

Web-Entwicklung

Hausarbeit im Sommersemester 2018

Aufgabenstellung

Entwickeln Sie eine Web-Anwendung zur Planung von amateurastronomischen Beobachtungssitzungen. Eine Sitzung weist dabei die folgenden Eigenschaften auf:

- Geplantes Datum
- Ort (Freitext oder GPS-Koordinaten)
- Geordnete Liste der zu beobachtenden Objekte (Freitext)

Der Nutzer soll über die clientseitige Anwendung die folgenden Möglichkeiten haben:

- Sitzungen anlegen und löschen
- Sitzungseigenschaften bearbeiten
- Beobachtungsobjekte zu einer Sitzung hinzufügen und entfernen
- Ort einer Sitzung mithilfe eines externen Kartendienstes¹ in einem neuen Browser-Fenster anzeigen
- Sitzung mit allen Eigenschaften ausdrucken

Ein mithilfe von Node.js und Express realisierter HTTP-Server soll einerseits die clientseitige Anwendung als statische Dateien an den Browser ausliefern, andererseits die Verwaltung der Beobachtungssitzungen über eine REST-konforme HTTP-Schnittstelle zur Verfügung anbieten. Sie können die Technologie zur Persistierung der Daten (z.B. eine relationale oder nicht relationale Datenbank, das Dateisystem, ...) selbst wählen. Sie dürfen von nur einem Nutzer ausgehen, d.h. Sie müssen keinerlei Benutzerverwaltung umsetzen.

Die clientseitige Anwendung soll alle vom Server zur Verfügung gestellten Sitzungen auflisten. Diese Liste soll paginiert werden, d.h. es sollen nur so viele Sitzungen auf einmal aufgelistet werden, wie bei aktueller Fenstergröße ohne Scrollen in das Browser-Fenster hineinpassen. Über entsprechende Schaltflächen soll zwischen den Seiten der 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 Liste neu paginiert werden.

Nach Auswahl einer Sitzung aus dieser Liste sollen die o.g. Funktionen zur Bearbeitung einer Liste (z.B. über ein Formular) bereitgestellt werden.

Weitere Anforderungen sind:

- Ihre Anwendung muss 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 ES6-Features einsetzen oder nicht.
- 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.

¹ z.B. https://tools.wmflabs.org/geohack/geohack.php?language=de¶ms=49.7596_N_6.6439_E

- 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 einzige CSS-Datei**, die aus einer Less-Datei heraus erzeugt wird,
 - referenziert **eine einzige JS-Datei**, die das Ergebnis einer Konkatenation mithilfe von Browserify ist,
 - reserviert einen bestimmten Bereich des Browser-Fensters für die Sitzungsliste,
 - reserviert den restlichen Bereich des Browser-Fensters zur Bearbeitung einer Sitzungsliste.
- 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 sowohl auf unixoiden Betriebssystemen als auch auf Windows-PCs in einer **Bash-Shell** ausgeführt werden kann,
 - 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 durch den Build-Prozess heruntergeladen oder generiert wurden.

Der Bearbeitungszeitraum für die Studienleistung beginnt mit Veröffentlichung dieser Aufgabenstellung und endet am 18.06.2018. Die Abgabe erfolgt bis zum **18.06.2018 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*. Hinsichtlich der Gestaltung sowie der Nutzerinteraktionsmechanismen sind Sie nicht an die beispielhafte Darstellung 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 15.07.2018. Die Abgabe erfolgt bis zum **15.07.2018 00:00 Uhr MESZ**.

In der Woche ab dem 16.07.2018 finden die Abnahmegespräche statt. Genaue Termine werden individuell vereinbart. Das Abnahmegespräch 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 (nicht als RAR-Datei o.ä.)**. Versehen Sie den Dateinamen mit Ihren **rzht-Benutzerkennungen (nicht mit Ihren Matrikelnummern)**, um eine eindeutige Zuordnung zu ermöglichen.

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