

Ausbaustufe 3

(Abgabe: 24. 1. 2021, 24:00 Uhr)

- (A3.1) *Apollo.Core*: Führen Sie die notwendigen Erweiterungen an *Apollo.Core* durch. Überprüfen Sie nochmals Ihr Design, vor allem hinsichtlich einer sauberen Trennung der verschiedenen Schichten und überarbeiten Sie gegebenenfalls Ihre Implementierung (Refactoring).
- (A3.2) *Apollo.Web*: Die Präsentationsschicht ist in Form einer Single-Page Applikation (SPA) mit Angular 10 zu realisieren. Versuchen Sie, in Ihrer Lösung soweit wie möglich zusätzliche Steuerelemente (z. B.: *Bootstrap*, *SemanticUI*, *DevExtreme*, etc.) zu verwenden. Diese Zusatzbibliotheken bieten diverse Komponenten an, die für die Erstellung des Web-Frontends hilfreich sind.

Legen Sie das Design Ihres Systems so aus, dass der Web-Klient über *Apollo.Api* auf die Geschäftslogik zugreifen kann. Konfigurieren Sie die Anwendung so, dass sie leicht an die Anforderungen eines Kunden angepasst werden kann (z. B. Host und Port des Web-Service). Achten Sie außerdem auch auf eine ausreichende client- und serverseitige Validierung.

Setzen Sie eine geeignete Authentifizierung mit OAuth2 und OpenID Connect ein, sodass nur jene Personen Daten modifizieren können, die dazu befugt sind. Es reicht aus, wenn die Web-Oberfläche authentifiziert wird. Das REST-Service muss das nicht leisten!

Fügen Sie geeignete Funktionalität zur Präsentationsschicht hinzu, sodass Benutzer von der besonderen Leistungsfähigkeit Ihrer Anwendung beeindruckt sind. Die Benutzerschnittstelle soll dabei den aktuellen Stand in der Gestaltung von Web-Anwendungen wiedergeben.

Da Sie den Umgang mit den Techniken von Angular üben sollen, ist es Ihnen **nicht** erlaubt, die Services aus z.B. OpenAPI zu generieren. Sie können sich jedoch gerne Umsetzungskonzepte davon abschauen und diese selbstständig einbauen (Anwendung von Konzepten, kein copy-paste). Beschreiben Sie diese dann auch entsprechend in der Dokumentation.

Für eine Benotung mit Sehr Gut ist ein erhöhter Einsatz gefordert: Beschäftigen Sie sich auch mit Techniken, die nicht in der Übung besprochen worden sind und wo ein Einsatz im Projekt sinnvoll erscheint, z.B. Pipes, Internationalisierung, Unit Testing, Validierung und Security, asynchrones Laden von Modulen, etc. Die Anwendung dieser Techniken sollte über bloßes Kopieren aus Tutorials, bzw. Anwenden von vorhandenen Bausteinen hinausgehen (z.B. erstellen Sie eigene Pipes, eigene Validatoren, etc.) und Sie sollten in der Lage sein, bei der Übungspräsentation die Funktionsweise der von Ihnen verwendeten Konzepte im Detail zu erläutern.

- (A3.3) *Apollo.Api*: Diese Systemkomponente ist mit ASP.NET Core MVC als REST-basiertes Web-Service zu realisieren. Die Anforderungen an die zu exportierende Funktionalität des Web-Service ergeben sich aus der Funktionalität von *Apollo.Web*. Das REST-API ist nicht mit OAuth2 abzusichern.
- (A3.4) *Tests*: Überprüfen Sie, ob auch nach Abschluss aller Entwicklungsarbeiten alle Unit- und Integrations- tests erfolgreich ausgeführt werden können.
- (A3.5) *Dokumentation*: Führen Sie die notwendigen Ergänzungen an der Dokumentation durch. Dokumentieren Sie auch den WEA5-Teil ausführlich (Architektur, UML-Diagramme, kommentierte Screenshots, Installationsanleitung sowie Benutzerschnittstellen-Beschreibung) und erstellen Sie dafür ein eigenständiges Dokument. Achten Sie darauf, dass auch der WEA5-Teil adäquat dargestellt wird.

Inhaltsverzeichnis

1.	Allgemein	2
1.1.	<i>Technologien</i>	2
1.2.	<i>Backend.....</i>	2
1.3.	<i>Installation und Inbetriebnahme.....</i>	3
2.	Architektur	4
2.1.	<i>Routen</i>	5
2.2.	<i>Kommunikation mit API.....</i>	5
2.3.	<i>Snackbar – Messages</i>	6
2.4.	<i>Keycloak</i>	7
3.	Benutzeroberfläche	8
3.1.	<i>Startseite</i>	8
3.2.	<i>Movie Details.....</i>	9
3.3.	<i>Admin – Menü</i>	10
3.4.	<i>Admin – Dashboard.....</i>	11
4.	Besondere Komponenten	22
4.1.	<i>Diagramme.....</i>	22
4.2.	<i>Corona Settings</i>	22
4.3.	<i>Id Validierung</i>	23
4.4.	<i>Messages – Snackbar</i>	23
4.5.	<i>Bild Upload.....</i>	24
4.6.	<i>Pipes</i>	24

1. Allgemein

1.1. Technologien

Dieses Projekt wurde mit den Technologien Angular, TypeScript, HTML und CSS umgesetzt. Eine ansprechende Benutzeroberfläche wurde mit Angular-Material erzielt. Zusätzlich zu dieser UI-Bibliothek wurden drei weitere Komponenten (mit „npm“) installiert. Dabei handelt es sich um File-Upload-, Zeitauswahl- sowie eine Diagramm-Komponenten.

1.2. Backend

Das Backend (SWK-Teil) wurde in .Net implementiert und stellt die folgenden Endpoints zur Verfügung (die folgende API-Dokumentation wurde mit „Swagger“ generiert):

Category

- GET /apollo/api/category
- POST /apollo/api/category
- PUT /apollo/api/category
- GET /apollo/api/category/{categoryId}

CinemaHall

- GET /apollo/api/cinemahall
- POST /apollo/api/cinemahall
- PUT /apollo/api/cinemahall
- GET /apollo/api/cinemahall/{cinemaHallId}
- GET /apollo/api/cinemahall/version/{cinemaHallVersionId}
- GET /apollo/api/cinemahall/version/{cinemaHallVersionId}/seats

Genre

- GET /apollo/api/genre
- POST /apollo/api/genre
- PUT /apollo/api/genre
- GET /apollo/api/genre/{genreId}

Movie

- GET /apollo/api/movie/{movieId}/cover
- GET /apollo/api/movie/{movieId}/cover/hd
- GET /apollo/api/movie/{movieId}
- DELETE /apollo/api/movie/{movieId}
- GET /apollo/api/movie
- POST /apollo/api/movie

PaymentStatistic

- GET /apollo/api/paymentstatistic/genre
- GET /apollo/api/paymentstatistic/month/{year}
- GET /apollo/api/paymentstatistic/year
- GET /apollo/api/paymentstatistic/weekday

Reservation

- GET /apollo/api/reservation
- GET /apollo/api/reservation/paid/{reservationId}
- GET /apollo/api/reservation/{reservationId}

Schedule

- GET /apollo/api/schedule
- POST /apollo/api/schedule
- PUT /apollo/api/schedule
- DELETE /apollo/api/schedule
- GET /apollo/api/schedule/{scheduleId}
- GET /apollo/api/schedule/current
- GET /apollo/api/schedule/filtered

1.3. Installation und Inbetriebnahme

- Klonen der Projekte (SWK5 und WEA5)
- Starten der Datenbank im Docker (SWK – Skript ist vorhanden)
- Befüllen der Datenbank mit Testdaten (SWK - Skript ist vorhanden)
- Starten des .NET API-Projekts
- Installation der Abhängigkeiten der Angular-Applikation mit „npm install“
- Konfiguration der Konstanten (Port,...) in „environments/environment.ts“
- Auswahl der Authentifizierungsmethode:
 - o auth.config.ts KeyCloak oder IdentityServer -> beide Konfigurationen stehen zur Verfügung.
- Starten der Angular-Applikation mit „ng serve“. Optional kann der Port durch die Option „-- Port“ explizit angegeben werden.

