Hochschule Anhalt – Institut für Geoinformation und Vermessung

Projektbericht

Interaktive Karte über den Studienstandort Dessau

Modul „Web Mapping“

Matthias Rüster (4057874)
Paul Schröter (4057873)

4. Semester 13VGI, Juli 2015

Inhaltsverzeichnis

Aufgabenstellung

Im Rahmen der Überarbeitung der Web-Seite des Instituts soll eine interaktive Karte mit Informationsmöglichkeiten für Studierende des Instituts und für potenzielle Studienbewerber erstellt werden. Die interaktive Karte soll über den Studienort informieren und sich vor allem auch an ortsfremde Interessierte wenden.

Die Bearbeitung der Aufgabe erfolgte in 2 Gruppen von jeweils 2 Studierenden.

Ermittlung der Inhalte

Gewählte Themengebiete

Die Themen, die in der Karte dargestellt werden sollen, wurden gemeinsam in beiden Gruppen erarbeitet. Dabei wurde vor allem der Schwerpunkt auf Informationen gelegt, die Studierende interessieren könnte (z.B. Freizeitbeschäftigungen, Einkaufmöglichkeiten, Bars, etc.) und Informationen für Studieninteressierte und Besucher (Hochschulgebäude, Studentenwohnheime, Parkplätze, etc.).

Das ausgearbeitete Konzept für die gewählten Themengebiete zur Darstellung in der Karte ist in Form einer Mind-Map im Anhang zu finden.

Hochschulgebäude sollten möglichst in der Karte hervorgehoben und mit Bildern hinterlegt sein. So können sich Besucher und Studieninteressierte auf dem Campus besser zurechtfinden.

Datenbeschaffung

Die eigentliche Beschaffung der Daten wurden unter den Gruppen aufgeteilt und später zusammengeführt. Die Daten wurden teilweise in freier Textform oder CSV-Dateien [1] gespeichert. Die Daten in freier Textform wurden dann in CSV-Dateien umgeformt. Alle CSV-Dateien wurden dann letztendlich in das GeoJSON-Dateiformat [2] umgewandelt. Die Konvertierungen erfolgten in Bash [3] und mit Hilfe von sed- und awk-Befehlen [4] [5].

Die Umrisse der Hochschulgebäude wurden aus OpenStreetMap-Daten [6] und mithilfe von QGIS [7] in eine GeoJSON-Datei umgewandelt. Zu jedem Hochschulgebäude wurden Fotos im Internet gesucht. Fehlende Bilder von Gebäuden der Hochschule (z.B. Gästehaus) wurden selbst nochmal in der Örtlichkeit aufgenommen.

Wahl des Datenformates

Das von uns gewählte Datenformat ist GeoJSON. Es ist ein offener Standard und dient zur Darstellung von einfachen Geometrien. Als Grundlage für dieses Datenformat dient die JavaScript

Object Notation (JSON) [8]. Eine leichte Einbindung der Daten in JavaScript ist also gegeben.

GeoJSON ist in GIS-Software und vielen Bibliotheken gut unterstützt und benötigt zudem weniger Zusatzdaten zur Beschreibung der Hierarchie der Daten als z.B. KML [9] oder GML [10]. Weitere Vorteile von GeoJSON sind die bessere Lesbarkeit (gegenüber KML, GML oder binären Datenformaten) und die Unabhängigkeit zur Darstellung der Features (keine Styles wie z.B. bei KML benötigt).

Wahl geeigneter Kartensignaturen

Die Erstellung der Icons für die markierten Punkte (Points of Interests – POIs) erfolgte mit dem Online-Dienst „The Map Icons Collection“ [11]. Die dort zusammengestellten und heruntergeladenen Icons unterliegen der Creative Commons Lizenz [12] und sind frei nutzbar.

Die Farben der Symbole sind auf die jeweiligen Themen abgestimmt. So haben beispielsweise alle Icons, die auf sportliche Aktivitäten verweisen eine orangene Farbe.

