.google.com/maps/embed/v1/place?
7  key=AIzaSyC2vfbDwS0mfFYhvFDzQFoaoLSnNOPH3LQ
8  &q=48.84104,10.06706&center=48.837771,10.073615&zoom=16"
9  width="1000" height="1000"></iframe>
10 </body>
11 </html>

```

Abbildung 5: Code der Google Maps Embed API.

```

1  <html>
2  <head>
3  <title>My first embedded OpenStreetMap page</title>
4  </head>
5  <body>
6  <iframe width="1000" height="900" frameborder="0" scrolling="no"
7  marginheight="0" marginwidth="0"
8  src="http://www.openstreetmap.org/export/embed.html?
9  bbox=10.0644&2C48.8337&2C10.0816&2C48.8425&
10 layer=mappnik&marker=48.84104&2C10.06706"
11 style="border: 1px solid black"></iframe>
12 <br />
13 <small>
14 <a href="http://www.openstreetmap.org/?lat=48.837771&lon=-
15 10.073615&zoom=16&layers=M">View Larger Map</a>
16 </small>
17 </body>
18 </html>

```

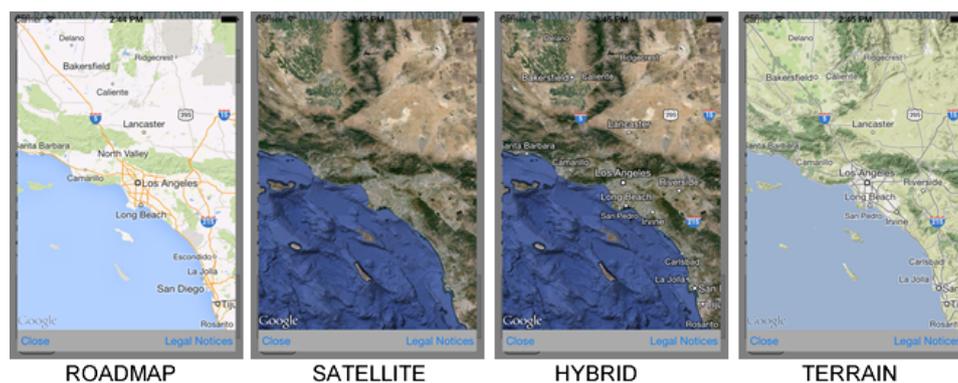
Abbildung 6: Code der OpenStreetMap Embed OpenLayers API.

## 2.2 Vergleich der Darstellungsmöglichkeiten

### 2.2.1 Google Kartenstile

*Kartenstile*, auch bekannt als *Maptypes*, regeln die Darstellung der Karte und bestimmen ihr Aussehen bevor Daten und Informationen eingetragen werden. Einer dieser Kartenstile ist die sogenannte *Roadmap*. Im Beispiel Marker setzen (Kapitel 2.1) wurde die Roadmap verwendet. Sie ist die *Default Einstellung* und gehört zu den vier Standard Kartenstilen, die Google zur Verfügung stellt. Es kann außerdem der Stil *Satellite* gewählt werden, bei dem Satellitenbilder verwendet werden, um die Karte darzustellen. Der Stil *Terrain*, bei dem die Darstellung auf Grundlage von Geländeinformationen erfolgt und dem Stil *Hybrid*, der eine Mischung aus Roadmap und Satellite ist. Beispiele der einzelnen Stile sind in der Abbildung 7 zu sehen.

Neben der Wahl eines der Standardstile, besteht zudem die Möglichkeit eigene Kartenstile zu entwerfen oder die bereits vorhandenen nach eigenen Wünschen zu modifizieren. [Devh]



Quelle: <https://github.com/wf9a5m75/phonegap-googlemaps-plugin/wiki/Map>

Abbildung 7: Beispiele der Google Maps Kartenstile.

### 2.2.2 Google Overlays

Über die jeweiligen Google Maps Karten können *Deckschichten* gelegt werden, um eigene Inhalte einzublenden, sogenannte *Overlays*. Dazu zählen zum Beispiel die gesetzten Marker aus dem Code Beispiel im Kapitel 2.1.

Die API bietet Overlays außerdem für Informationsfenster, Hilfskonturen, Bilder und vorgegebenen, sowie willkürlichen Formen, die durch Vierecke repräsentiert werden. Darüber hinaus werden noch weitere Funktionen zur besseren Benutzerdefinition bereitgestellt. [Sup, Deva]

### 2.2.3 Google Layers

Um Daten auf der Google Maps Karte anzeigen zu können gibt es verschiedene Schichten, sogenannte *Layers*. Die Datenobjekte werden, anders als bei den Overlays, nicht einzeln manipuliert, sondern als eine gemeinsame Schicht behandelt.