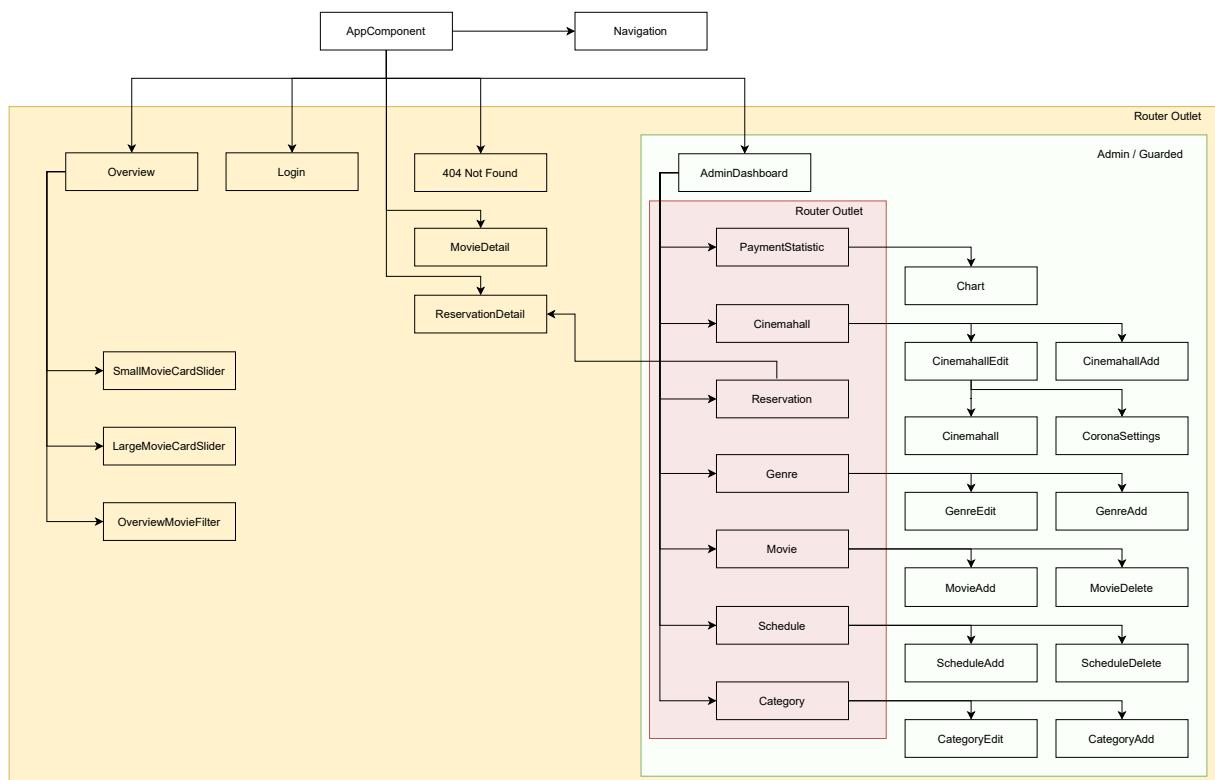
Hinweise:

- In der API-Implementierung wurden explizit Anfragen des Default-Port 4200 erlaubt. Sollte ein anderer Port verwendet werden, so muss der Browser ohne CORS – Sicherheitsregeln gestartet werden.
- Wird KeyCloak als Authentifizierungsmethode verwendet, so muss der Browser ohne CORS – Regeln gestartet werden. (KeyCloak wird über Port 8080 angesprochen)

2. Architektur

Wie bei Angular-Applikationen üblich, wurde die Trennung zwischen API-Schnittstelle (Service) und Code-Behind / UI strikt getrennt. Darüber hinaus wurde die Applikation auf mehrere Komponenten (> 40) aufgeteilt. Admin-Seiten werden durch einen Guard geschützt. Sollte der Benutzer nicht angemeldet sein, so wird dieser auf eine Login-Seite weitergeleitet.

Das nachstehende Diagramm zeigt den Aufbau der Angular-Applikation. Startpunkt der Applikation ist die Komponente „AppComponent“. Es wurden zwei Router-Outlets und ein Guard verwendet. Aus Gründen der Übersichtlichkeit wurde auf die Darstellung aller Komponenten verzichtet.



Alle Bearbeitungsformulare wurden in Form eines Dialogs (PopUp) implementiert. Die Daten werden über den Konstruktor weitergereicht bzw. beim Schließen des Dialogs wieder zurückgegeben.

2.1. Routen

Das folgende Listing zeigt die verwendeten Routen:

```
App-routing.module.ts
const routes: Routes = [
  { path: '', redirectTo: 'overview', pathMatch:'full' },
  {
    path: 'index.html',
    redirectTo: 'overview',
    pathMatch: 'full'
  },
  { path: 'overview', component: MovieOverviewComponent },
  { path: 'movies/:id', component: MovieDetailsComponent },
  { path: 'reservations/:id', component: ReservationDetailComponent },
  {
    path: 'admin',
    component: AdminDashboradComponent,
    canActivate: [NavigateToAdminGuard],
    children : [
      { path: '', redirectTo: 'statistic', pathMatch:'full' },
      { path: 'statistic', component: AdminPaymentStatisticsComponent },
      { path: 'cinemahall', component: AdminCinemaHallComponent },
      { path: 'reservation', component: AdminReservationComponent },
      { path: 'genre', component: AdminGenreComponent },
      { path: 'movie', component: AdminMovieComponent },
      { path: 'schedule', component: AdminScheduleComponent },
      { path: 'category', component: AdminCategoryComponent }
    ]
  },
  { path: 'login', component: AdminLoginComponent },
  { path: '**', component: PageNotFoundComponent }
];
```

2.2. Kommunikation mit API

In der Angular-Applikation stehen insgesamt 7 Services zur Verfügung. Die Funktionalität wurde wie im Backend nach Anwendungsbereichen sortiert.

- CategoryService
- CinemahallService
- GenreService
- MovieService
- PaymentStatisticService
- ReservationService
- ScheduleService

Für eine generische „Live“-ID-Validierung wurde zusätzlich ein Interface „BasisService“ implementiert. Dadurch kann ein beliebiger Service der Validierungsfunktion übergeben werden. Dieses Interface wird von den Services: CategoryService, CinemaHallService, MovieService, GenreService implementiert (da bei den zugehörigen Datensätzen der Benutzer die ID vergibt -> zb Kinosaalbezeichnung: „IMAX“). Das folgende Listing zeigt die Implementierung der Live-Async-ID-Validierung. Bei dieser Implementierung wird eine Debounce-Time von 500ms verwendet.

```
IdValidator.ts
export class IdExistsValidator {
  static createValidator(service: BasisService, time: number = 500): AsyncValidatorFn {
    return (control: AbstractControl): Observable<ValidationErrors> => {
      return timer(time).pipe(
        switchMap(() => service.getById_(control.value)),
        map(isTaken => (isTaken ? { idExists: true } : null)),
        catchError(() => of(null))
      );
    }
}
```

```
    };
}
```

2.3. Snackbar – Messages

Tritt ein Fehler (HTTP Status-Code) bei der Kommunikation mit dem Backend auf, so werden diese für den Benutzer sichtbar in einer Snackbar am unteren Bildschirmrand angezeigt. Die Snackbar wurde als „Injectable“ implementiert und bei Bedarf aufgerufen. Auch „Success“-Nachrichten werden mit Hilfe der Snackbar eingeblendet.

Für die Fehler- und Erfolgsbehandlung wurden ebenfalls generische Funktionen implementiert, die von allen Service-Methoden verwendet werden. Durch dieses Konzept erhält der Benutzer nach Aktionen wie „Create Genre“ Feedback. Die folgenden Listings zeigen einen Ausschnitt der Implementierungen:

Genre.service.ts (Ausschnitt)

```
create(genre: Genre): Observable<any> {
  return this.http.post<Genre>(`${environment.server}/genre`, genre)
    .pipe(
      tap(() => successHandler("genre created", this.snackBarMessage)),
      catchError((err => errorHandler(err, "genre", this.snackBarMessage)))
    );
}
```

Success.ts

```
import { SnackbarMessageService } from "../snackbar-message/snackbar-message.service";

export const successHandler = (message: string, snackBarMessage: SnackbarMessageService) =>
{
  snackBarMessage.openSnackBar(message, "snackbar-message-success");
}
```

Error.ts

```
import { Observable, of } from "rxjs";
import { SnackbarMessageService } from "../snackbar-message/snackbar-message.service";

export const errorHandler = (error: Error | any, message: string, snackBarMessage: SnackbarMessageService): Observable<any> => {
  if (error.status == 409)
    snackBarMessage.openSnackBar(`"${message} id conflict occurred`, "snackbar-message-error");
  else if (error.status == 404)
    snackBarMessage.openSnackBar(`"${message} not found`, "snackbar-message-error");
  else
    snackBarMessage.openSnackBar(`something went wrong :(`, "snackbar-message-error");
  console.log(error);
  return of(null);
}
```

Snackbar-message.service.ts (Ausschnitt)

```
openSnackBar(message: string, styleClass: string) {
  this.snackBar.open(message, null, {
    duration: 5000,
    panelClass: [styleClass]
  });
}
```

2.4. KeyCloak

Als Identity Server wurde, zusätzlich zu jenem aus der Übung, KeyCloak über Docker aufgesetzt. Der Docker-Container kann über den Port 8080 angesprochen bzw. konfiguriert werden. Die Konfiguration wurde mit Hilfe der offiziellen Dokumentation durchgeführt.