Lösungsideen

Die POIs sollen durch die gewählten Icons bzw. Marker hervorgehoben werden. Eine Hervorhebung des Instituts für Geoinformation und Vermessung wird durch die Darstellung des Instituts-Logos realisiert. Bei Klick auf einen POI-Marker sollen weitere Informationen zu diesem Standort aufgezeigt werden.

Die Hochschulgebäude werden hervorgehoben um anzuzeigen, dass zu diesen Gebäuden weitere Informationen vorliegen.

Zu Beginn sollte der Kartenausschnitt den gesamten Campus aufzeigen, um einen Überblick zu verschaffen. Als Ausgangseinstellung sollen bereits wichtige Themen als Layer eingeschaltet sein, die für einen Studieninteressierten oder Besucher von Interesse sind.

Eine eingeblendete Mini-Map, wie es beispielsweise bei Google Maps [13] üblich ist, sehen wir als überflüssig an. Die meisten Besucher dieser Internetseite interessieren sich im Grunde nur für den Studienstandort Dessau. Die interaktive Karte soll über den Campus der Hochschule Anhalt und dessen Umgebung informieren. Für andere Fragestellungen, wie z.B. „Wie komme ich nach Dessau (mit Auto, Bahn, …)?“ oder „Wo liegt Dessau?“, werden die Benutzer andere Dienste verwenden mit denen sie vertraut sind. Der Kartenausschnitt kann also auf Dessau begrenzt und ein weites Hinauszoomen auf z.B. ganz Deutschland ignoriert werden.

Die vielen Unterthemen der POIs können zu 6 großen Themen zusammengefasst werden: Hochschule, Täglicher Bedarf, Nahverkehr, Freizeit, Gesundheit und Sonstiges.

Als Grundkarten bieten sich die OpenStreetMap-Karte von Mapbox [14] und die Satelliten-Bilder über die Google Maps API [14] an.

Die Karte sollte möglichst auch für mobile Geräte optimiert sein. Deshalb wären weitere Funktionen, wie z.B. Vollbildmodus, Gerätestandortanzeige oder Ausrichtung der Karte nach dem Kompass des mobilen Gerätes, denkbar.

Das Layout sollte intuitiv bedienbar und besonders für mobile Geräte kompakt gehalten sein. Eventuelle Steuerungsknöpfe müssen groß genug sein, um diese auf kleinen Displays mit dem Finger bedienen zu können.

Abwägung Client- und Server-basierte Lösung

Aufgrund der geringen Datenmengen und Aufrufen der Campus-Seite bietet sich eine clientseitige Lösung an. Die Einrichtung eines Geodaten-Servers für diese geringen Anforderungen ist aus unserer Sicht deshalb nicht notwendig.

Bei einer serverseitigen Lösung käme ein weiterer Administrationsaufwand hinzu. Beide Teile, Webseite mit WMS-Client und WMS-Server, müssten gepflegt werden. Würde man als darunterliegende Datenquelle beispielsweise ein Shapefile [16] wählen, so gäbe es eventuell Probleme bei Umlauten oder bei der Begrenzung der Textlängen auf 255 Zeichen.

Umsetzung mit Leaflet

Die Umsetzung der Ideen erfolgte mit Leaflet. Durch vorhergehende Übungen waren wir bereits mit dieser Bibliothek vertraut. Durch Plugins können dort viele Erweiterungen vorgenommen werden, um schließlich die erforderlichen Funktionen realisieren zu können.

Festlegung der Systemstruktur

Die Inhalte wurden je nach Datentyp und Funktion in einezelne Ordner getrennt. Alle CSS-spezifischen Dateien sind in einem Unterordner zusammengefasst. Gleiches gilt für Bilder und JavaScript-Quellcode. Beim Einfügen einiger Plugins waren zusätzlich noch Dateien in einem Ordner „dist“ erforderlich. Bilder der Hochschulgebäude wurden nochmal separat in einem Unterordner aufgeführt.

