

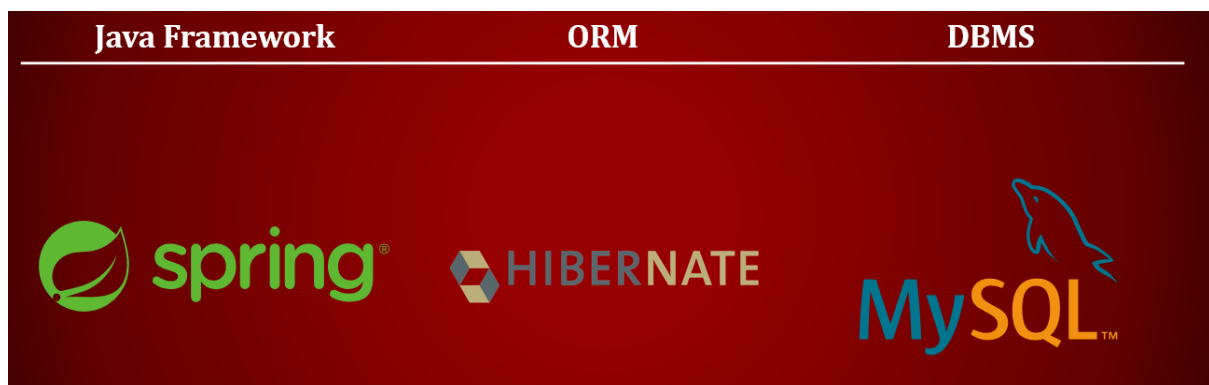
Dokumentation

1. Überblick über das Projekt/Umgesetzte User Storys

Im Zuge der Veranstaltung "Fallstudie" im WiSe 2022/23 wurde ein Kinoticketreservierungssystem entwickelt, welches eine Kinowebsite abbildet, auf der Kinotickets zu unterschiedlichen Veranstaltungen reserviert und gebucht werden können. Dabei wurden folgende User Stories aus der vorangegangenen Planungsveranstaltung "Systemanalyse und Entwurf" (SoSe 2022) umgesetzt:



2. Technologien



3. Pipeline

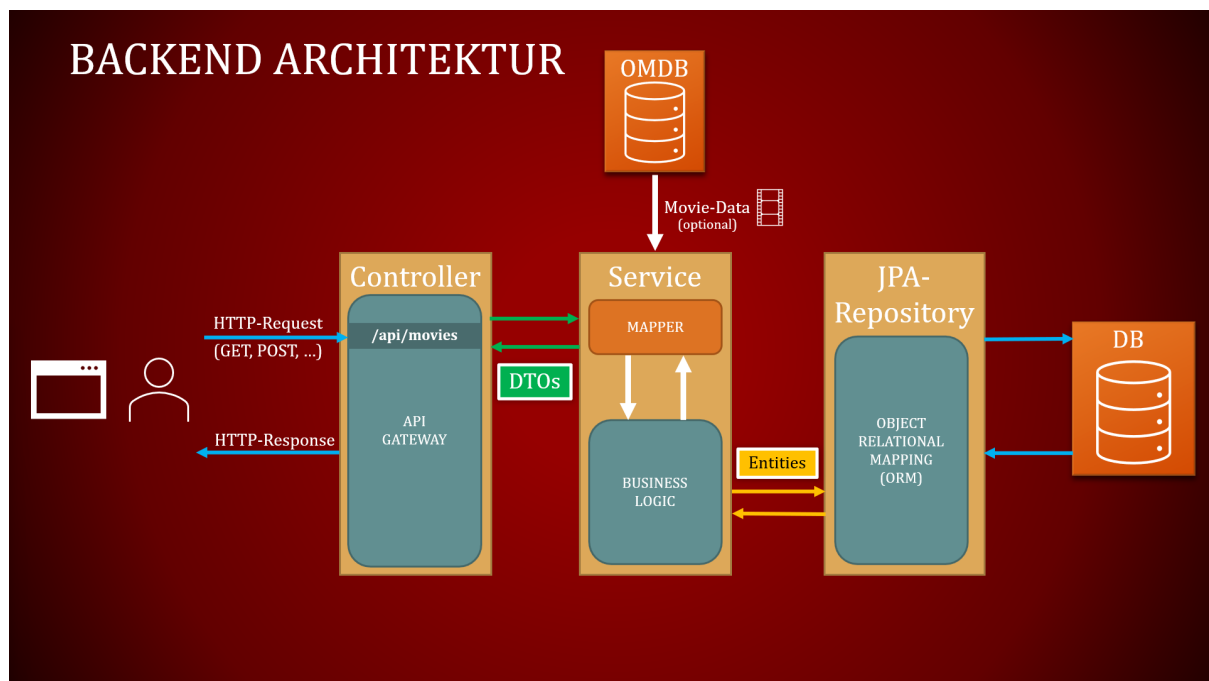
Frontend:

Auf Seite des Frontends haben wir uns ausschließlich auf Funktionen von Github Actions beschränkt und sind im Repository unter Github actions zu finden.

