

# Abschlussprojekt

Alexander Bazo und Jakob Fehle

## Abschlussprojekt Movie-Explorer

### Aufgabenstellung

Im Rahmen Ihres Abschlussprojektes implementieren Sie eine Webanwendung, die der Darstellung und Exploration einer Film-Datenbank dient. Mit Hilfe der Anwendung sollen NutzerInnen einen Überblick über die in der Datenbank gespeicherten Filme erhalten, neue Einträge anlegen und alle Einträge auf verschiedene Weise visualisieren können. Hierzu soll die Javascript-Bibliothek *Alchemy* verwendet werden, um eine explorierbare Graphen-ähnliche Ansicht aller Filme zu erstellen. Zur Implementierung des Projekts setzen Sie die im Kurs erworbenen Fähigkeiten sowie die dort besprochenen Technologien und Techniken ein. Das Projekt besteht aus zwei Teilen:

1. Der erste Teil liefert die Datenbasis für den zweiten Teil des Projekts. Hierfür sollen Sie mit Hilfe des Skripts *Movie-Explorer Data-Importer* eine Datenbank mit den Filmdaten erstellen. Gestalten Sie ein geeignetes Datenbankschema und halten Sie sich dabei an die im Kurs besprochenen Normalisierungsregeln.
2. Der zweite Teil des Projekts ist die eigentliche Webanwendung, in die Sie die Datenbank aus Teil 1 einbinden und anschließend die in den Anforderungen gelisteten Funktionalitäten umsetzen.

### Vorgaben

Im [Starterpaket](#) finden Sie ein angefangenes Projekt, welches bereits die grobe Ordner- und Dateistruktur der Anwendung vorgibt.

Verwenden Sie das Ihnen zur Verfügung gestellte Starter-Paket, um die Webanwendung (Teil 2 des Projekts) zu implementieren. Das Starter-Paket enthält die Ihnen bekannten node.js Module (`index.js`, `Database.js`) zur Realisierung des Webservers sowie Module, die für die clientseitigen Funktionalitäten der Anwendung zuständig sind (`app.js`, `FetchHelper.js`). Nutzen sie die bestehenden Module und entwickeln sie diese weiter, und erweitern Sie die Anwendung darüber hinaus auch um zusätzliche Module, um Ihre Applikation modularisiert aufzubauen.

Über die bereits existierende Klasse `FetchHelper` können Sie vordefinierte Anfragen (`GET/POST`) an Ihren Webserver senden. Es bleibt Ihnen allerdings offen, ob Sie die Client-seitige Kommunikation mit dem Webserver eigenständig implementieren und/oder die bereits bestehende Klasse verwenden und ggfs. abändern und so auf Ihre Bedürfnisse zuschneiden.

Die zu persistierenden Daten werden in einer SQLite-Datenbank gespeichert. Zur Gestaltung der Benutzeroberfläche verwenden Sie HTML & CSS. Die Graphen-ähnliche Aufbereitung der Datenbank soll mit der Javascript-Bibliothek *Alchemy* gestaltet werden.

## Starten der Anwendung

Um die Anwendung korrekt auszuführen, wird ein lokaler Webserver benötigt, welcher direkt über die *Node.js*-Umgebung gestartet werden kann. Installieren Sie *Node.js* über die entsprechende Installationsdatei für Ihr Betriebssystem, die Sie [hier](#) herunterladen können. Öffnen Sie dann den Projektordner in *Visual Studio Code* und Starten Sie das *integrierte Terminal*. Führen Sie dort den Befehl **npm install** aus, um das Projekt vorzubereiten. Dannach können Sie über die Eingabe des Befehls **npm start** den Server starten und den Client im Browser über die Adresse **http://localhost:8080/** aufrufen. Wenn Sie im integrierten Terminal die Tastenkombination **STRG + C** drücken und die Eingabe bestätigen, wird der Server beendet.

## Anforderungen

### Teil 1: Importer

Importieren Sie **alle** der in *diesem Archiv* bereitgestellten Filmdaten in eine SQLite-Datenbank. Entwerfen Sie dazu zuerst ein passenden Datenbankschema (mit Tabellen für Filme, Genres und SchauspielerInnen), um die Filmeinträge vollumfänglich in der Datenbank zu repräsentieren. Beachten Sie die im Kurs besprochenen Normalisierungsregeln, denken Sie an die Anforderungen, die sich aus Aufgabenteil 2 für das Schema ergeben und vergessen Sie nicht einmalige Identifizierer (IDs) für jeden Film-Eintrag zu vergeben, um diese auch nachträglich noch eindeutig ansprechen zu können. Verwenden Sie für den automatischen Import das Skript, das wir Ihnen *hier* bereitstellen. **Diesen, in JavaScript bzw. für Node.js geschriebenen, Importer müssen Sie an einigen Stellen noch selbständig anpassen.** Der angepasste Skript ist Teil des Projekts und wird zusammen mit der Lösung aus Aufgabenteil 2 eingereicht.

