

Web-Entwicklung

Hausarbeit im Sommersemester 2017

Aufgabenstellung

Entwickeln Sie eine Web-Anwendung zur Anzeige von GPS-Tracks. Die GPS-Tracks liegen in Form von Dateien im GeoJSON-Format vor. Wie Sie den beigefügten Beispieldateien entnehmen können, sind das JSON-Dateien, die im Wesentlichen aus einer geordneten Liste von Geokoordinaten bestehen.¹

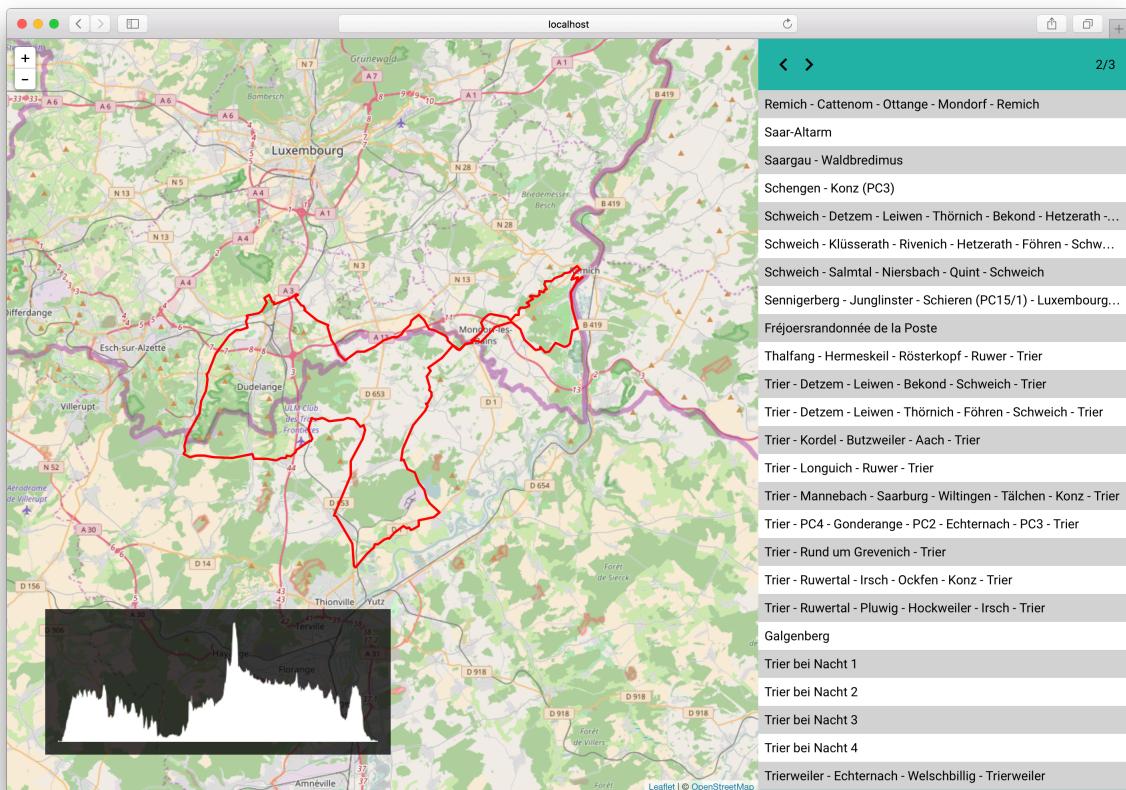


Abbildung 1: Finaler Zustand der clientseitigen Anwendung

Ein mithilfe von Node.js und Express realisierter HTTP-Server soll einerseits die clientseitige Anwendung als statische Dateien an den Browser ausliefern, andererseits die Track-Informationen aus allen GeoJSON-Dateien, die in einem bestimmten Unterordner der serverseitigen Anwendung abgelegt wurden, über eine REST-Schnittstelle zur Verfügung stellen. Das Hochladen, Bearbeiten oder Löschen von Tracks ist nicht vorgesehen.

Die clientseitige Anwendung soll alle vom Server zur Verfügung gestellten Tracks mit den in der Datei hinterlegten Namen auflisten. Nach Auswahl eines Tracks aus dieser Liste soll dieser in einer in die Anwendung integrierte Karte dargestellt und ein Höhenprofil gezeichnet werden. Achten Sie darauf,

¹ Ein kleiner Tipp: Anders als in den meisten anderen Formaten, wird eine Geokoordinate in GeoJSON in der Reihenfolge [Längengrad, Breitengrad, Höhe] und nicht [Breitengrad, Längengrad, Höhe] notiert. Die vollständige Spezifikation finden Sie unter <http://geojson.org/>.

dass der gesamte Track im Kartenausschnitt sichtbar, ausschnittfüllend und zentriert ist. Sie dürfen den Kartendienst bzw. das verwendete npm-Modul selbst wählen.