Die API stellt elf, teilweise schon eingestellte Schichten zur Verfügung, welche der Karte hinzugefügt werden können. Darunter die *Google Maps Data Layer Schicht*, die zur Speicherung raumbezogener Daten dient. Die Daten können sowohl benutzerdefiniert sein, als auch *GeoJSON* entstammen. Für den Fall, dass die Daten geometrischer Art sind, errechnet die API automatisch entsprechende Marker oder Linien und zeichnet diese in die Karte ein. Die Data Layer ist eine vergleichsweise einfache Methode, um benutzerdefiniert raumbezogene Daten visuell darzustellen.

Zudem gibt es eine Schicht, die in Verbindung mit der Tabellen API *Google Fusion Tables* arbeitet und die Daten aus der Tabelle auf der Mappe sichtbar macht. So können zum Beispiel gesammelte Erdbebendaten als Wärmebild auf einer Karte dargestellt werden, wie in der Abbildung 8 zu sehen ist.

Während die *Fusion Table Layer Schicht* serverseitig Verwendung findet, gibt es eine weitere, die speziell darauf ausgerichtet ist Wärmebildkarten Client Seitig zu erstellen, die *Heatmap Layer Schicht*.

Es existieren außerdem noch Schichten zur Darstellung von Verkehrsbedingungen, dem öffentlichen Verkehrsnetz oder von Radwegen, sowie eine Schicht, welche die Datenformate *KML* und *GeoRSS* unterstützt.

Der Support für die *Google Maps Engine Layers* endet am 29. Januar 2016. Bereits eingestellt wurden die *Panoramio Layer Schicht* und die *Weather and Cloud Layers Schichten*. [Devf, Devb, Devg, Devc, Devd, Deve]

### 2.2.4 OpenStreetMaps Overlays

In OpenStreetMap ist es möglich durch Overlays ergänzende Informationen in der aktuellen Karte anzuzeigen. Zum Beispiel zu Skipisten, Seefahrtsdaten, Wander- und Radwege und zur Eisenbahninfrastruktur. [Opej] Die OSM lässt auch zu mit Overlays sogenannte Polygonzüge zu erzeugen, das sind Wege aus endlich vielen Geradenstücken. [Blo] Zusätzlich kann durch Overlays Boxen, Marker, Bilder, Text und Vektoren die OSM ergänzt werden. [Opeb]

### 2.2.5 OpenStreetMap Layers

Im Fall von OpenStreetMap werden zahlreiche Schichten als eine OpenSource-Anwendung der Slippy Map Schnittstelle angeboten. [Opej] Die am meisten verwendete Schicht ist die *Mapnik-Layer*, diese wurde auch im Beispielcode

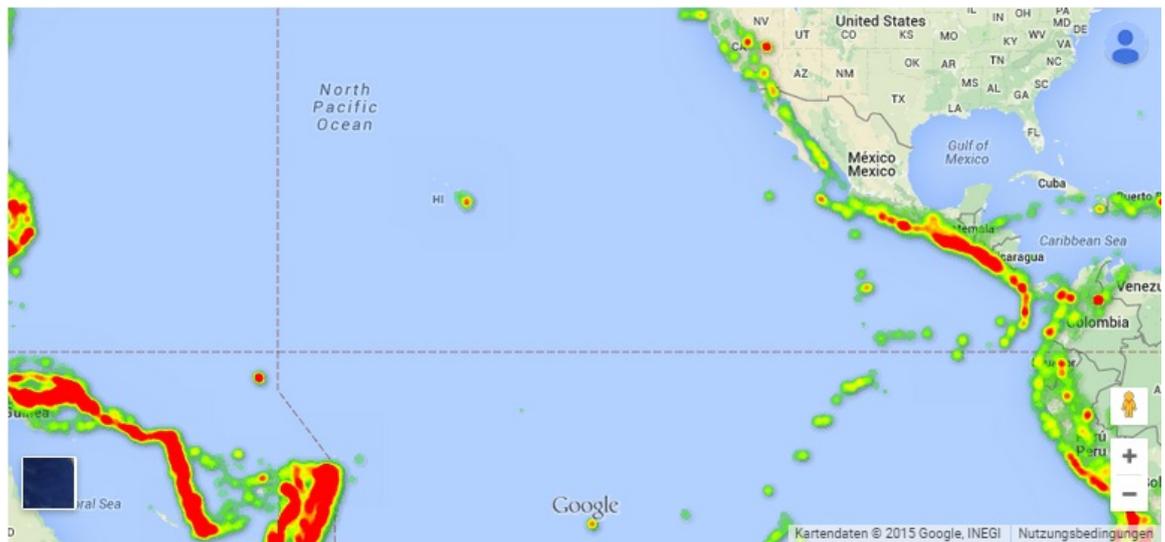


Abbildung 8: Darstellung einer Data Layer am Beispiel von Erdbebendaten als Wärmebild. [Devc]

in Abbildung 4 in der Zeile 51 verwendet. Die Mapnik ist auch in Graustufen darstellbar, diese Schicht nennt sich *OSM B&M-Layer*.