Backend:

Die CI/CD-Pipeline wird über eine selbst gehostete Instanz von [DroneIO](#) gesteuert. DroneIO startet automatisch bei jedem Commit auf das Repository einen CI-Zyklus und führt alle Tests in einem isolierten Container aus. Eine Änderung auf dem Main-Branch wird getestet und zusätzlich noch auf ein [Docker Hub-Repository](#) als Docker-Image gepusht. Die Image-Erstellung erfolgt mithilfe von Jib. Das Backend erkennt automatisch die Änderungen am Docker Hub-Repository und wechselt automatisch auf die neueste Image-Version.

4. Backend Architektur



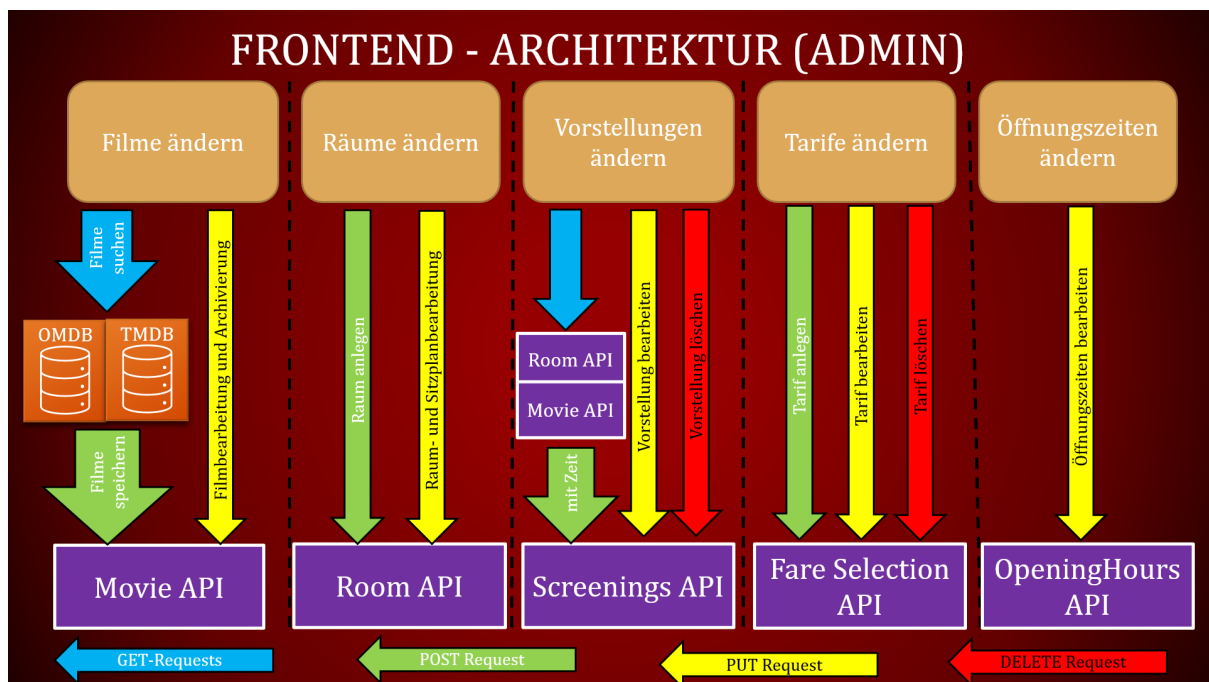
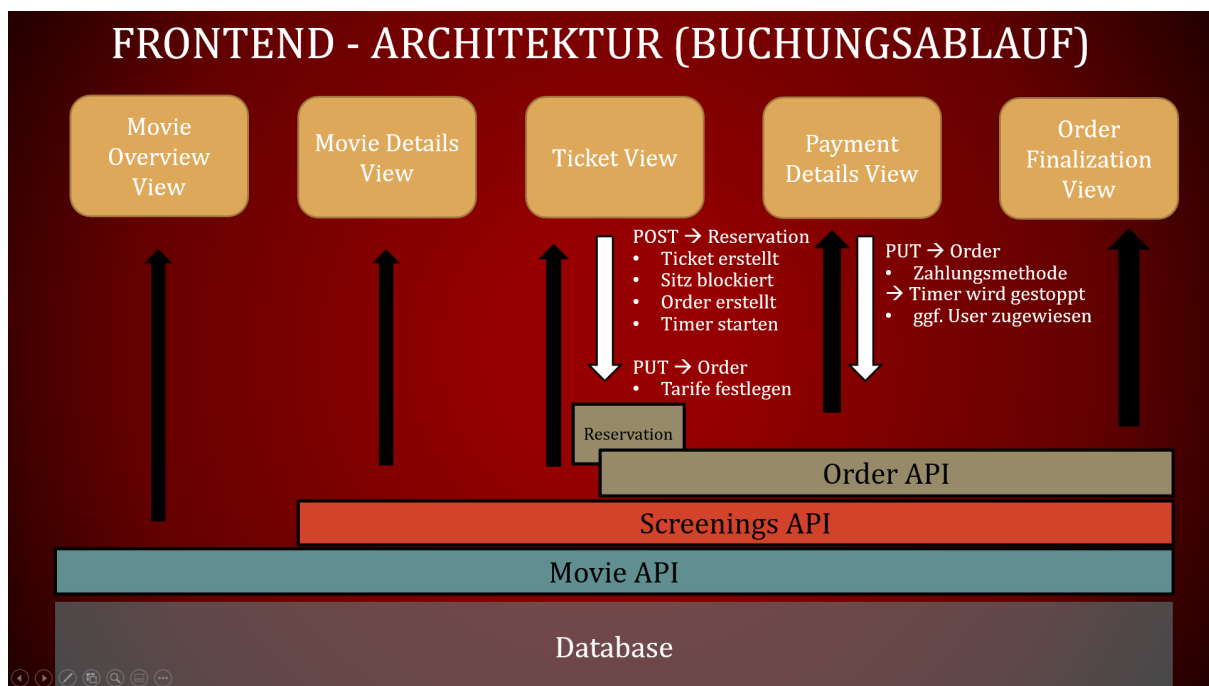
5. Backend Tests