Die Geodaten werden im JavaScript-Code durch JSON-Variablen zu Beginn eingelesen. Zum Zeitpunkt der Entwicklung der Karte wurde der Quellcode mit den JSON-Objekten in eine separate Datei ausgelagert, um so ein gleichzeitiges Bearbeiten des Projektes zu ermöglichen. Nach gewissen Arbeitsschritten wurden die Dateien für die Kartengestaltung und die Datei mit den Geodaten untereinander ausgetauscht.

Für die Übersichtlichkeit während des Programmierens der Karte wurden auch Icons und Filterfunktionen in separate JavaScript-Dateien ausgelagert.

Da die JavaScript-Dateien untereinander Variablen-Abhängigkeiten haben (Script X benötigt Variablen aus Script Y um funktionieren zu können) werden für den Produktiveinsatz alle JavaScript-Dateien zusammengeführt. Denn die Reihenfolge des Ladens von JavaScript-Dateien ist von Browser zu Browser unterschiedlich und es würde nur mehr Aufwand bedeuten um diese Abhängigkeiten programmtechnisch auflösen und sichern zu können.

Benutzte Plugins und Bibliotheken

Um die Lösungsideen umzusetzen zu können bedient wir uns einiger Erweiterungen für Leaflet.

jQuery

Die JavaScript-Bibliothek jQuery [17] ist ein Hilfswerkzeug, welches viele nützliche Funktionen zur Manipulation des Document Object Models (DOM) bereitstellt.

Mit Hilfe von jQuery konnten wir beispielsweise realisieren, dass ein Popup-Fenster eines Icons geschlossen wird, sobald gezoomt wurde. Dabei wird ein Element im DOM gesucht, dass die Klasse `leaflet-popup-close-button` besitzt und an dieses Objekt dann ein Click-Event gesendet.

JQuery ist die meistverwendete JavaScript-Bibliothek und wird in jeder zweiten Webseite verwendet [18]. Für spätere Erweiterungen oder kleinere Anpassungen an unserer Umsetzung steht diese Bibliothek also bereits zur Verfügung.

Control.Fullscreen

Für die bessere Handhabung auf mobilen Geräten wird das Plugin Control.Fullscreen [18] genutzt, um ein Vollbildmodus zur Verfügung zu stellen.

Zur Aktivierung des Plugins wird bei der Erstellung der Variable für die Leaflet-Karte der Parameter `fullscreenControl: true` angegeben.

Das Icon für den Button des Vollbildmodus wurde von uns angepasst, sodass die Bedeutung des Buttons eindeutiger für den Benutzer ist.

Leaflet.Zoomhome

Zur besseren Steuerung der Karte wird ein zusätzlicher Home-Button in die ZoomControl-Leiste hinzugefügt [19].

Um das Plugin nutzen zu können muss zunächst die standardmäßige ZoomControl-Leiste von Leaflet durch die Option `zoomControl: false` deaktiviert werden. Danach wird eine

Variable von `L.Control.zoomHome` angelegt und der Karte hinzugefügt (`addTo`).

Als Ergebnis erhält man einen zusätzlichen Button in der Zoom-Steuerung, der den Ausschnitt der Karte auf das Campusgebiet zurücksetzt.

Control.Locate

Für mobile Geräte ist es sinnvoll den eigenen Standort in der Karte anzeigen zu lassen.

Der Button zur Anzeige des Gerätestandorts wird jedoch nur angezeigt, wenn es sich um einen mobilen Browser handelt.

Eingebunden wird das Plugin durch Erzeugung einer Variablen durch `L.control.locate` und das Hinzufügen zur Karte (`addTo`).

In den Optionen des Plugins haben wir eingestellt, dass die Karte zum Standort des Geräts verschoben wird, sobald die Position erfasst wurde (Option: `setView: true`). Der Standort des mobilen Gerätes wird solange auf der Karte verfolgt, bis der Benutzer die Karte manuell verschiebt (Optionen: `follow: true` und `stopFollowingOnDrag: true`). Durch erneutes Klicken auf den Button gelangt der Benutzer wieder zu seinem Standort zurück und die Verfolgung ist wieder aktiv.