Weiter Darstellungsformen durch Mapnik-Schicht werden unter der Homepage [www.maps.stamen.com](http://www.maps.stamen.com) gezeigt. Zum Beispiel die Version *Toner*, einer schwarz-weißen Karte mit hohem Kontrast oder der Version *Watercolor*, hierbei werden nur die Umrisse der Landschaft im Stil von Wassermalfarben dargestellt.

Für Radfahrer wurde die *OpenCycleMap-Layer* bereitgestellt, diese zeigt speziell die Radwege an, welche von OSM-Mappern eingetragen wurden, zu sehen in der Abbildung 9. [Opea]

In Zusammenarbeit mit der Organisation *Humanitarian OpenStreetMap Team* (kurz: HOT) hat die OSM-Community ein Layer erzeugt, um aktuelle Daten aus Krisengebieten darzustellen, das sogenannte *HOT-Style-Layer*. Die Daten hierfür werden innerhalb von Projekten ausgearbeitet, zum Beispiel über das Erdbeben in Haiti im Januar 2010. [Tea]

Zur Darstellung von öffentlichen Verkehrsmitteln steht die Schicht *Transport-Map-Layer* zur Verfügung. Diese werden in der *Öpnvkarte-Layer* noch ausführlicher dargestellt, indem die einzelnen Verkehrslinien farblich differenziert werden. [Oped] [Opej]

OSM-Standard (Mapnik)



OpenCycleMap

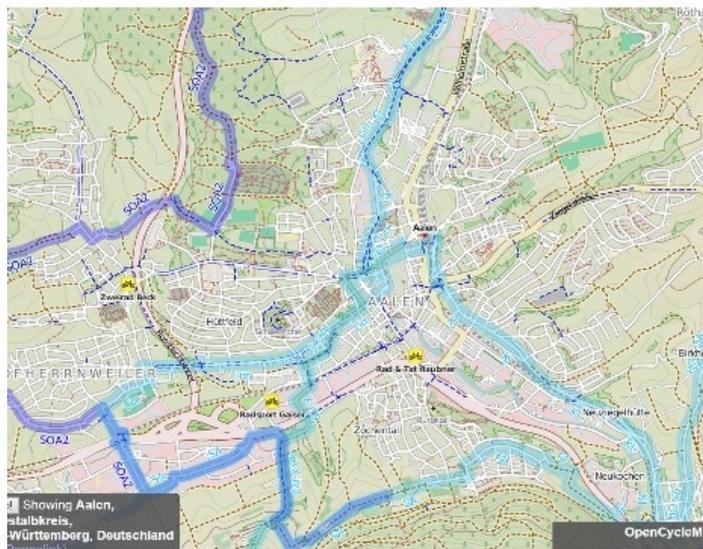


Abbildung 9: Darstellung der Mapnik-Layer und der OpenCycleMap-Layer

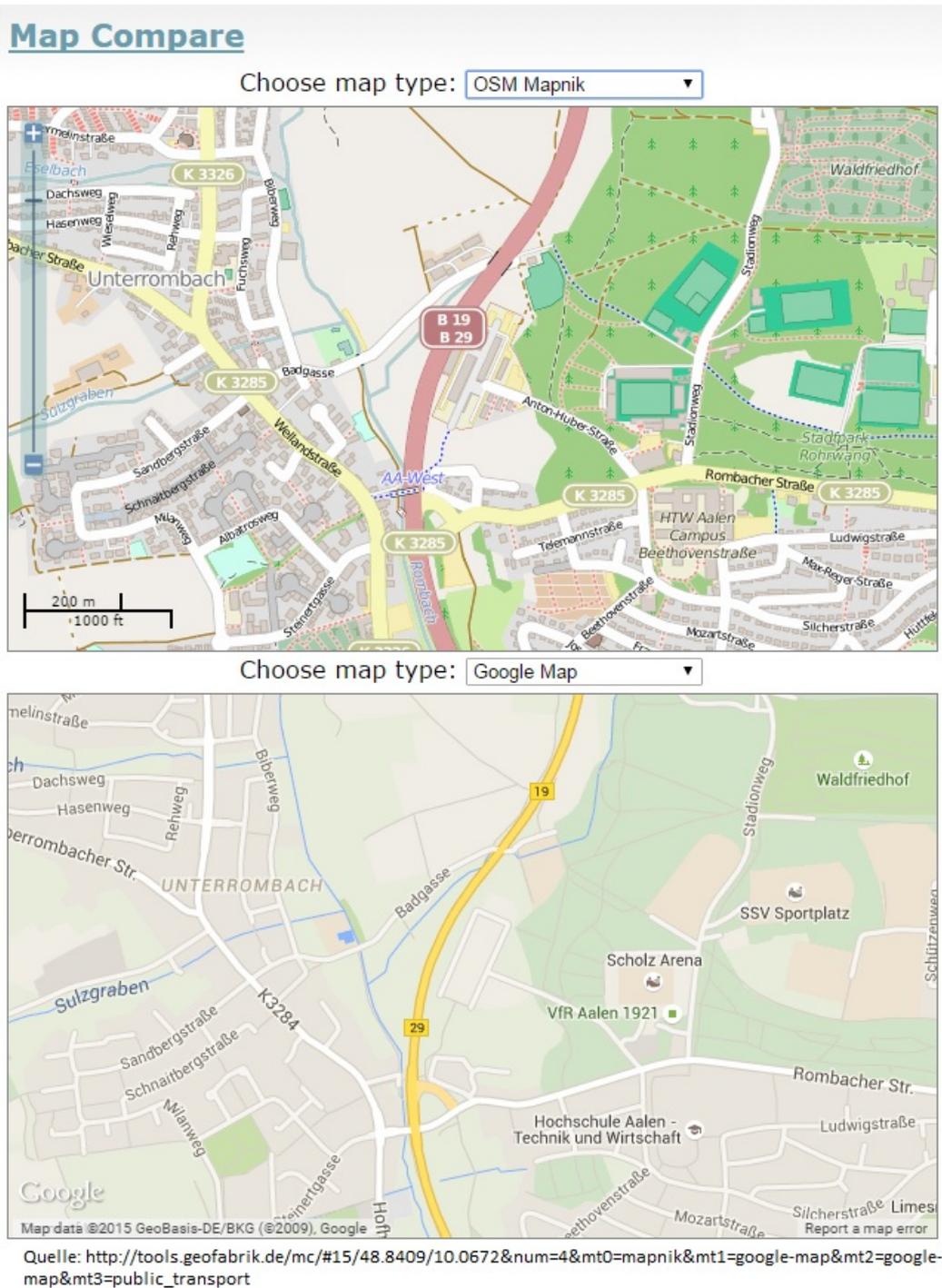


Abbildung 10: Direkter Vergleich der Basiskarten von OpenStreetMap und Google Maps mit Blick auf den Campusteil Burren.