Es wurde ein Client „apollo-web“ und ein Benutzer „Michael Neuhold“ angelegt. Zusätzlich wurde noch für den Client der Implizite Flow erlaubt. Die entsprechenden Endpoints für den Login wurden in der Angular-Applikation im Authentifizierungs-Konfigurationsfile hinterlegt. Zusätzlich wurde der „SilentRefresh“ aktiviert, um den id_token zu aktualisieren. Die Session endet nach 1h Inaktivität.

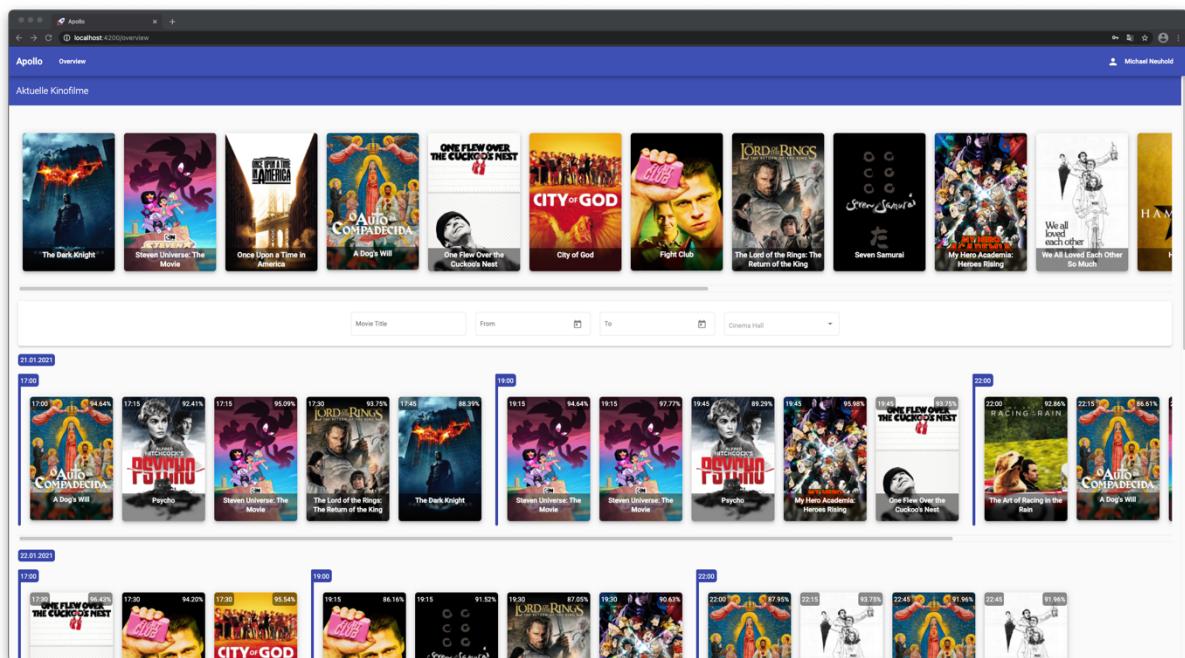
The screenshot shows the Keycloak Admin UI for the 'Apollo' realm. On the left, a sidebar menu is visible with options like 'Configure', 'Clients' (which is selected), 'Client Scopes', 'Roles', 'Identity Providers', 'User Federation', and 'Authentication'. Below that is the 'Manage' section with 'Groups', 'Users', 'Sessions', 'Events', 'Import', and 'Export'. The main content area is titled 'Clients > apollo-web' and shows the 'Apollo-web' client configuration. The 'Settings' tab is active. The configuration fields include:

- Client ID: apollo-web
- Name: (empty input field)
- Description: (empty input field)
- Enabled: ON
- Always Display in Console: OFF
- Consent Required: OFF
- Login Theme: (empty input field)
- Client Protocol: openid-connect
- Access Type: public
- Standard Flow Enabled: ON
- Implicit Flow Enabled: ON
- Direct Access Grants Enabled: OFF

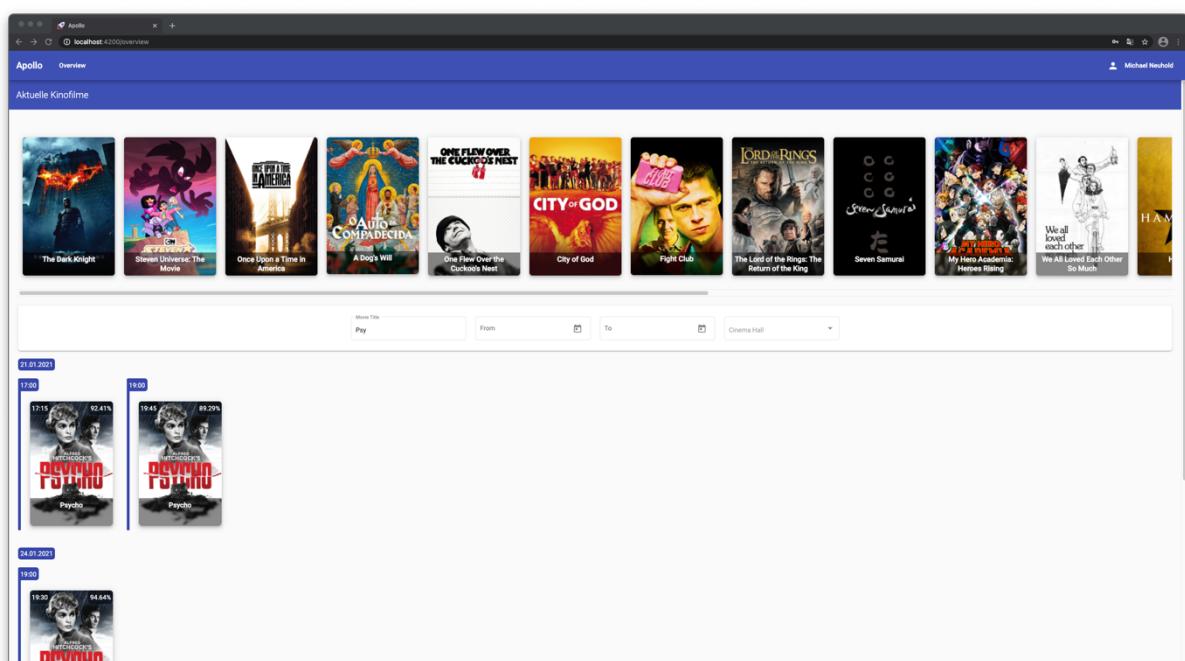
3. Benutzeroberfläche

3.1. Startseite

Die nachstehenden Screenshots zeigen die Filmübersichtsseite mit Suche nach Film-Titel, Datum der Aufführung und Kinosaal. Bei diesen Screenshots sieht man zu dem, dass ein Admin (rechts oben) angemeldet ist. In der ersten Reihe werden die aktuellsten Filme (nicht redundant) angezeigt. Alle Film-Reihen unterhalb der Filtereingabe zeigen eine gruppierte Auflistung nach Aufführungszeit.



Der folgende Screenshot zeigt die Übersicht mit angewandter Filmsuche nach „Psy“.



Der nachstehende Screenshot zeigt die Übersicht mit angewandter Datumssuche (auf einen Tag beschränkt).

The screenshot shows the Apollo web application's movie overview page. At the top, there is a search bar with fields for 'Movie Title', 'From' (set to 1/22/2021), 'To' (set to 1/23/2021), and 'Cinema Hall'. Below the search bar is a grid of movie posters for various films like 'The Dark Knight', 'Steven Universe: The Movie', 'Once Upon a Time in America', etc. The main content area displays a grid of movie posters for the selected date range, showing multiple instances of the same movies like 'The Dark Knight', 'City of God', 'Fight Club', 'Seven Samurai', 'Lord of the Rings: The Return of the King', 'My Hero Academia: Heroes Rising', and 'A Dog's Will'.