### Teil 2: Webanwendung

Das Hauptziel des Projekts besteht darin, die bereitgestellten Daten über Ihre Datenbank in das System einzugeben und anschließend den Benutzern auf verschiedene visuelle Weise zugänglich zu machen. Ihre Applikation benötigt hierfür mindestens drei Seiten:

1. Eine Startseite, die beim Start Ihrer Anwendung angezeigt wird. Dort sollen die am neuesten zur Datenbank hinzugefügten Filme mit wichtigen Informationen (z.B. Posterbild, Titel, Erscheinungsjahr und Genre) als Überblick in Spalten-/Listenansicht dargestellt werden. Zudem sollen allgemeine Informationen und Metriken bezüglich der in der Datenbank gespeicherten Filme ausgegeben werden (z.B. die Anzahl enthaltener Filme, neu hinzugefügte Filme, das Genre mit den meisten Filmen, der Jahrgang mit den meisten Neuerscheinungen, etc. ).
2. Eine Unterseite, mit dieser NutzerInnen über eine graphische Schnittstelle neue Filmeinträge zur Datenbank hinzuzufügen können. Hierbei kann jeweils der Titel, der Direktor, das Genre, eine beliebige Anzahl an SchauspielerInnen, das Erscheinungsjahr und eine URL zum Filmposter eingegeben werden. Die neuen Einträge werden in der Datenbank, auf verschiedene Tabellen verteilt, persistent gespeichert.
3. Eine Hauptseite, in welcher alle in der Datenbank gespeicherten Daten für eine Exploration aufbereitet und visuell in einer Graphenstruktur angezeigt werden. Verwenden Sie hierfür die Bibliothek *Alchemy*, welche auf Basis vordefinierter Datenstrukturen die Integration

benutzerdefinierter und interaktiver Graphen anbietet. Informationen bezüglich der Implementation können Sie der *Dokumentation* der Bibliothek entnehmen, welche zusätzlich auch *beispielhafte Implementationen* verschiedener Graphenstrukturen aufzeigt.

Innerhalb des Graphen sollen Cluster bezüglich der Genre der Filme erstellt werden. Die SchauspielerInnen der Filme werden einmalig als Datenpunkte dargestellt und mit allen Filmen verbunden, in welchen die SchauspielerInnen mitspielen. Bis auf diese wenigen Vorgaben ist die Umsetzung und Gestaltung des Graphen Ihnen überlassen, achten Sie jedoch auf eine visuell ansehnliche und zugleich informative Darstellungsform. Vergessen Sie nicht, die Eigenschaften des Graphen (z.B. die Farben, Knoten und Kanten) in einem kurzen Text zu erklären.

Um diese Anforderung umzusetzen müssen die in verschiedenen Tabellen gespeicherten Film-Informationen aus der Datenbank extrahiert, zusammengefügt und an den Client übermittelt werden. Damit die Alchemy-Bibliothek daraus einen Graphen erstellen kann, müssen die Daten zusätzlich in JavaScript-/JSON-Objekte mit, von der Bibliothek vorgegebener Objektstruktur, transformiert werden. Die jeweils benötigte Objektstruktur, zusammengesetzt aus den Knoten und Kanten des Graphen, kann der *Dokumentation* oder den *beispielhaften Implementationen* der Bibliothek entnommen werden. Ein Auszug eines solchen Objekts ist im nachfolgenden Code-Snippet zu sehen, welches einen Graphen mit den Dozenten des WebTechnologien-Kurses des SS2021 darstellt:

```
{
  "nodes": [
    {
      "id":1,
      "caption": "Webtechnologien SS2021",
      "root": true
    },
    {
      "id": 2,
      "caption": "Alexander Bazo",
      "type": "lecturer"
    },
    {
      "id": 3,
      "caption": "Jakob Fehle",
      "type": "lecturer"
    },
  ],
  "edges": [
    {
      "source": 1,
      "target": 2,
      "caption": "taught by"
    },
    {
      "source": 1,
      "target": 3,
      "caption": "taught by"
    }
  ]
}
```

```
]
}
```

Die Startseite soll mit beiden Unterseiten verbunden sein. Wie Sie diese Verlinkung realisieren ist Ihnen überlassen, achten Sie allerdings auf eine einleuchtende und intuitive Bedienung, um sowohl von der Startseite auf die Unterseiten gelangen zu können als auch von den Unterseiten zurück zur Startseite wechseln zu können.

Orientieren Sie sich für die Gestaltung der drei Seiten an den im Folgenden dargestellten Skizzen der Benutzeroberfläche.

**Allgemein Gilt: Achten Sie auf bisher gelernte Prinzipien zur Code-Qualität, verteilen Sie Ihren Code entsprechend auf verschiedene Dateien, Methoden und Klassen und kommentieren Sie Ihre Lösung ausreichend.**

### Skizzen der Benutzeroberfläche

Die folgenden Skizzen beschreiben die wesentlichen Bestandteile der Benutzeroberfläche. Bei der Implementierung der Anwendung müssen Sie sich an diesen Vorgaben orientieren. Bitte beachten Sie dabei jedoch, dass die Skizzen lediglich die Struktur und den Inhalt der Benutzeroberfläche beschreiben, nicht aber deren finales oder vollständiges Aussehen.