### 2.2.6 Auswertung der Darstellungsmöglichkeiten

Beide APIs bieten mehrere vorgegebene Darstellungsweisen der Karte zwischen denen gewählt werden kann. Die Terrain-Darstellung ist, sowohl bei Google Maps, als auch bei OpenStreetMaps vorhanden. Die Satelliten-Darstellung ist allerdings nur bei Google Maps möglich.

Während Google vier eher sachliche Stile bereit stellt, liegt der Fokus bei OpenStreetMap mehr auf Originalität der Darstellung. Jedoch besteht bei beiden Anbietern die Möglichkeit eigene Kartenstile zu kreieren.

Es sind ähnliche Darstellungsmöglichkeit von Informationen und Daten auf den Karten vorhanden, wie zum Beispiel die Ansicht der Radwegskarte. Polygonzüge sind bei Google Maps und OpenStreetMaps eintragbar.

Bei beiden APIs variiert der Detaillierungsgrad je nach Gebietserforschung, sehr gut zu sehen ist dies in der Abbildung 10. Falls die Information vorhanden ist, wird bei OpenStreetMaps zusätzlich kleine Details eingetragen, wie zum Beispiel Treppen, Schleichwege oder Hydranten.

## 2.3 Vergleich und Auswertung der Lizenztypen

Der Lizenztyp von *Google Maps* ist proprietär, das bedeutet Google Maps steht, anders als OpenStreetMap, unter keiner freien Lizenz. [ubu, Wika]

Da Google Maps häufig auch als öffentliche Recherchen- oder Informationsmaterialquelle verwendet wird, hat Google die Berechtigungen in den Bereichen Fernsehen, Film, Internet, Mobil und Printmedien angepasst. Bei der Verwendung von Google Maps im Fernsehen oder Film wird eine kostenlose *Rundfunklizenz* verlangt, welche mit einem Rundfunklizenzformular bei Google angefordert werden kann.