Anpassungen für mobile Geräte

Zum Verringern der zu übertragenen Daten an das mobile Gerät kann ein Verkürzen des JavaScript-Codes durch Minify-Programme/Dienste (siehe z.B. [17]) realisiert werden. So kam es in unserem Fall zu einer Ersparnis von ca. 100KB.

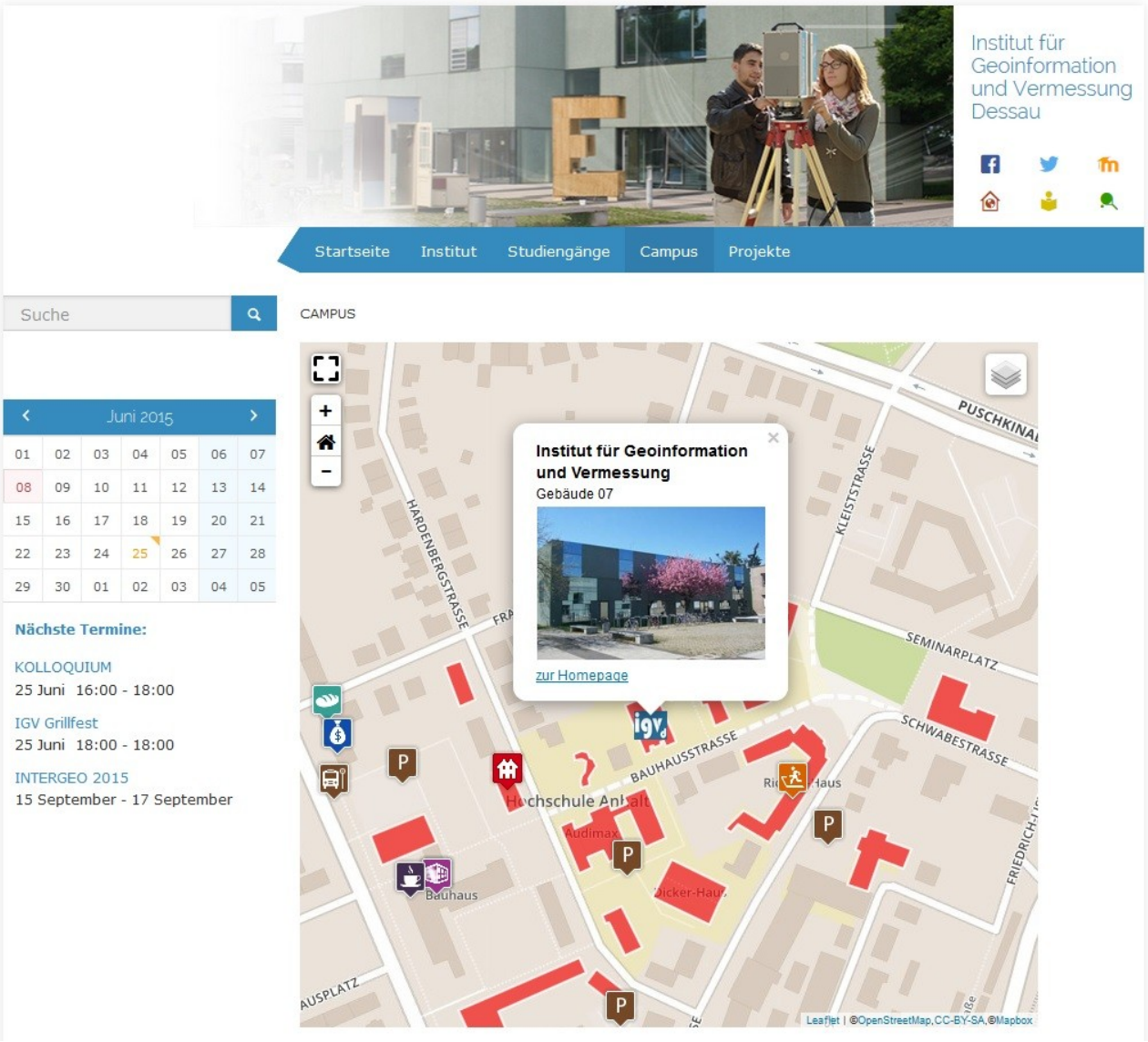
Bilder sollten bereits so zugeschnitten sein, dass diese nicht größer sind als es im HTML-Code vorgesehen ist. Dadurch werden nicht die hochaufgelösten Bilder übertragen und dann runterskaliert, sondern nur die Bilder in der benötigten Auflösung.

