

MovieWorld

Miguel Iglesias

Klasse 4I

1 VORAUSSETZUNGEN

Für die Anwendung wird das Backend benötigt. (Siehe Github)

2 EINGESETZTE TOOLS UND LIBRARIES

- **NPM**

Wird als Packet Manger für die Library Abhängigkeiten genutzt für die Laufzeitumgebung

- **WEBPACK**

Wird für die Paketierung der Ressourcen welche am index.html angehängt werden genutzt.

- **MONGODB**

NoSQL DB welche im Backend für die Speicherung der Daten verwendet wird.

- **RIMRAF**

Wird für die Ordnersteuerung der Paketierung eingesetzt

- **JEST**

Wurde nicht eingesetzt

- **TSLINT**

Wird als statische Code Analyse eingesetzt um die Typescript Code Qualität zu prüfen.

- **TYPESCRIPT**

Die Metasprache wird für Programmierung und Logikumsetzung der Single-Page Applikation eingesetzt.

- **JQUERY**

JavaScript Library wird eingesetzt um hauptsächlich DOM-Manipulationen durchzuführen. Zusätzlich bittet es weitere Funktionen welche Steuerungen und Zugriffe vereinfachen.

- **BOOTSTRAP**

Wird für die erstellen des Grundlayouts benötigt sowie für die Komponenten und deren standardisierter Darstellung.

- **LESS**

Wird als Sytlesheetsprache für die CSS Definition der Komponenten eingesetzt.

3 FUNKTIONALITÄT

3.1 FILME SUCHEN

Es können Filme durch die Eingabe eines Textes in der «The movie DB» gesucht werden-

3.2 TOP20 NACH RATING

Es werden die Top20 Filme nach Rating angezeigt. Vordefinierte Abfrage durch «The movie DB»

3.3 TOP20 NACH GENRE

Es könne die Top20 Filme nach Genre angezeigt werden. Die Genre-Auswahl Erfolg durch eine DropDown.

3.4 TOP20 NACH POPULARITÄT

Es werden die Top20 Filme nach Popularität angezeigt. Vordefinierte Abfrage durch «The movie DB»

3.5 BALD ERSCHEINENDE FILME

Es können die bald erscheinende Filme angezeigt werden. Vordefinierte Abfrage durch «The movie DB».

3.6 FAVORITEN VERWALTEN

Die Filme können aus jeder Anzeige zu einer Favoritenliste hinzugefügt und entfernt werden. Die Favoritenliste wird in einem als separate Sicht angezeigt.

3.7 SUCHABFRAGEN ANZEIGEN

Nach jeder manuellen Suchabfrage werden die Informationen abgespeichert und können in einer separaten Sicht angezeigt werden.