Bei der Einbettung der Funktionen von Google Maps in eine Webseite mit einer API oder bei der Verwendung mit einem mobilen Endgerät ist keine Berechtigung erforderlich. In allen eben genannten Bereichen muss jedoch der Name Google und deren Datenanbieter deutlich erkennbar sein. [Inca]

Für Printmedien und in internen berufliche Dokumenten ist es ebenfalls erlaubt die Daten von Google Maps zu verwendet, allerdings darf das Erscheinungsbild der Karten nicht verändert werden und es muss ebenfalls die Namensnennung erfolgen.

Es ist explizit verboten ohne eine entsprechende Lizenz, die Daten aus Google Maps als Hauptbestandteil eines Reiseführers einzusetzen. [Incb]

Da im Fall von *OpenStreetMap* die Daten von einzelnen Community Mitgliedern erhoben sind, diese jedoch von OpenStreetMap veröffentlicht werden, muss dies im Bezug auf die Lizenz berücksichtigt werden.

Die Verwendung der Kartenkacheln und Dokumentationen stand vor dem 12. September 2012 unter der Lizenz *Creative Commons Attribution-ShareAlike 2.01* (kurz: CC-BY-SA 2.0) geregelt. Dann wurde eine Lizenzänderung vorgenommen zur *“Open Database License”* (kurz: ODbL). Diese Änderung benötigte die Zustimmung der Eigentümer, um deren Daten unter der neuen Lizenz verbreiten zu dürfen.

Wenn ein Community Mitglied seine Daten auf den OSM-Server ladet bleibt dieser der Urheber der Geodaten, es werden dennoch gewisse Rechte innerhalb der Teilnehmervereinbarung an die OpenStreetMap Foundation übergeben, somit muss bei der Verwendung nicht explizit der Urheber genannt werden.

Die Urheberschaft von OSM muss über die Ergänzung “©OpenStreetMap-Mitwirkende” beziehungsweise auf Englisch mit “©OpenStreetMap contributors” gekennzeichnet werden. [Opei]

Die Unterprojekte und die Verwendung der Software von OSM haben unterschiedliche Lizenzen. [Opee, Opek]

### 3 Persönliches Fazit

Welt ist nicht gleich Welt. Wie wir in unserer Ausarbeitung gezeigt haben, ist es möglich sie in unterschiedlichen Arten und Qualität darzustellen. Daher ist es wichtig den benötigten Verwendungszweck zu kennen, um effizient mit der richtigen Karte arbeiten zu können.

Bei der Arbeit mit der API eines Kartenanbieters gibt es viele Faktoren die beachtet werden müssen, zum Beispiel was die Karte darstellen soll und ob dies bei dem gewählten Anbieter möglich ist. Dies spiegelt die Benutzerfreundlichkeit der API wider. Für Personen ohne Erfahrungen mit APIs kann es schwierig sein deren Verwendung zu verstehen. Da bei Google Maps eine ausführliche Beschreibung der einzelnen APIs mit deren Funktionen vorhanden ist und weniger Code angepasst werden muss, ist dieser Anbieter für Anfänger ohne Programmierkenntnisse geeigneter.

Für fortgeschrittenere API-Verwender gibt es bei Google Maps zahlreiche Möglichkeiten durch Extras und Spielereien die erzeugte Karte individuell zu gestalten. Diese gibt es bei OpenStreetMap auch, jedoch nicht so umfangreich und ausgearbeitet.

Allerdings werden bei OpenStreetMaps die Geodaten ausführlicher erhoben, es wird von der OSM-Community viel Liebe ins Detail gesteckt.

Der Stil von OpenStreetMap spricht uns mehr an, da die Orientierung aufgrund der vielen Details leichter fällt. Alternativ würde das Satelitenbild von Google Maps diese Details ebenso abbilden.

Wir vermuten, dass die Datenqualität bei diesen zwei Anbietern hinsichtlich der Erhebungsmethode teilweise unterschiedlich ausfällt. Bei OpenStreetMaps muss man sich auf die OSM-Community verlassen, wobei Google Maps einen seriöseren Eindruck macht.

Für den Privatgebrauch könnten wir uns vorstellen mit jeder der beiden APIs zu arbeiten. Da unterschiedliche Benutzergruppen, auch unterschiedliche Ansprüche haben, würden wir für ein jüngeres, eher unkonventionelles Publikum OpenStreetMaps empfehlen. Google Maps ist dagegen für ältere Personen, die Wert auf einen vertrauenswürdigen Eindruck einer Website legen, unserer Meinung nach besser geeignet.

Abschließend kann man sagen, dass die Umsetzung einer frei zugänglichen Welt, zumindest online, in beiden Fällen sehr gelungen ist.