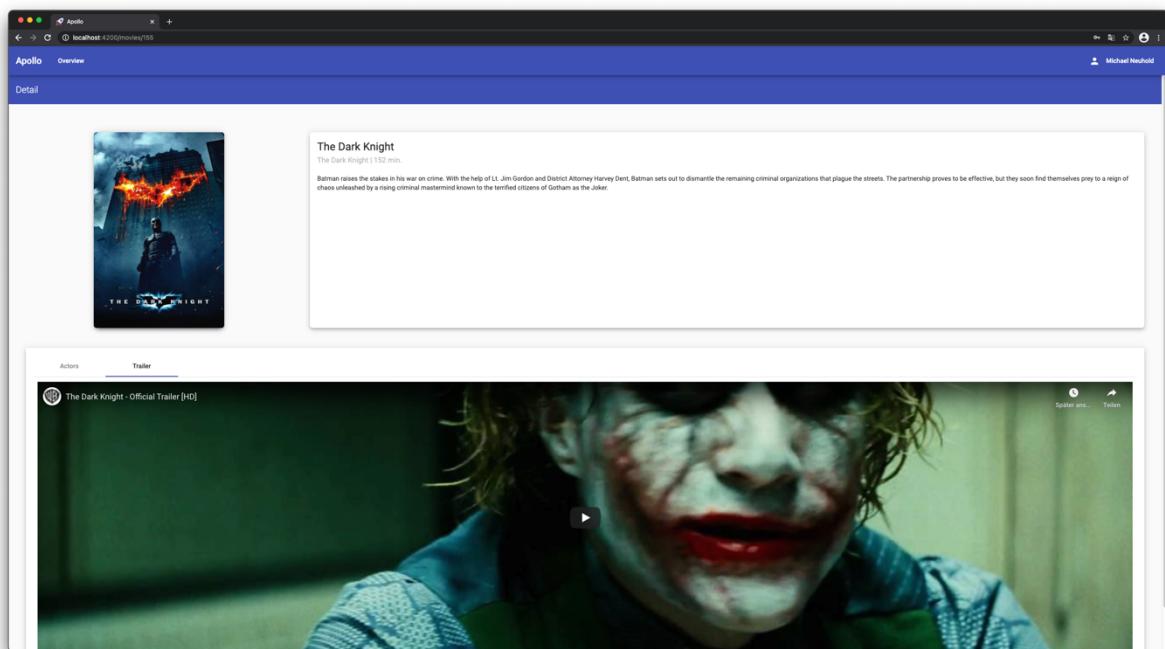
3.2. Movie Details

The screenshot shows the Apollo web application's movie detail page for 'The Dark Knight'. The top section features the movie's poster and the title 'The Dark Knight | 132 min.'. A brief plot summary follows: 'Batman raises the stakes in his war on crime. With the help of Lt. Jim Gordon and District Attorney Harvey Dent, Batman sets out to dismantle the remaining criminal organizations that plague the streets. The partnership proves to be effective, but they soon find themselves prey to a reign of chaos unleashed by a rising criminal mastermind known to the terrified citizens of Gotham as the Joker.' Below the plot summary, there are tabs for 'Actors' and 'Trailer'. The 'Actors' tab lists numerous cast members and their roles, including Heath Ledger (Joker), Michael Caine (Alfred Pennyworth), Gary Oldman (Commissioner James Gordon), Aaron Eckhart (Harvey Dent / Two-Face), Maggie Gyllenhaal (Rachel Dawes), Morgan Freeman (Lucius Fox), Nestor Carbonell (Major Anthony Garcia), Monica Calvani (Carmen (Anna Ramirez)), Ron Dean (Michael Wurtzel), Ng Chin Han (Lau), Eric Roberts (Sal Maron), Ritchie Coster (Chechen), Anthony Michael Hall (Mike Engle), Keith Szarabajka (Gerald Stephens), Joshua Hart (Coleman Reese), Melinda McGraw (Barbara Gordon), Nathan Gamble (James Gordon Jr.), Michael Jai White (Gambit), Colin McFarlane (Gillian R. Loeb), Lydia Rodriguez Terricua (Judge Janet Sunillo), Tom Lister Jr. (Tattooed Prisoner), William Fichtner (Bank Manager), Cillian Murphy (Jonathan Craven / Scarcrow), Patrick Leahy (Gentleman at Party), Matt Skiba (Man Trying to Reach Coleman Reese (uncredited)), David Dastmalchian (Thomas Schiff), Michael Vlach (Rossi), Michael Stoyanov (Dopey), William Smilie (Happy), Danny Goldring (Grumpy), Matthew O'Neill (Chuckles), Olumij Obasuyi (Drug Dealer), Greg Beam (Drug Buyer), Erik Hellman (Junkie), Beatrice Rosen (Natascha), Vincent Nicoll (Crime Boss), Edison Chen (LSI VP), Andy Luther (Brain), James Farugia (Man No. 1), Tom McElroy (Man No. 2), Will Zahn (Assistant DA), James Fierro (Thug #1 Party), Sam Derecke (Male Guest), Jennifer Knue (Female Guest), Patrick Clear (Judge Feed), Sarah Jayne Dunn (Manson's Mistress), Charles Venn (Gambito Bodyguard), Winston G. Ellis (Gambito Bodyguard (or Winston Ellis)), Sophia Hinshelwood (Reporter), Keith Kefner (Hecker), Joseph Luis Caballero (Cop Hecker), Richard Dilane (Acting Commissioner), Daryl Sather (Officer at Intersection), Chris Petzold (Cinny Judge Leader), Alden Feuer (Fat Thug), Philip Bullock (Murphy), Paul Breckin (Cop with Fat Thug), Walter Lewis (Medic), Vincent Scilla (Cop #120 5240 51), Nancy Crane (Nurse), K. Todd Freeman (Polt), Matt Shallenberger (Berg), Michael Andrew Gorman (Cop at Hospital), Larry Lutz (Bartender), Peter DeCarlo (Civillian), Matt Rippy (First Mate), Andrew Blackwell (Prison Ferry Pilot), Arlyon Baker (Guard Commander), Doug Ballard (Businessman), Helene Wilson (Mother), Tommy Campbell (Passenger), Craig Heaney (Passenger), Lorna Gayle (Passenger), Lisa McAllister (Passenger), Peter Brooke (Passenger), Joshua Rollins (SWAT Snipe), Dale Rivers (SWAT Leader), Matthew Letich (Prisoner or Ferry), Thomas Gatisch (Reporter #3), William Armstrong (Dad), Adam Kalepski (Honour Guard Man), Trista Tait (Uniform Cop), Bronson Webb (Doubtful Hunter), David Ajala (Boys' Secretary), Gertrude Kyles (Fox's Secretary), Jonathan Ryland (Passenger Ferry Pilot), James Scales (Guardian), Nigel Carrington (Warden), Ian Pile (Corrections Officer), Latest Lovley (Passenger), Graham Edwards (Prisoner), Roger Monk (Prisoner), Ronan Summers (Prisoner), Wil Wong (Hong Kong Detective), Michael Cory Foster (Honor Guard Leader), Hannah Gunn (Gordon's Daughter), Brandon Lambdin (Armond Cor SWAT), Jon Lee Brody (Water (uncredited)), Debbi Burns (Bank Patron (uncredited)), Maritza Cabrera (Party Guest (uncredited)), Shirin Caola (Party Guest with Glass (uncredited)), Laura Chernicky (Party Guest (uncredited)), Henry Milton Chu (Lau Henchman (uncredited)), Kelli Clavenger (Paramedic (uncredited)), Richard Dixie (Chechen Gangster (uncredited)), Tony Dorino (Press Conference Hacker (uncredited)), David Futre (Pedestrian (uncredited)), Natalie Hallam (Ferry Passenger (uncredited)), Jordan Hodges (Police Officer (uncredited)), Eron Jay (Prisoner (uncredited)), Nicky Katt (Shotgun SWAT (uncredited)), Thomas Kork (Parade Police Officer (uncredited)), Don Kress (Maron's Henchman (uncredited)), Tim Kruger (Assistant D.A. (uncredited)), Dan Latham (Police Sgt. Spellman / Gotham Bomb Squad (uncredited)), Tom McComas (Helicopter SWAT Sniper (uncredited)), James Mellor (Ferry Passenger (uncredited)), Joseph Oliveira (Officer (uncredited)), Buster Reeves (Joker's Thug #2 (uncredited)), Peter Ricci (Prisoner (uncredited)), Amit Shah (Party Guest (uncredited)), Michelle Shields (Angry Hospital Relative (uncredited)), Sofiya Simmone (Evacuee (uncredited)), Bruce Spiegelauer (High-Ranking Police Officer (uncredited)), Robert Patrick Stern (Extra (uncredited)), Robert Stone (Dept. of Corrections Resident (uncredited)), Richard Strabel (Detective (uncredited)), Tim Townsend (Police Officer (uncredited)), John Turk (Chechen Bodyguard (uncredited)), John Warman (Detective (uncredited)))'. The bottom of the page has a footer with links to 'Actors', 'Trailer', and other sections.

Anmerkung:

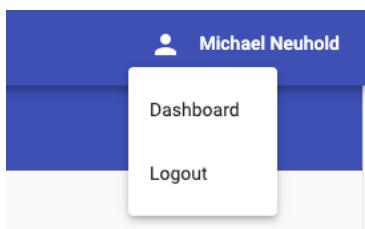
Für performantes Laden der Seiten wurden zwei verschiedene Bilder bzw. Bild-Endpoints verwendet. Bei der Filmübersichtsseite werden Bilder mit geringerer Auflösung angezeigt und auf der Movie Detail Seite werden Bilder mit HD-Auflösung angezeigt.

Es kann zwischen den Anzeigen „Schauspieler“ und „Trailer“ gewechselt werden. Der Trailer wurde durch ein „IFrame“ eingebunden.