Für die mobile Ansicht wird zusätzlich noch eine CSS-Datei geladen, die die Auswahlknöpfe in der Layer-Steuerung größer macht. Ebenso wurde darauf geachtet, dass die Layer-Steuerung nicht zu breit für die kleinen Displays von Smartphones ist.

Wie bereits erwähnt wurden zwei Plugins benutzt, die die Steuerung für mobile Geräte einfacher macht. Dies ist zum einen der Vollbildmodus und zum anderen die Bestimmung des Gerätestandorts. Der Knopf zur Bestimmung des Standortes wird nur angezeigt, wenn der Browser auch zu einem mobilen Gerät gehört.

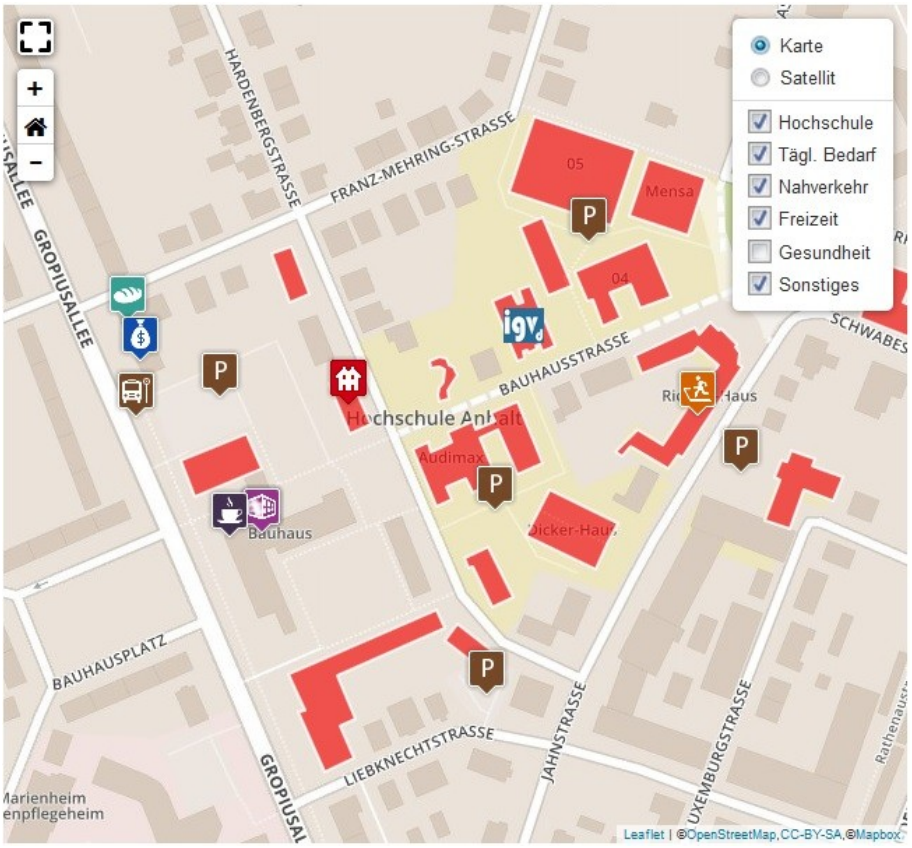
Layout

Die Integration unserer Karte in die vorgesehene Webseitenstruktur sieht folgendermaßen aus:



Zu Beginn wird der Standort des Instituts durch ein Popup hervorgehoben. Zoomt man in der Karte hinaus oder herein, wird das Popup geschlossen.