Für das Backend existieren für alle Hauptkomponente eines Spring-Boot-Projektes (Entities, Controller, Services) die dafür benötigten UnitTests. Als grobe Übersicht sind hier alle wichtigen Zahlen (die Werte von links nach recht Class-, Methods-, Lines-Coverage):

> <code>de.cinetastisch.backend.controller</code>	100% (10/10)	96% (59/61)	94% (90/95)
> <code>de.cinetastisch.backend.model</code>	100% (10/10)	94% (126/133)	87% (259/296)
> <code>de.cinetastisch.backend.service</code>	100% (9/9)	91% (55/60)	87% (287/327)

6. Datenbank Schema

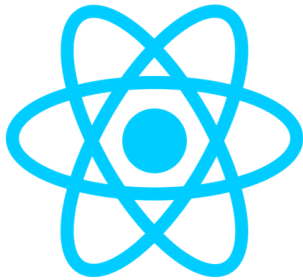
7. Frontend Struktur



8. Frontend Technologien

Im Frontend wurden folgende Technologien verwendet:

- React: einfache Erstellung dynamischer Anwendungen, höhere Effizienz durch wiederverwendbare Komponenten, gleichzeitige Entwicklung von Web Apps und mobilen Anwendungen (erleichterte Reaktivität)
- Typescript: echte Obermenge von JavaScript mit optionaler statischer Typisierung, sodass bereits zur Übersetzungszeit bestimmte Fehler erkannt werden können
- MUI-Components: Nutzung von vorgestylten Komponenten



8. APIs

Zu Dokumentationszwecken haben wir uns dazu entschieden, Swagger zu nutzen, damit wir eine übersichtliche Sammlung unserer API-Schnittstellen haben. Dies ist unter folgendem Link zu finden: <https://api.c930.net/swagger-ui/index.html>

9. Sonstige Informationen und Tutorials

Unsere im Web aufrufbare Azure Webseite ist unter folgendem Link zu finden:

<https://cinema-group4.azurewebsites.net/>

Hinweise zum Testen der Webseite:

- zur Nutzung müssen Drittanbieter-Cookies im Browser erlaubt sein
- Für die Nutzung administrativer Funktionen kann der Nutzer "admin@admin.de" mit dem Passwort "password" verwendet werden

Links zu den beiden Repositories:

Frontend: <https://github.com/mabu2807/Kino>

Backend: <https://github.com/mabu2807/cinema-backend>