Die Liste der Tracks soll paginiert werden, d.h. es sollen nur so viele Tracks auf einmal aufgelistet werden, wie bei aktueller Fenstergröße ohne Scrollen in das Browser-Fenster hineinpassen. Über entsprechende Schaltflächen soll zwischen den "Seiten" der Track-Liste gewechselt werden können. Der aktuelle Seitenindex sowie die Gesamtanzahl der Seiten soll angezeigt werden. Wird die Größe des Browser-Fensters durch den Nutzer verändert, soll die Track-Liste neu paginiert werden.

Weitere Anforderungen:

- Es ist Ihnen freigestellt, ob Sie ES6-Features einsetzen oder nicht. Sollten Sie ES6-Features einsetzen, muss Ihre Anwendung zumindest in aktuellen Versionen von Google Chrome und Mozilla Firefox (idealerweise auch von Apple Safari und Microsoft Edge) funktionsfähig sein.
- Es ist Ihnen freigestellt, ob Sie ein MV*-Framework einsetzen oder nicht.
- Achten Sie auf eine sinnvolle Ordnerstruktur und Modularisierung Ihres Projekts mithilfe von CommonJS und Browserify.
- Sie dürfen neben den explizit genannten auch weitere npm-Module einsetzen, insofern diese in den Build-Prozess (s.u.) eingebunden sind.
- Kommentieren Sie Ihren Code sinnvoll.
- Ihr Code darf auf Grundlage der beigefügten ESLint-Konfigurationsdatei keine Fehler aufweisen. Warnungen sollten beachtet werden, führen aber nicht zu einer Abwertung Ihrer Lösung.

Bearbeitung in Teams

Die Bearbeitung der Hausarbeit (sowohl der Studienleistung als auch der Prüfung) erfolgt in Teams mit je zwei Studierenden. Die Zusammensetzung der Teams muss spätestens eine Woche nach Beginn der Bearbeitungszeit durch Eintragung in eine entsprechende Liste festgelegt werden. Die Teilnahme an der Studienleistung und der Prüfung setzt neben der Anmeldung im QIS zusätzlich die Eintragung in diese Liste voraus.

Falls Studierende sich ohne Partner in die Liste eingetragen haben, werden diese zugewiesen.

Studienleistung

Voraussetzung zur Zulassung zur Prüfung ist der Nachweis der Studienleistung. Um diesen zu erbringen muss ein Grundgerüst der Hausarbeit erstellt und fristgerecht abgegeben werden.

Dieses Grundgerüst muss über folgende Komponenten verfügen und die folgenden Anforderungen erfüllen:

- Die clientseitige Anwendung
 - referenziert eine CSS-Datei, die aus einer Less-Datei heraus erzeugt wird,
 - referenziert eine einzige JS-Datei, die das Ergebnis einer Konkatenation mithilfe von Browserify ist,
 - enthält eine Karte, die per JavaScript im Einstiegspunkt der Anwendung zunächst auf die Stadt Trier gezoomt wird,
 - reserviert einen bestimmten Bereich des Browser-Fensters für die Track-Liste.
- Die serverseitige Anwendung
 - startet einen HTTP-Server an einem Port, der als Kommandozeilenargument übergeben werden kann,

- liefert die clientseitige Anwendung als statische Dateien an den Browser aus. Einen npm-Build-Prozess
 - der durch Aufruf von `npm run lint` alle JS-Dateien im Projekt auf Grundlage der beigefügten ESLint-Konfigurationsdatei lintet,
 - der durch Aufruf von `npm run build` das gesamte Projekt erzeugt, dabei CSS- und JS-Dateien minifiziert,
 - der durch Aufruf von `npm run debug` ebenfalls das gesamte Projekt erzeugt, dabei CSS- und JS-Dateien aber nicht minifiziert,
 - der das Erzeugen des Projekts mit `npm run build` oder `npm run debug` abbricht, falls ESLint Fehler findet,
 - der durch Aufruf von `npm run start` oder `npm start` den HTTP-Server an Port 8080 startet,
 - der durch Aufruf von `npm run clean` das Projekt bereinigt, d.h. alle Dateien löscht, welche nachträglich durch den Build-Prozess heruntergeladen oder generiert wurden.