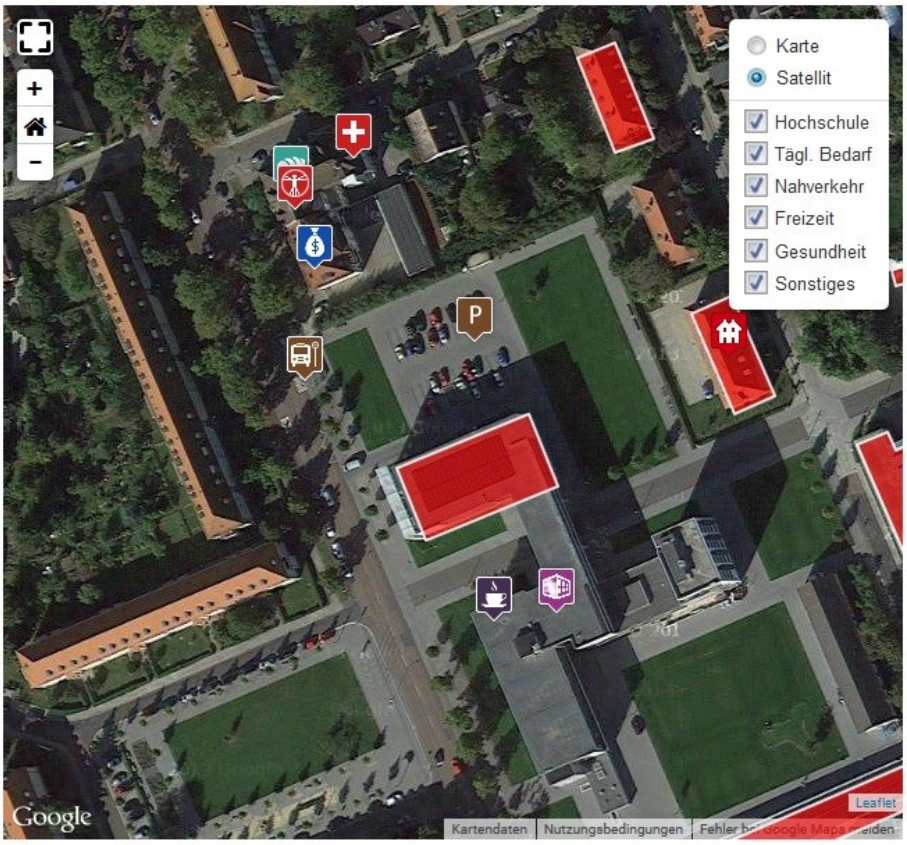
Möchte man noch weitere Layer aus- oder einschalten, so wird die Layersteuerung ausgeklappt:



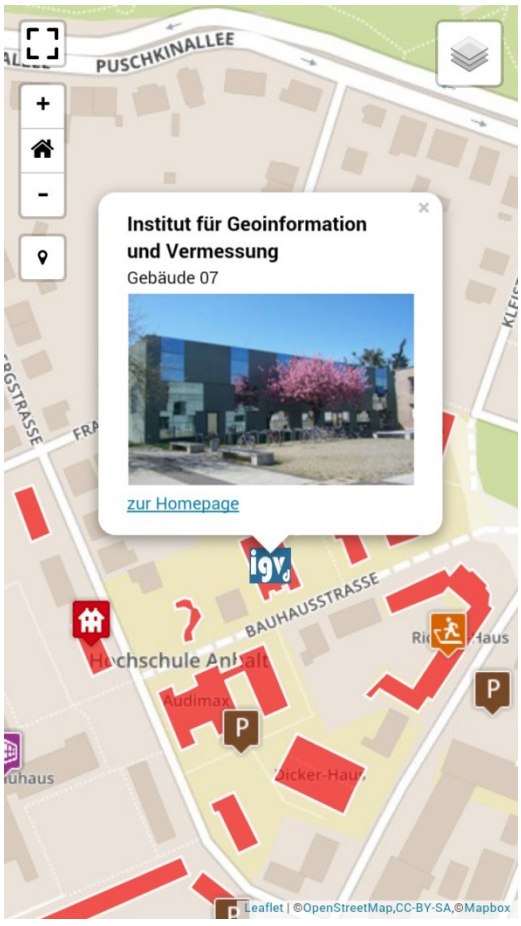
Sollten noch weitere Informationen zu einem Standort verfügbar sein, werden diese in den Popups angezeigt:

Durch das graue Kreuz oben rechts können die Popups geschlossen werden.