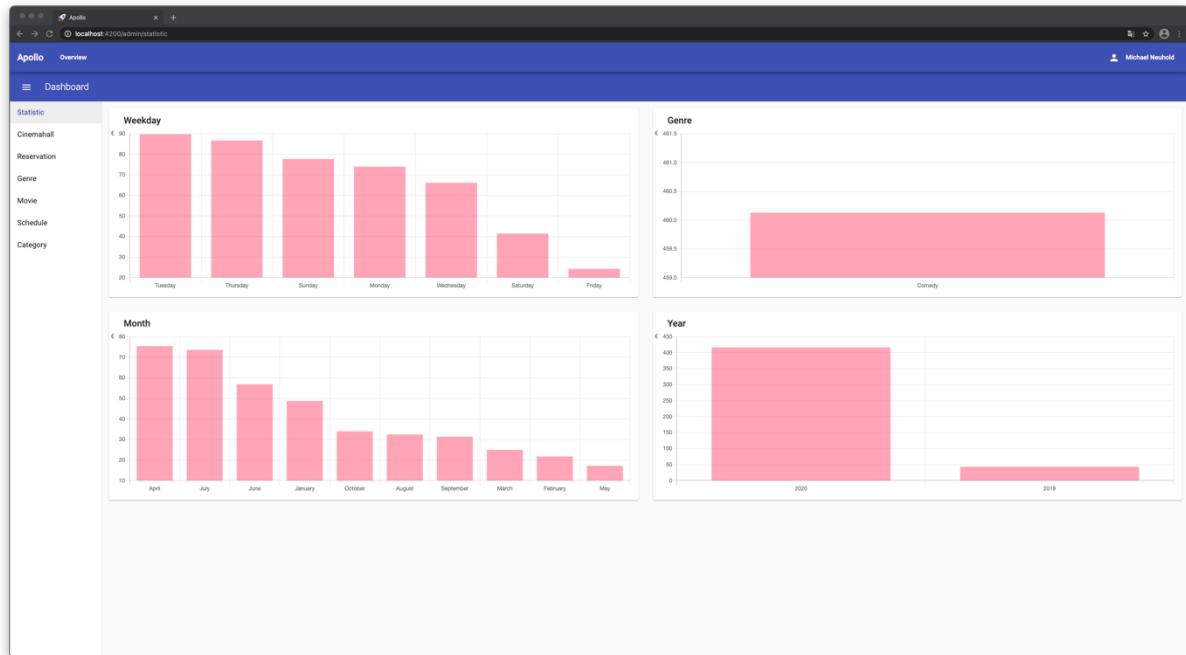
3.3. Admin – Menü

Ist man angemeldet, so kann man durch Klick auf den Benutzernamen auf das Admin-Dashboard wechseln oder ein „Logout“ durchführen. Der Benutzername wird von den „IdentityClaims“ ausgelesen und angezeigt.

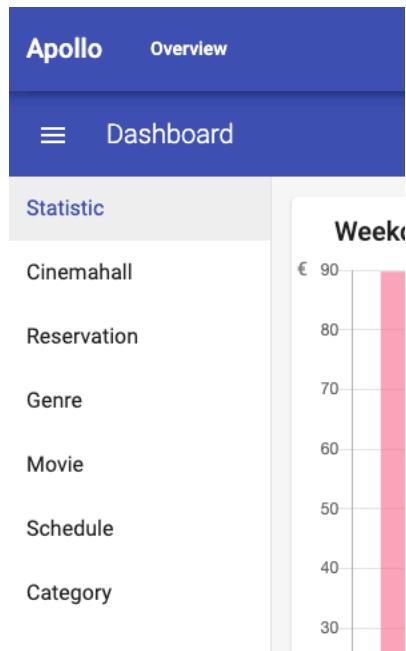


3.4. Admin – Dashboard

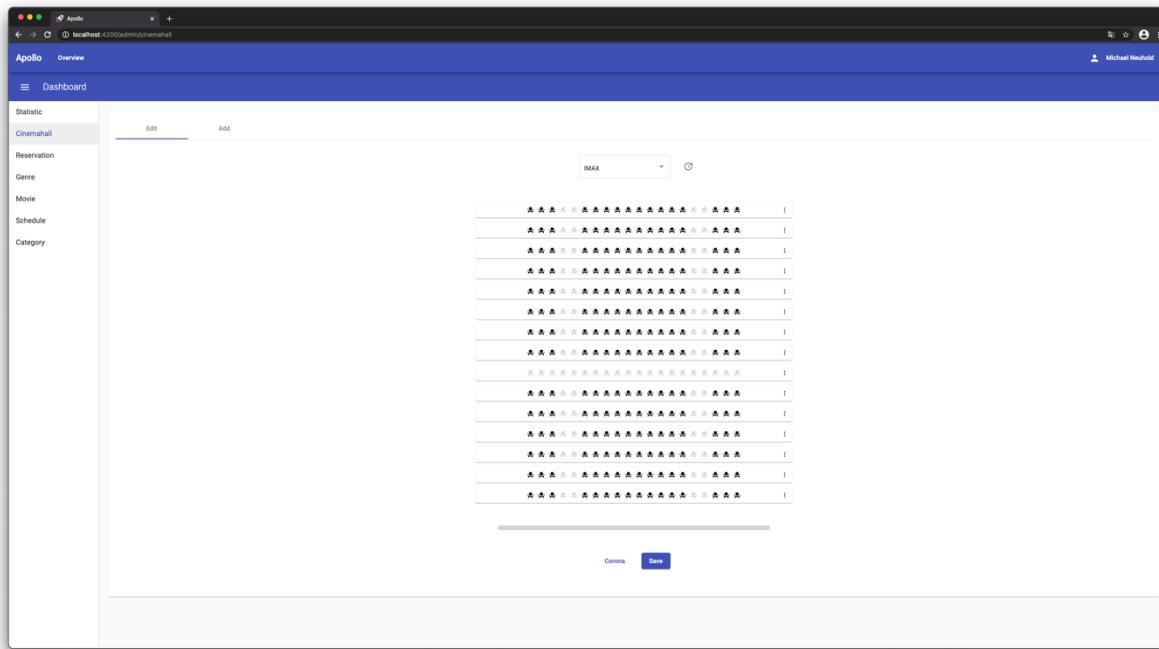
Der nachstehende Screenshot zeigt einen Überblick über das Admin-Dashboard. Das Backend unterstützt Auswertungen über Bezahlte Reservierungen. Aus diesem Grund wurden Diagramme zur Visualisierung eingebunden.



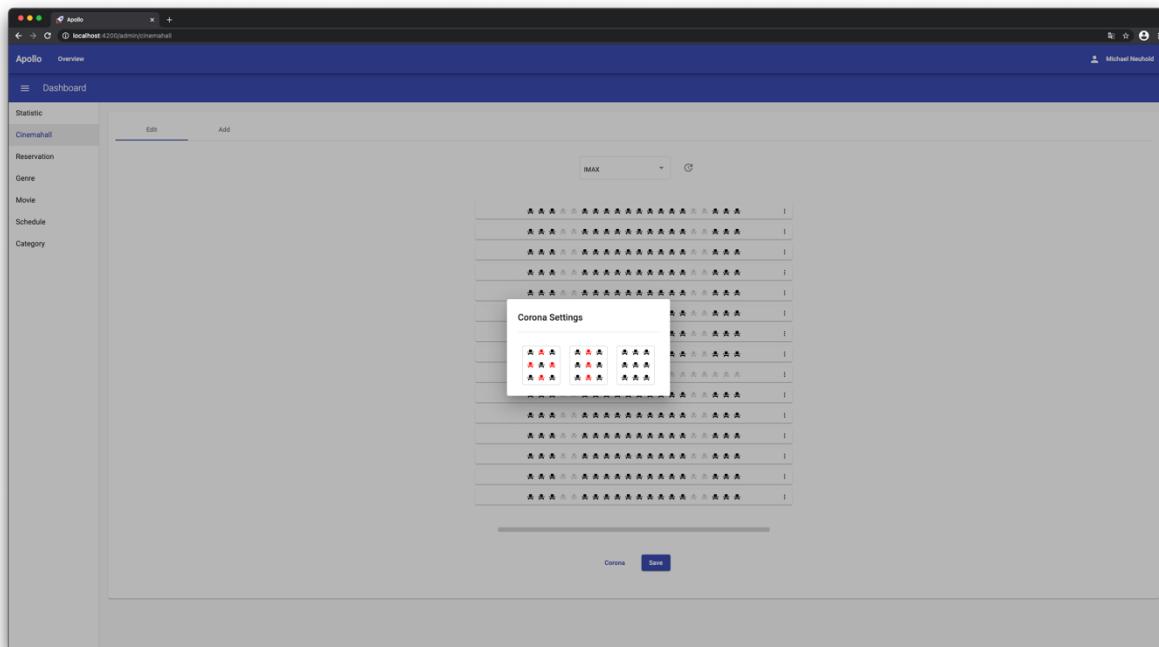
Konkret können folgenden Punkte über das Admin-Dashboard gepflegt werden:



Die nachstehenden Screenshots zeigen die Funktionalitäten für „CinemaHall“.