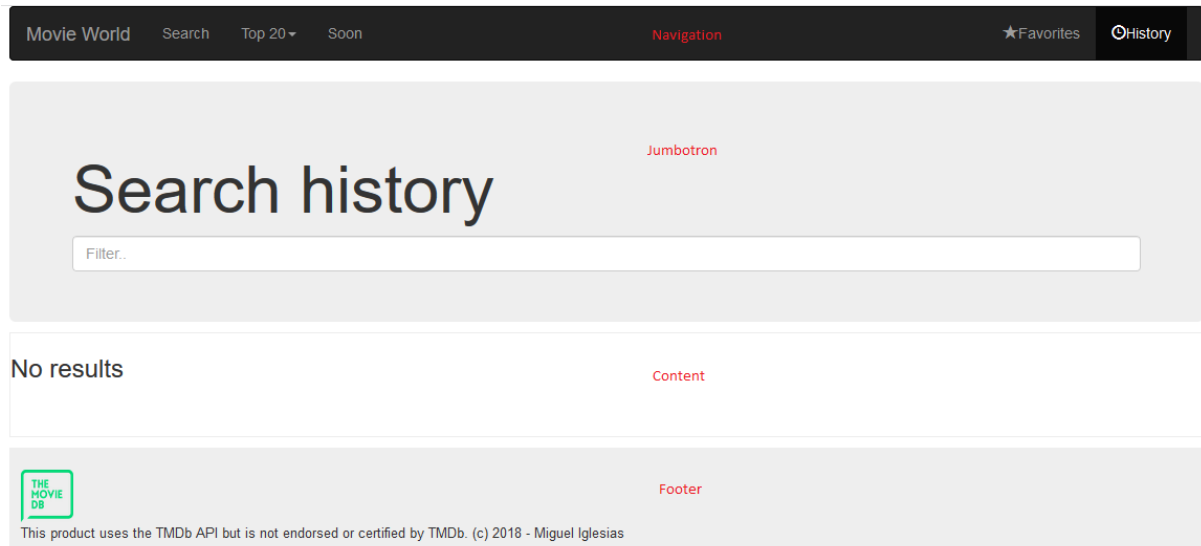
3.8 FILM DETAILS

Die Film Details stehen in jeder Ansicht per Default zur Verfügung. Bei jeder Anzeige werden automatisch die Details des obersten Eintrags dargestellt. Weitere Details können durch das klicken des Filmtitels angezeigt werden.

4 LAYOUT

4.1 AUFBAU

Die Applikation ist in 4 Bereichen unterteilt, die jeweils eine bestimmte Funktion abdecken



4.1.1 Navigation

Wird für die Anzeige sowie Navigierung der Menüs verwendet.

4.1.2 Jumbutron

Dient als Titelanzeige sowie für die zusätzliche Eingabe von Inputdaten wie z.B Genre-DropDown oder Such-Input. Die Eingaben beeinflussen die Resultate, welche im Content dargestellt werden.

4.2 CONTENT

Dien für die Anzeige der Filme, Favoriten und Suchhistory. Die Resultate stellen jeweils zusätzliche Funktionen zur Verfügung wie z.B. Favoriten hinzufügen und entfernen oder Details anzeigen.

4.2.1 Footer

Statische Anzeige für die Darstellung der von «The movie DB» vorausgesetzten Texten und Grafiken. Dies gilt als Vorgabe um das API zu nutzen.

5 APPLIKATION

Die Applikation besteht aus mehreren Dateien, welche die Seiten Logik implementieren:

5.1 APP.TS

Einstiegspunkt der Applikation. Beim Starten werden die Callback für die Routes definiert.

5.2 ROUTING.JS

Implementiert die Route Funktionen. In diesen werden jeweils die entsprechenden Anzeigen für die Generierung aufgerufen. Zusätzlich wird das Rendering der Menus darüber abgebildet.

5.3 CONSTANTS.TS

Ansammlung von Applikationsweiten Konstanten. Beinhaltet die URL für die WS Abfragen. Zusätzlich lässt sich das Mocken des Backends ein- und ausschalten.

5.4 PAGE-CONTROLLER.TS

Implementiert die Logik für das Rendering der einzelnen Anzeigen inkl. Komponenten. Zusätzlich werden alle UI Aktionen implementiert sowie Datenabfragen durchgeführt.

5.5 TEMPLATES.TS

Verwalten von Template Konstanten von Anzeigen und Komponenten.

5.6 MODELS

Es wird jede enthält für jede spezifische Anzeige ein Model inkl. Basisklasse für die Datenhaltung.

5.7 BACKEND

Das Backend stellt GET und POST Methoden zur Verfügung um die Favoriten und Suchabfragen zu persistieren und abzufragen. Für die Datenspeicherung wird die MongoDB verwendet. Die Funktionen stehen über die app.ts zur Verfügung.

6 TODO

6.1 FILTERN DER SUCHHISTORY

Für das filtern der Suchhistory steht bereits eine Eingabemaske zur Verfügung. Es muss jedoch noch das effektive filtern implementiert werden.

6.2 UNIT-TEST

Es müssen jeweils noch Unit-Tests definiert werden.