Eine Umschaltung auf die Google-Satelliten-Bilder liefert folgende Ansicht:

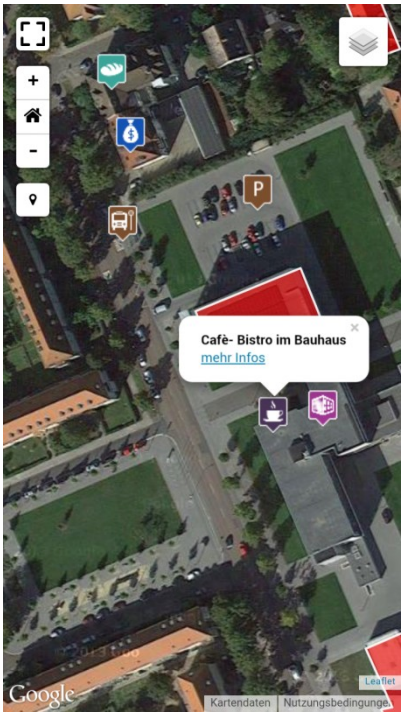


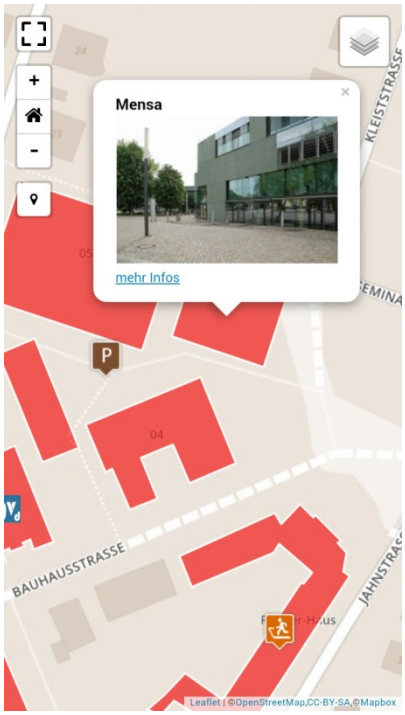
In der mobilen Ansicht sieht die Startseite im Vollbildmodus wie folgt aus:



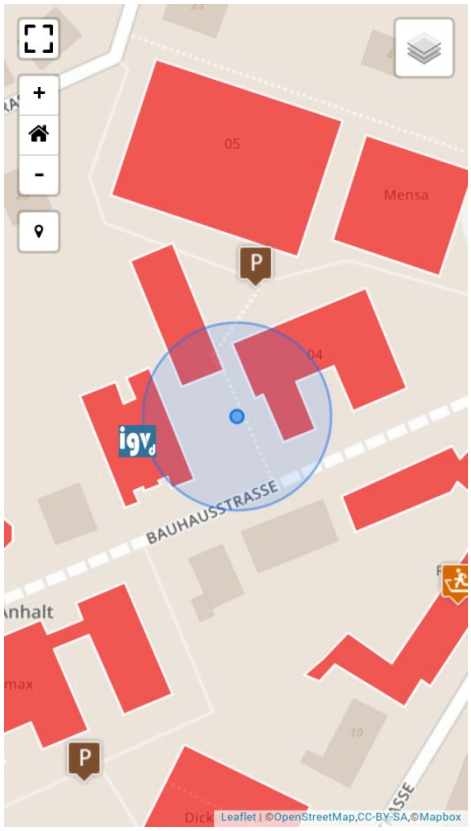
Das Fenster für die Layersteuerung passt auch auf ein kleines Display und die Knöpfe für das Aus- und Abwählen der Layer sind ausreichend groß:

Die Popups von Standorten sind nicht größer als die Displaygröße:





Die Anzeige des Standorts sieht wie folgt aus:



Verbesserungsmöglichkeiten

Leider ist es in Leaflet (noch) nicht möglich die Karte über Touch-Gesten zu drehen oder die Karte über den integrierten Kompass in mobilen Geräten auszurichten. Es gibt lediglich ein Plugin, das ein Kompass-Symbol einblendet, welches sich nach Norden ausrichtet.