Der nachstehende Screenshot zeigt die Corona-Einstellungen für das Speeren von Sitzplätzen nach einem bestimmten Muster. Bereits Reservierte Sitze sind nicht betroffen, da die Cinemahalls in der Datenbank versioniert werden.



Sitze können auch manuell gesperrt (corona) oder als Gang (not available / grau) markiert werden.



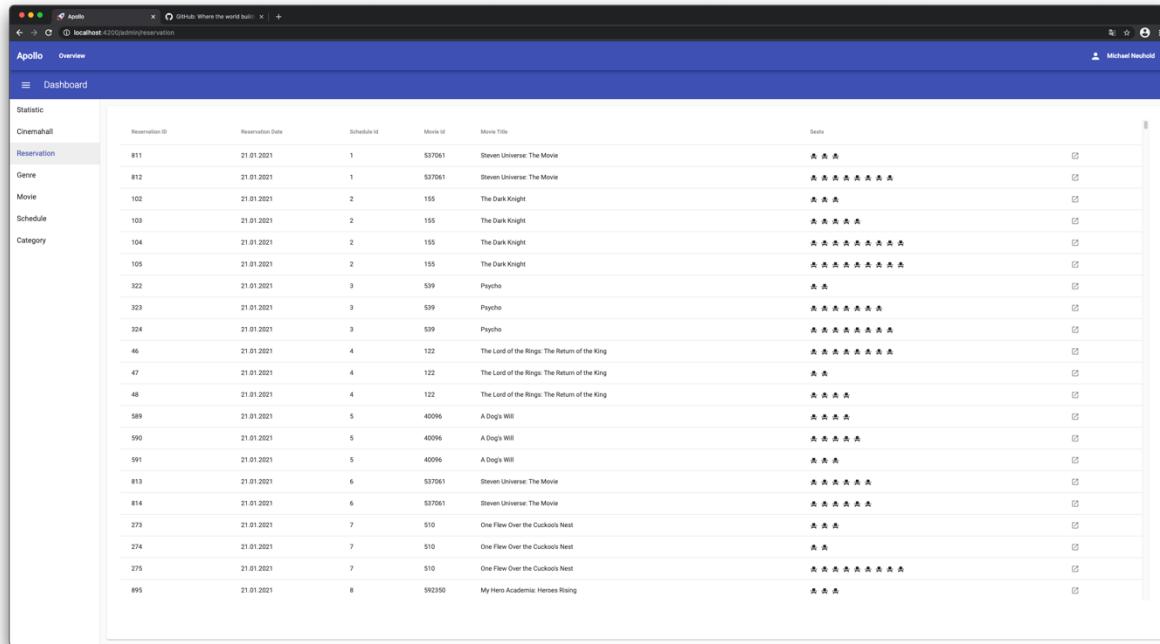
Wird der Button auf für eine Sitzreihe betätigt, so kann die Kategorie der Sitzreihe geändert werden.



Wird auf den Tab „Add“ gewechselt, so kann ein neuer Kinosaal hinzugefügt werden. Dabei kann die „Standardgröße (15x20)“ oder eine neue Größe ausgewählt werden. Hier wird der Name des Kinosaals als Schlüssel verwendet. Aus diesem Grund wird während der Eingabe mit dem zuvor angesprochenen Async-Validator überprüft, ob die ID (Saal 10) noch verfügbar ist.

Name	Name Saal 10 <small>valid</small>	Name IMAX <small>cinemahall name already exists</small>
Rows	Rows 15	Rows <small>row number is required</small>
Seats	Seats 20	Seats <small>seats number is required</small>
<input type="checkbox"/> use default size	<input checked="" type="checkbox"/> use default size	<input type="checkbox"/> use default size
<button>Save</button>	<button>Save</button>	<button>Save</button>

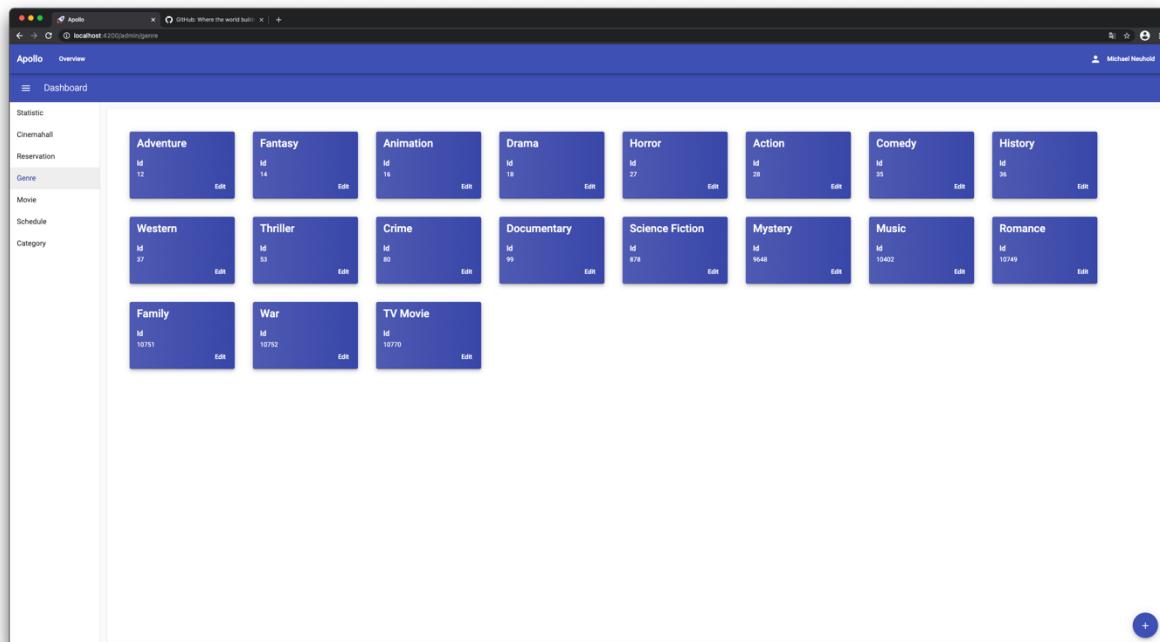
Der nachstehende Screenshot zeigt eine tabellarische Auflistung der bereits getätigten Reservierungen.



The screenshot shows a table titled "Reservations" with the following columns: Reservation ID, Reservation Date, Schedule ID, Movie ID, Movie Title, and Seats. The data includes various movie titles like "Steven Universe: The Movie", "The Dark Knight", and "My Hero Academia: Heroes Rising". Each row has an edit icon on the right.

	Reservation ID	Reservation Date	Schedule ID	Movie ID	Movie Title	Seats	
Reservation	811	21.01.2021	1	537061	Steven Universe: The Movie	▲ ▲ ▲	
Genre	812	21.01.2021	1	537061	Steven Universe: The Movie	▲ ▲ ▲ ▲ ▲ ▲ ▲	
Movie	102	21.01.2021	2	155	The Dark Knight	▲ ▲ ▲	
Schedule	103	21.01.2021	2	155	The Dark Knight	▲ ▲ ▲	
Category	104	21.01.2021	2	155	The Dark Knight	▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲	
	105	21.01.2021	2	155	The Dark Knight	▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲	
	322	21.01.2021	3	539	Psycho	▲ ▲	
	923	21.01.2021	3	539	Psycho	▲ ▲ ▲ ▲ ▲ ▲	
	324	21.01.2021	3	539	Psycho	▲ ▲ ▲ ▲ ▲ ▲ ▲	
	46	21.01.2021	4	122	The Lord of the Rings: The Return of the King	▲ ▲ ▲ ▲ ▲ ▲ ▲	
	47	21.01.2021	4	122	The Lord of the Rings: The Return of the King	▲ ▲	
	48	21.01.2021	4	122	The Lord of the Rings: The Return of the King	▲ ▲ ▲ ▲	
	589	21.01.2021	5	40096	A Dog's Will	▲ ▲ ▲	
	590	21.01.2021	5	40096	A Dog's Will	▲ ▲ ▲ ▲	
	591	21.01.2021	5	40096	A Dog's Will	▲ ▲ ▲	
	813	21.01.2021	6	537061	Steven Universe: The Movie	▲ ▲ ▲ ▲ ▲	
	814	21.01.2021	6	537061	Steven Universe: The Movie	▲ ▲ ▲ ▲ ▲	
	273	21.01.2021	7	510	One Flew Over the Cuckoo's Nest	▲ ▲	
	274	21.01.2021	7	510	One Flew Over the Cuckoo's Nest	▲ ▲	
	275	21.01.2021	7	510	One Flew Over the Cuckoo's Nest	▲ ▲ ▲ ▲ ▲ ▲ ▲	
	895	21.01.2021	8	592350	My Hero Academia: Heroes Rising	▲ ▲ ▲	

Der nachstehende Screenshot zeigt alle verfügbaren Genre. Mit Klick auf Edit kann die Bezeichnung geändert werden. Mit Klick auf den „Hinzufügen“-Button (rechts unten) können neue Genre hinzugefügt werden.



