

Anforderungen

Movie World (SWE Web, Special Edition)

Abgabe

- Github / Bitbucket Repository
- Dokumentation im README: Technische Zusammenfassung wie die Applikation funktioniert und welche Technologien zu welchen Zwecken eingesetzt wurden. Umfang max. 5 A4 Seiten.
- Vortrag/ Folien

Bewertung (noch tbd)

- | | |
|---------------------------|-----|
| - Funktionalität | 30% |
| - Architektur / Codestyle | 20% |
| - Layout | 10% |
| - Tests | 10% |
| - Dokumentation | 20% |
| - Präsentation | 10% |

Präsentation

- Reflexion ca. 5min
- Welche Aufgaben hatte ich im Team?
- Was habe ich gelernt?
- Was ist mir gut gelungen, was weniger?
- Wo lagen meine grössten Schwierigkeiten und wie habe ich diese gelöst?
- Was würde ich nächstes Mal anders machen?
- Wie zufrieden bin ich mit dem Resultat?
- etc.

Anforderungen

Es soll eine Webapplikation erstellt werden die Daten von TMDb über ihr public API holt und anzeigt.

<https://www.themoviedb.org>

Rechtliche Absicherung:

<https://www.themoviedb.org/faq/api> -> What are the attribution requirements?

Logo: <https://www.themoviedb.org/about/logos-attribution>

Der Benutzer soll in der Lage sein:

- Filme zu suchen (Nach Titel)
- Die 20 am besten bewertete Filme anzuzeigen
- Die 20 am besten bewertete Filme pro Kategorie anzuzeigen
- Bald erscheinende Filme anzuzeigen
- Die 20 populärsten Filme anzuzeigen
- Eine Liste mit seinen letzten Suchabfragen anzeigen und filtern

- Filter nach Datum
 - Filter nach Anzahl Einträge
- Filme zu einer Favoritenliste hinzufügen / entfernen
- Anzeigen der Favoritenliste

Kein Login verlangt. Jeder Benutzer hat die gleichen Favoriten.

Mehrere Resultate (Filme) müssen in einer Übersicht dargestellt werden.

Detailansicht eines Resultats (Film) muss angezeigt werden können.

Technische Vorgaben

Die Tools und Libraries müssen verwendet werden und werden parallel im Unterricht behandelt:

- Java 8
- IntelliJ (Entwicklungsumgebung)
- Apache Maven (Build-System)
- Vaadin (UI Framework)
- RedHat Thorntail (Microservices)
- REST Services
- Visual Paradigm (UML Design, Architektur)
- MongoDB

Die Zusammenarbeit erfolgt über die Cloud:

- GitHub (Tasks, Bugs, Source Code Repository, CI)
- Slack (Chat)
- Clockify (Zeitabrechnung)