## 4 Eidesstaatliche Erklärung

Hiermit versichern wir, Daria Kern und Marina Schwab, die vorliegende Ausarbeitung selbständig und unter ausschließlicher Verwendung der angegebenen Quellen und Hilfsmittel erstellt zu haben.

Die Arbeit wurde bisher keiner anderen Prüfungsbehörde in gleicher oder ähnlicher Form vorgelegt und auch nicht veröffentlicht.

21. Juni 2015

---

Daria Kern

---

Marina Schwab

## Abbildungsverzeichnis

1	Funktion von API . . . . .	2
2	Komponenten von OpenStreetMap . . . . .	6
3	Google Maps JavaScript API v3 Codeauszug . . . . .	9
4	OpenStreetMap JavaScript API Codeauszug . . . . .	9
5	Google Maps Embed API Codeauszug . . . . .	10
6	OpenStreetMap Embed OpenLayers API Codeauszug . . . . .	10
7	Beispiele von Google Maps Kartenstile . . . . .	11
8	Beispiel Google Maps Erdbeben Wärmebild . . . . .	13
9	Screenshots OSM Mapnik und OpenCycleMap . . . . .	14
10	Direkter Vergleich von OSM und Google Maps . . . . .	15

## Literatur

- [Blo] Georgs Blog. Einfache openstreetmap-overlays mit easy-map erstellen. <http://www.georglutz.de/blog/2010/07/26/einfache-openstreetmap-overlays-mit-easymap-erstellen/>. abgerufen am 01.Juni 2015.
- [Cona] Google Developers Console. APIs Library. <https://console.developers.google.com/project/keen-quest-95815/apiui/apis/library>. abgerufen am 01.Juni 2015. Als PDF beigelegt.
- [Conb] Google Developers Console. Directions Backend Overview. [https://console.developers.google.com/project/keen-quest-95815/apiui/apiview/directions\\_backend/overview](https://console.developers.google.com/project/keen-quest-95815/apiui/apiview/directions_backend/overview). abgerufen am 01.Juni 2015. Als PDF beigelegt.
- [Conc] Google Developers Console. Distance Matrix Backend Overview. [https://console.developers.google.com/project/keen-quest-95815/apiui/apiview/distance\\_matrix\\_backend/overview](https://console.developers.google.com/project/keen-quest-95815/apiui/apiview/distance_matrix_backend/overview). abgerufen am 01.Juni 2015. Als PDF beigelegt.
- [Cond] Google Developers Console. Elevation Backend Overview. [https://console.developers.google.com/project/keen-quest-95815/apiui/apiview/elevation\\_backend/overview](https://console.developers.google.com/project/keen-quest-95815/apiui/apiview/elevation_backend/overview). abgerufen am 01.Juni 2015. Als PDF beigelegt.
- [Cone] Google Developers Console. Embedded Backend Overview. [https://console.developers.google.com/project/keen-quest-95815/apiui/apiview/maps\\_embed\\_backend/overview](https://console.developers.google.com/project/keen-quest-95815/apiui/apiview/maps_embed_backend/overview). abgerufen am 01.Juni 2015. Als PDF beigelegt.
- [Conf] Google Developers Console. Geocoding Backend Overview. [https://console.developers.google.com/project/keen-quest-95815/apiui/apiview/geocoding\\_backend/overview](https://console.developers.google.com/project/keen-quest-95815/apiui/apiview/geocoding_backend/overview). abgerufen am 01.Juni 2015. Als PDF beigelegt.
- [Cong] Google Developers Console. Geolocation Overview. <https://console.developers.google.com/project/keen-quest-95815/apiui/apiview/geolocation/overview>. abgerufen am 01.Juni 2015. Als PDF beigelegt.
- [Conh] Google Developers Console. Maps Android Backend Overview. [https://console.developers.google.com/project/keen-quest-95815/apiui/apiview/maps\\_android\\_backend/overview](https://console.developers.google.com/project/keen-quest-95815/apiui/apiview/maps_android_backend/overview). abgerufen am 01.Juni 2015. Als PDF beigelegt.

- [Coni] Google Developers Console. Maps iOS Backend Overview. [https://console.developers.google.com/project/keen-quest-95815/apiui/apiview/maps\\_ios\\_backend/overview](https://console.developers.google.com/project/keen-quest-95815/apiui/apiview/maps_ios_backend/overview). abgerufen am 01.Juni 2015. Als PDF beigelegt.
- [Conj] Google Developers Console. maps.backend. [https://console.developers.google.com/project/keen-quest-95815/apiui/apiview/maps\\_backend/overview](https://console.developers.google.com/project/keen-quest-95815/apiui/apiview/maps_backend/overview). abgerufen am 01.Juni 2015. Als PDF beigelegt.