The screenshot shows a grid of genre cards. Each card contains the genre name, its ID, and an edit icon. A plus sign icon is located in the bottom right corner of the grid area.

Adventure Id 12 Edit	Fantasy Id 14 Edit	Animation Id 16 Edit	Drama Id 18 Edit	Horror Id 27 Edit	Action Id 28 Edit	Comedy Id 35 Edit	History Id 35 Edit
Western Id 27 Edit	Thriller Id 53 Edit	Crime Id 80 Edit	Documentary Id 99 Edit	Science Fiction Id 878 Edit	Mystery Id 9548 Edit	Music Id 10402 Edit	Romance Id 10749 Edit
Family Id 10751 Edit	War Id 10752 Edit	TV Movie Id 10770 Edit					

The screenshot shows the Apollo Web application's 'Genre' management interface. On the left, a sidebar menu includes 'Statistic', 'Cinemahall', 'Reservation', 'Genre' (which is selected), 'Movie', 'Schedule', and 'Category'. The main area displays a grid of genre cards. A card for 'Family' (Id: 10751) has an 'Edit' button highlighted. A modal window titled 'Edit genre' is open over the grid, containing a single input field with the value 'Family'. Below the input are 'Cancel' and 'Save' buttons.

This screenshot shows the same Apollo Web application interface, but the 'Genre' card in the grid is now highlighted with a blue border. A modal window titled 'Add genre' is open, featuring two input fields: one with the value '24324' and another with the placeholder 'Documentation WEA5'. At the bottom of the modal are 'Cancel' and 'Add' buttons.

Der folgende Screenshot zeigt eine Auflistung aller verfügbaren Filme. Filme können hinzugefügt und gelöscht werden. Durch eine Verlinkung gelangt man auf die Movie – Detail Seite.

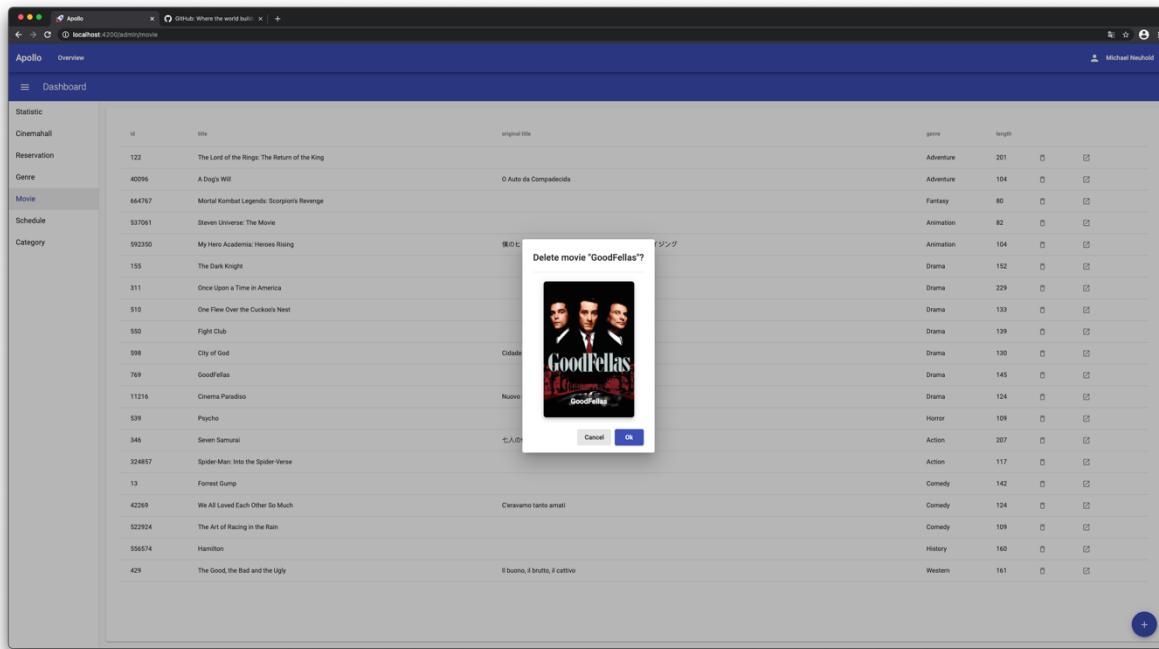
The screenshot shows a web-based administration interface for a movie database. The left sidebar has a navigation menu with items like Statistic, Cinemall, Reservation, Genre, Movie (which is selected), Schedule, and Category. The main content area displays a table of movies with columns: id, title, original title, genre, and length. Each row in the table represents a movie entry, with a small thumbnail image and edit/delete icons. A blue '+' button is located at the bottom right of the table.

	id	title	original title	genre	length
Cinemall	122	The Lord of the Rings: The Return of the King		Adventure	201
Reservation	40996	A Dog's Will	O Auto da Compadecida	Adventure	104
Genre	664767	Mortal Kombat Legends: Scorpion's Revenge		Fantasy	80
Movie	537061	Steven Universe: The Movie		Animation	82
Schedule	992350	My Hero Academia: Heroes Rising	僕のヒーローアカデミア THE MOVIE ヒーローズ・ライジング	Animation	104
Category	155	The Dark Knight		Drama	152
	311	Once Upon a Time in America		Drama	229
	510	One Flew Over the Cuckoo's Nest		Drama	133
	550	Fight Club		Drama	139
	998	City of God	Cidade de Deus	Drama	130
	769	Goodfellas		Drama	145
	11216	Cinema Paradiso	Nuovo Cinema Paradiso	Drama	124
	539	Psycho		Horror	109
	346	Seven Samurai	七人の侍	Action	207
	324857	Spider-Man: Into the Spider Verse		Action	117
	13	Forrest Gump		Comedy	142
	42269	We All Loved Each Other So Much	C'eravamo tanto amati	Comedy	124
	522924	The Art of Racing in the Rain		Comedy	109
	556574	Hamilton		History	160
	429	The Good, the Bad and the Ugly	Il buono, il brutto, il cattivo	Western	161

Der nachstehende Screenshot zeigt das Formular zum Anlegen eines neuen Films. Es können zwei Bilder durch einen Fileupload hinzugefügt werden. Diese werden „base64“-formatiert an das Backend übertragen und in der Datenbank gespeichert.

This screenshot shows the 'Add movie' form overlaid on the movie list. The form fields include: movie #*, movie title*, movie original title*, title des neuen Films, and trailer (optional). There is a text area for a description and a dropdown for genre. Below the form, two movie posters are displayed as thumbnails with their respective base64 URLs. The 'Add' button is visible at the bottom right of the form.

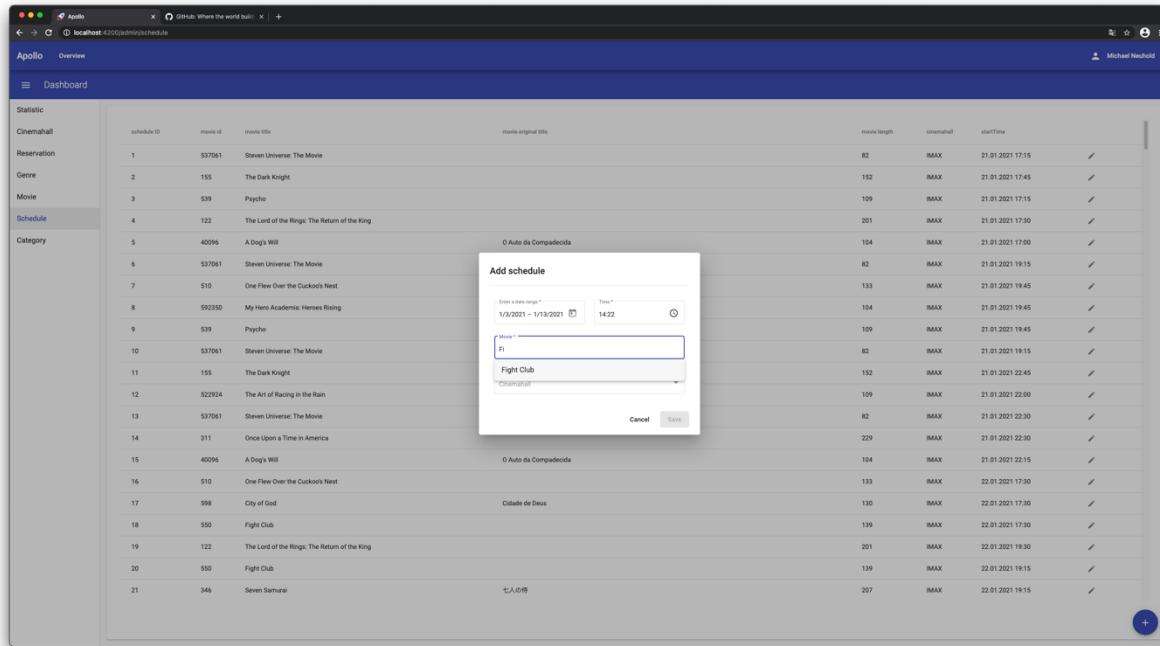
Der nachstehende Screenshot zeigt den Dialog zum Löschen eines Films.