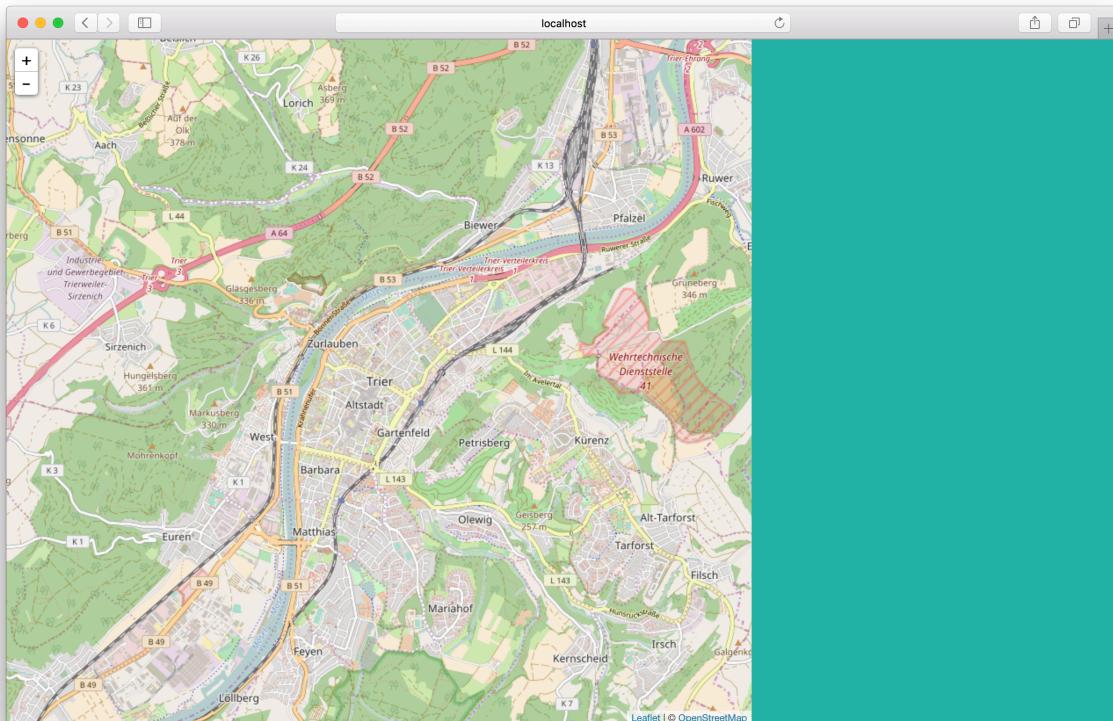


Abbildung 2: Zustand der clientseitigen Anwendung bei Abgabe der Studienleistung

Der Bearbeitungszeitraum für die Studienleistung beginnt mit Veröffentlichung dieser Aufgabenstellung und endet am 12.06.2017. Die Abgabe erfolgt bis zum 12.06.2017 00:00 Uhr MESZ.

Prüfung

Der minimale Funktionsumfang der finalen Abgabe ergibt sich aus der zu Beginn dargestellten Aufgabenstellung sowie den Anforderungen in den Abschnitten "Aufgabenstellung" und "Studienleistung". Gestalterisch sind Sie nicht an die beispielhafte Lösung in den obigen Abbildungen gebunden.

Die Bewertung der Prüfung erfolgt auf Grundlage der Abgabe sowie eines Abnahmegesprächs.

Der Bearbeitungszeitraum für die Prüfung beginnt ebenfalls mit Veröffentlichung dieser Aufgabenstellung und endet am 30.07.2017. Die Abgabe erfolgt bis zum 30.07.2017 00:00 Uhr MESZ.

In der Woche ab dem 31.07.2017 finden die Abnahmegeräte statt. Genaue Termine werden individuell vereinbart. Das Abnahmegeräte muss unabhängig von der Bewertung der Abgabe bestanden werden. Abwesenheit führt zu der Gesamtbewertung „nicht bestanden“.

Abgabe

Die Abgabe (sowohl der Studienleistung als auch der Prüfung) erfolgt via Stud.IP in den entsprechend benannten Ordner der Lehrveranstaltung. Sie umfasst das bereinigte Projekt als ZIP-Datei. Versehen Sie den Dateinamen mit Ihren Rechenzentrumskennungen, um eine eindeutige Zuordnung zu ermöglichen.

Fügen Sie eine zusätzliche Datei in das ZIP-Archiv ein, aus der hervorgeht, in welcher Shell (z.B. Windows cmd.exe, Bash, Cygwin, Git BASH...) Sie den Build-Prozess getestet haben.

Eine fehlende, unvollständige oder verspätete Abgabe führt zur Gesamtbewertung der Studienleistung bzw. der Prüfung mit "nicht bestanden". Maßgeblich ist die Zeitangabe in Stud.IP.