- [Conk] Google Developers Console. Places Android Overview. <https://console.developers.google.com/project/keen-quest-95815/apiui/apiview/placesandroid/overview>. abgerufen am 01.Juni 2015. Als PDF beigelegt.
- [Conl] Google Developers Console. Places Backend Overview. [https://console.developers.google.com/project/keen-quest-95815/apiui/apiview/places\\_backend/overview](https://console.developers.google.com/project/keen-quest-95815/apiui/apiview/places_backend/overview). abgerufen am 01.Juni 2015. Als PDF beigelegt.
- [Comm] Google Developers Console. Places iOS Overview. <https://console.developers.google.com/project/keen-quest-95815/apiui/apiview/placesios/overview>. abgerufen am 01.Juni 2015. Als PDF beigelegt.
- [Conn] Google Developers Console. Roads Overview. <https://console.developers.google.com/project/keen-quest-95815/apiui/apiview/roads/overview>. abgerufen am 01.Juni 2015. Als PDF beigelegt.
- [Cono] Google Developers Console. Static Maps Backend Overview. [https://console.developers.google.com/project/keen-quest-95815/apiui/apiview/static\\_maps\\_backend/overview](https://console.developers.google.com/project/keen-quest-95815/apiui/apiview/static_maps_backend/overview). abgerufen am 01.Juni 2015. Als PDF beigelegt.
- [Conp] Google Developers Console. Street View Image Backend Overview. [https://console.developers.google.com/project/keen-quest-95815/apiui/apiview/street\\_view\\_image\\_backend/overview](https://console.developers.google.com/project/keen-quest-95815/apiui/apiview/street_view_image_backend/overview). abgerufen am 01.Juni 2015. Als PDF beigelegt.
- [Conq] Google Developers Console. Timezone Backend Overview. [https://console.developers.google.com/project/keen-quest-95815/apiui/apiview/timezone\\_backend/overview](https://console.developers.google.com/project/keen-quest-95815/apiui/apiview/timezone_backend/overview). abgerufen am 01.Juni 2015. Als PDF beigelegt.

- [Deva] Google Developers. Drawing on the Map. <https://developers.google.com/maps/documentation/javascript/overlays>. abgerufen am 02.Juni 2015.
- [Devb] Google Developers. Google Maps JavaScript API - Data Layer. <https://developers.google.com/maps/documentation/javascript/datalayer>. abgerufen am 02.Juni 2015.
- [Devc] Google Developers. Google Maps JavaScript API - Fusion Tables Layer (Experimental). <https://developers.google.com/maps/documentation/javascript/fusiontableslayer>. abgerufen am 02.Juni 2015.
- [Devd] Google Developers. Google Maps JavaScript API - Heatmap Layer. <https://developers.google.com/maps/documentation/javascript/heatmaplayer>. abgerufen am 02.Juni 2015.
- [Deve] Google Developers. Google Maps JavaScript API - KML and GeoRSS Layers. <https://developers.google.com/maps/documentation/javascript/kmlayer>. abgerufen am 02.Juni 2015.
- [Devf] Google Developers. Google Maps JavaScript API - Layers. <https://developers.google.com/maps/documentation/javascript/layers>. abgerufen am 02.Juni 2015.
- [Devg] Google Developers. Google Maps JavaScript API - Maps Engine Layers. <https://developers.google.com/maps/documentation/javascript/mapsenginelayers>. abgerufen am 02.Juni 2015.
- [Devh] Google Developers. Map Types. <https://developers.google.com/maps/documentation/javascript/maptypes>. abgerufen am 06.Juni 2015.
- [Dev15] Google Developers. Snappy travels with the Roads API. <https://www.youtube.com/watch?v=e5YDb-XnDVk>, 2015. abgerufen am 30. Mai 2015.
- [Fos] Fossgis. Fossgis e.V. <http://www.fossgis.de/verein.html>. abgerufen am 31. Mai 2015.
- [Grü09] Gründerszene. Ein nicht-technischer Erklärungsversuch. <http://www.gruenderszene.de/allgemein/web-apis-ein-nicht-technischer-erklarungsversuch>, 2009. abgerufen am 30. Mai 2015.

- [Inca] Google Inc. Berechtigung und Lizenzierung. <http://www.google.de/earth/media/licensing.html>. abgerufen am 07.Juni 2015.
- [Incb] Google Inc. Genehmigungsrichtlinien für Google Maps und Google Earth. <http://www.google.com/permissions/geoguidelines.html>. abgerufen am 21.Juni 2015.
- [Opea] OpenCycleMap. About OpenCycleMap. <http://www.opencyclemap.org/about>. abgerufen am 03.Juni 2015.