In der mobilen Ansicht der Homepage des Instituts fehlt ein kleiner Abstand links oder rechts vom Inlineframe. Dadurch ist es bei kleinen Displays von Smartphones nur schwer möglich an der Karte „vorbeizuscrollen“. Stattdessen wird die Kartenansicht verschoben und man gelangt nicht zum untersten Teil der Webseite.

Die Unterstützung der gängigen Browser im Desktop-Bereich (Firefox, Chrome, IE) ist gegeben. Jedoch funktioniert lediglich beim Internet Explorer (IE) der Vollbildmodus nicht. Ein Test der Webseite mit all Ihren Funktionen konnten wir leider nicht auf Windows-Phones durchführen.

Literaturverzeichnis

[1] „CSV-Standard,“ [Online]. Available: https://tools.ietf.org/html/rfc4180. [Zugriff am 30 06 2015].

[2] „GeoJSON,“ [Online]. Available: http://geojson.org. [Zugriff am 30 06 2015].

[3] „GNU Bash,“ [Online]. Available: https://www.gnu.org/software/bash. [Zugriff am 30 06 2015].

[4] „GNU sed,“ [Online]. Available: http://www.gnu.org/software/sed. [Zugriff am 30 06 2015].

[5] „GNU awk,“ [Online]. Available: https://www.gnu.org/software/gawk. [Zugriff am 30 06 2015].

[6] „OpenStreetMap,“ [Online]. Available: http://www.openstreetmap.de. [Zugriff am 30 06 2015].

[7] „Quantum GIS,“ [Online]. Available: http://www.qgis.org. [Zugriff am 30 06 2015].

[8] „JSON,“ [Online]. Available: http://json.org. [Zugriff am 30 06 2015].

[9] „KML,“ [Online]. Available: https://developers.google.com/kml/documentation. [Zugriff am 30 06 2015].

[10] „GML,“ [Online]. Available: http://www.opengeospatial.org/standards/gml. [Zugriff am 30 06 2015].

[11] „The Map Icons Collection,“ [Online]. Available: http://mapicons.mapsmarker.com. [Zugriff am 30 06 2015].

[12] „Creative Commons,“ [Online]. Available: http://de.creativecommons.org/was-ist-cc. [Zugriff am 30 06 2015].

[13] „Google Maps,“ [Online]. Available: http://www.google.de/maps. [Zugriff am 30 06 2015].

[14] „Mapbox,“ [Online]. Available: https://www.mapbox.com/developers/vector-tiles/. [Zugriff am 01 07 2015].

[15] „Google Maps API,“ [Online]. Available: http://maps.google.com/maps/api/js?v=3.2&sensor=false. [Zugriff am 30 06 2015].

[16] „ESRI Shapefile,“ [Online]. Available: https://www.esri.com/library/whitepapers/pdfs/shapefile.pdf. [Zugriff am 01 07 2015].

[17] „jQuery,“ [Online]. Available: http://jquery.com/. [Zugriff am 01 07 2015].

[18] „W3Techs,“ [Online]. Available: http://w3techs.com/blog/entry/jquery_now_runs_on_every_second_website. [Zugriff am 03 07 2015].

[19] „Leaflet.Control.Fullscreen,“ [Online]. Available: http://github.com/brunob/leaflet.fullscreen. [Zugriff am 02 07 2015].

[20] „Leaflet.Zoomhome,“ [Online]. Available: http://github.com/torfuspolymorphus/leaflet.zoomhome. [Zugriff am 02 07 2015].

[21] „Minify-Onlinedienst,“ [Online]. Available: http://jscompress.com/. [Zugriff am 01 07 2015].