Figure 1: Startseite

## Movie Explorer

Create New:

Title	<input type="text"/>	
Director	<input type="text"/>	<input type="text"/>
Genre	<input type="text"/>	
Year	<input type="text"/>	
Poster-URL	<input type="text"/>	
Actors	<input type="text"/>	<input type="text"/>
	<input type="text"/>	<input type="text"/>

+

Figure 2: Seite für neue Einträge

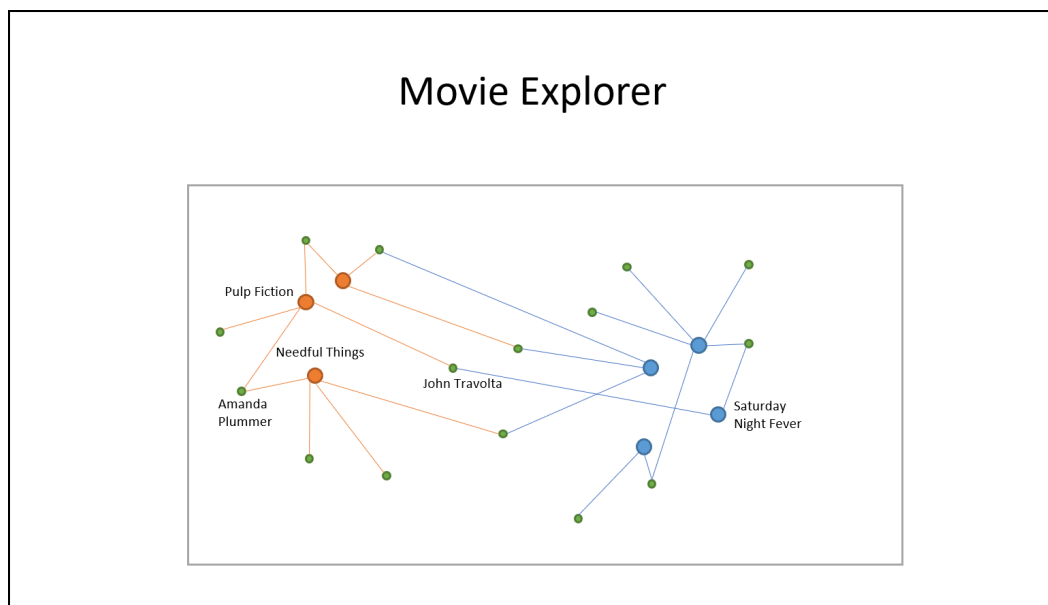


Figure 3: Visueller Graph für Exploration: Orangene und blaue Punkte stehen für Filme der Genres Crime und Drama und grüne Punkte für die SchauspielerInnen der Filme

## Projektablauf

### Hilfestellung und Feedback

Sollten Sie während der Arbeit an Ihrem Projekt Fragen haben oder Hilfestellungen benötigen, können Sie uns per Mail oder Discord kontaktieren. Zusätzlich dazu bieten wir Ihnen folgende Termine an, um Ihnen die Möglichkeit zu geben, den Start und die Finalisierung des Projekts zu besprechen. Wir werden für beide Zeiträume Terminumfragen verschicken und mit Ihnen einen Zeitslot festlegen.

**Zwischen 26. und 30. Juli:** Projekt-Kickoff zum Klären offener Fragen und zum Abstimmen des Vorgehens.

**Mitte September:** Finale Besprechung des aktuellen Stands vor der Abgabe.

### Abgabekriterien

Laden Sie Ihr gesamtes Projekt bis spätestens 30.09.2020 (23:59 Uhr) als zip-komprimierten Ordner auf GRIPS hoch. Denken Sie daran, dass Sie auch mehrfach eine Lösung einreichen können und die vorherige Abgabe damit überschreiben. Bitte reichen Sie bei der Abgabe die erstellte und befüllte Datenbank mit ein. Fügen Sie der Abgabe **nicht** den gesamten JSON-Korpus hinzu. Ihr Abgabe sieht wie folgt aus:

- Teil 1: den Movie-Explorer-Importer mit Ihrer Implementierung.
- Teil 2: Ihre Movie-Explorer Webanwendung inkl. Datenbank.

Der Name der Zip-Datei ergibt sich aus dem Präfix **Projekt\_WT\_SS21**, ihrem Vor- und Nachnamen jeweils getrennt durch **\_**.

Beispiel: **Projekt\_WT\_SS21\_Max\_Mustermann.zip**

### Bewertungskriterien

Wesentliches Kriterium zur Bewertung Ihres Projektes ist die Umsetzung aller beschriebenen Funktionen. Ihre Anwendung muss für die definierten Anforderungen einen ernst gemeinten, erkennbaren Lösungsvorschlag beinhalten. Zusätzlich bewerten wir den Aufwand und die Qualität, die bei der Umsetzung der Funktionen erkennbar sind. Dazu gehört die Verwendbarkeit, Fehlertoleranz und ästhetische Qualität der Benutzeroberfläche sowie die Qualität (Struktur) des Datenbankschemas.

*Wir wünschen Ihnen viel Erfolg!*