- [Opeb] OpenLayers. Layers. <http://docs.openlayers.org/library/layers.html#overlay-layers>. abgerufen am 03.Juni 2015.
- [Opec] OpenStreetMap. API v0.6. [http://wiki.openstreetmap.org/wiki/API\\_v0.6](http://wiki.openstreetmap.org/wiki/API_v0.6). abgerufen am 01.Juni 2015.
- [Oped] OpenStreetMap. DE: Opnvkarte. <http://wiki.openstreetmap.org/wiki/DE:C3%96pnvkarte>. abgerufen am 03.Juni 2015.
- [Opee] OpenStreetMap. DE:Legal FAQ. [http://wiki.openstreetmap.org/wiki/DE:Legal\\_FAQ](http://wiki.openstreetmap.org/wiki/DE:Legal_FAQ). abgerufen am 06.Juni 2015.
- [Opef] OpenStreetMap. DE:Overpass API. [http://wiki.openstreetmap.org/wiki/DE:Overpass\\_API](http://wiki.openstreetmap.org/wiki/DE:Overpass_API). abgerufen am 01.Juni 2015.
- [Opeg] OpenStreetMap. FAQs: Fragen und Antworten. <http://www.openstreetmap.de/faq.html>. abgerufen am 30. Mai 2015.
- [Opeh] OpenStreetMap. Frameworks. <http://wiki.openstreetmap.org/wiki/Frameworks>. abgerufen am 02.Juni 2015.
- [Opei] OpenStreetMap. OpenStreetMap Copyright. <http://www.openstreetmap.org/copyright>. abgerufen am 01.Juni 2015.
- [Opej] OpenStreetMap. OpenStreetMap OpenLayers. <http://wiki.openstreetmap.org/wiki/OpenLayers>. abgerufen am 03.Juni 2015.
- [Opek] OpenStreetMap. Wir möchten die Lizenz ändern .... <http://www.openstreetmap.de/lizenzaenderung.html>. abgerufen am 06.Juni 2015.
- [Rit13] Frank Ritter. Google maps: Das beste Kartenwerk der Welt fuer Android, ios und im Web. <http://www.giga.de/webapps/google-maps/#2227164>, 2013. abgerufen am 30. Mai 2015.

- [Sch] Thomas Schlesinger. OpenStreetMap in R - Freie räumliche Daten für geostatistische Analysen. [http://epub.ub.uni-muenchen.de/12463/1/BA\\_Schlesingert.pdf](http://epub.ub.uni-muenchen.de/12463/1/BA_Schlesingert.pdf). abgerufen am 01.Juni 2015.
- [Sup] Google Support. Eigene Inhalte mit Overlays auf der Karte einblenden. <https://support.google.com/mapmaker/answer/155420?hl=de>. abgerufen am 02.Juni 2015.
- [Tea] Humanitarian OpenStreetMap Team. About humanitarian openstreetmap team. <http://hotosm.org/about>. abgerufen am 03.Juni 2015.
- [Tuc75] Kurt Tucholsky. *Gesammelte Werke*. Verlag Rowohlt, 62. ausgabe edition, 1975.
- [ubu] ubuntuusers.de. Wiki / unfreie Software. [https://wiki.ubuntuusers.de/unfreie\\_Software](https://wiki.ubuntuusers.de/unfreie_Software). abgerufen am 07.Juni 2015.
- [Wika] Wikipedia. Comparison of web map services. [http://en.wikipedia.org/wiki/Comparison\\_of\\_web\\_map\\_services](http://en.wikipedia.org/wiki/Comparison_of_web_map_services). abgerufen am 07.Juni 2015.
- [Wikb] Wikipedia. Erdoberfläche. <http://de.wikipedia.org/wiki/ErdoberflE4che>. abgerufen am 30. Mai 2015.
- [Wikc] Wikipedia. Geschichte der Kartografie. [http://de.wikipedia.org/wiki/Geschichte\\_der\\_Kartografie](http://de.wikipedia.org/wiki/Geschichte_der_Kartografie). abgerufen am 30. Mai 2015.
- [Wik15a] Wikipedia. Geodaten. <http://de.wikipedia.org/wiki/Geodaten>, 2015. abgerufen am 30. Mai 2015.
- [Wik15b] Wikipedia. Google maps. [http://de.wikipedia.org/wiki/Google\\_Maps](http://de.wikipedia.org/wiki/Google_Maps), 2015. abgerufen am 30. Mai 2015.
- [Wro10] Serge Wroclawski. OpenStreetMap for Web Developers. <http://www.emacsen.net/osm/osm-web-tutorial.pdf>, 2010. abgerufen am 02.Juni 2015.