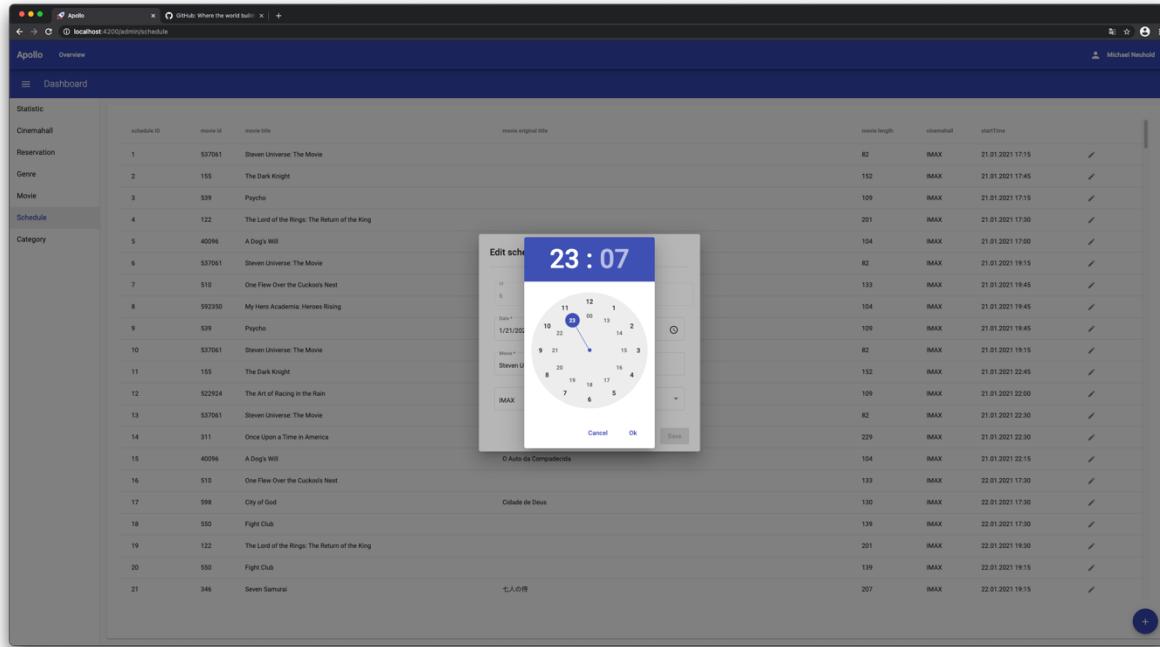
Der nachstehende Screenshot zeigt die Übersicht über alle angelegten Filmvorstellungen. Filmvorstellungen können bearbeitet und angelegt werden.

	schedule ID	movie ID	movie title	movie original title	movie length	cinemahall	startTime	
Cinemahall	1	537061	Steven Universe: The Movie		82	IMAX	21.01.2021 17:15	✓
Reservation	2	155	The Dark Knight		152	IMAX	21.01.2021 17:45	✓
Genre	3	539	Psycho		109	IMAX	21.01.2021 17:15	✓
Movie	4	122	The Lord of the Rings: The Return of the King		201	IMAX	21.01.2021 17:30	✓
Schedule	5	40996	A Dog's Will	O Auto da Compadecida	104	IMAX	21.01.2021 17:00	✓
Category	6	537061	Steven Universe: The Movie		82	IMAX	21.01.2021 17:15	✓
	7	510	One Flew Over the Cuckoo's Nest		133	IMAX	21.01.2021 19:45	✓
	8	592350	My Hero Academia: Heroes Rising	僕のヒーローアカデミア THE MOVIE ヒーローズ、ライジング	104	IMAX	21.01.2021 19:45	✓
	9	539	Psycho		109	IMAX	21.01.2021 19:45	✓
	10	537061	Steven Universe: The Movie		82	IMAX	21.01.2021 19:15	✓
	11	155	The Dark Knight		152	IMAX	21.01.2021 22:45	✓
	12	522924	The Art of Racing in the Rain		109	IMAX	21.01.2021 22:00	✓
	13	537061	Steven Universe: The Movie		82	IMAX	21.01.2021 22:30	✓
	14	311	Once Upon a Time in America		229	IMAX	21.01.2021 22:30	✓
	15	40996	A Dog's Will	O Auto da Compadecida	104	IMAX	21.01.2021 22:15	✓
	16	510	One Flew Over the Cuckoo's Nest		133	IMAX	22.01.2021 17:30	✓
	17	598	City of God	Cidade de Deus	130	IMAX	22.01.2021 17:30	✓
	18	550	Fight Club		139	IMAX	22.01.2021 17:30	✓
	19	122	The Lord of the Rings: The Return of the King		201	IMAX	22.01.2021 19:30	✓
	20	550	Fight Club		139	IMAX	22.01.2021 19:15	✓
	21	346	Seven Samurai	七人の侍	207	IMAX	22.01.2021 19:15	✓

Der folgende Screenshot zeigt das Anlegen einer neuen Vorstellung. Es kann eine Datumsspanne, eine Uhrzeit, ein Film sowie ein Kinosaal gewählt werden. Bei der Auswahl des Films unterstützt die Applikation durch eine Autovervollständigung.



Der nachstehende Screenshot zeigt das Ändern der Uhrzeit einer Vorstellung. Hier wird die zusätzlich installierte Time-Picker Komponente verwendet.



Der nachstehende Screenshot zeigt die Übersicht der Sitzkategorien.

The screenshot shows the Apollo Web application's interface. On the left, there is a sidebar with a navigation menu containing links: Statistic, Cinemahall, Reservation, Genre, Movie, Schedule, and Category. The 'Category' link is highlighted. The main content area displays three seat categories in cards:

- Standard**: Description: Standard Sitz mit wenig Beinfreiheit. Price: 10 €. Edit button.
- Deluxe**: Description: Deluxe Sitz mit viel Beinfreiheit und Tisch. Price: 12 €. Edit button.
- 4D**: Description: Ganz neues Kinoerlebnis... Price: 14 €. Edit button.

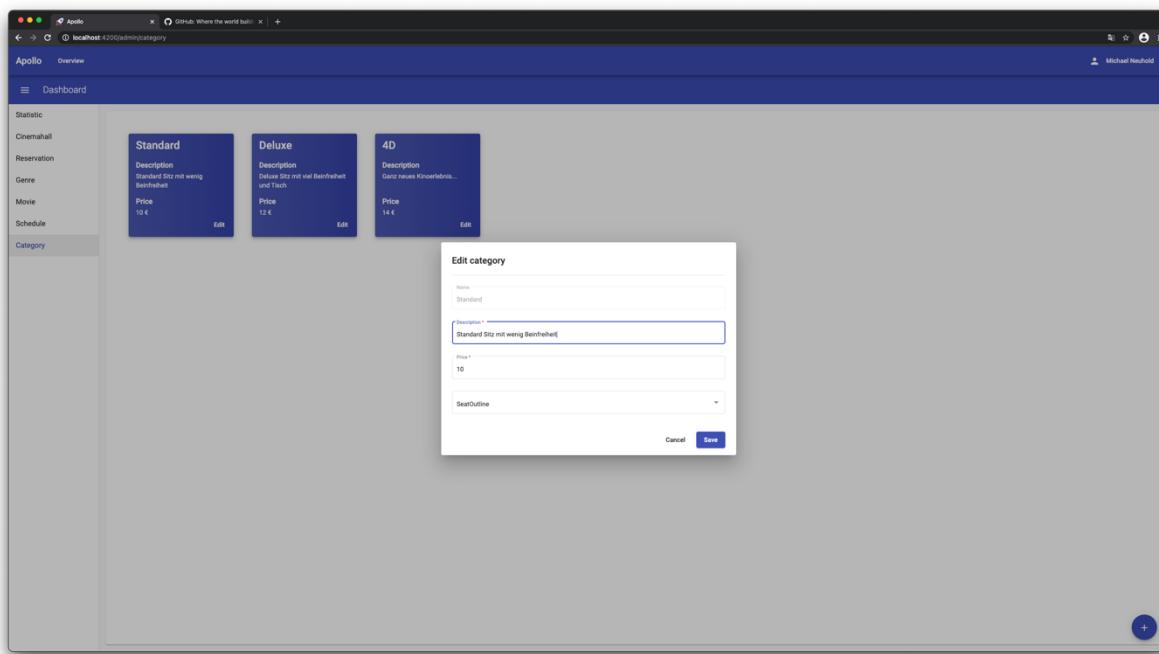
A blue circular button with a '+' sign is located in the bottom right corner of the main content area.

Die folgenden Screenshots zeigen das Hinzufügen und Ändern der Kategorien.

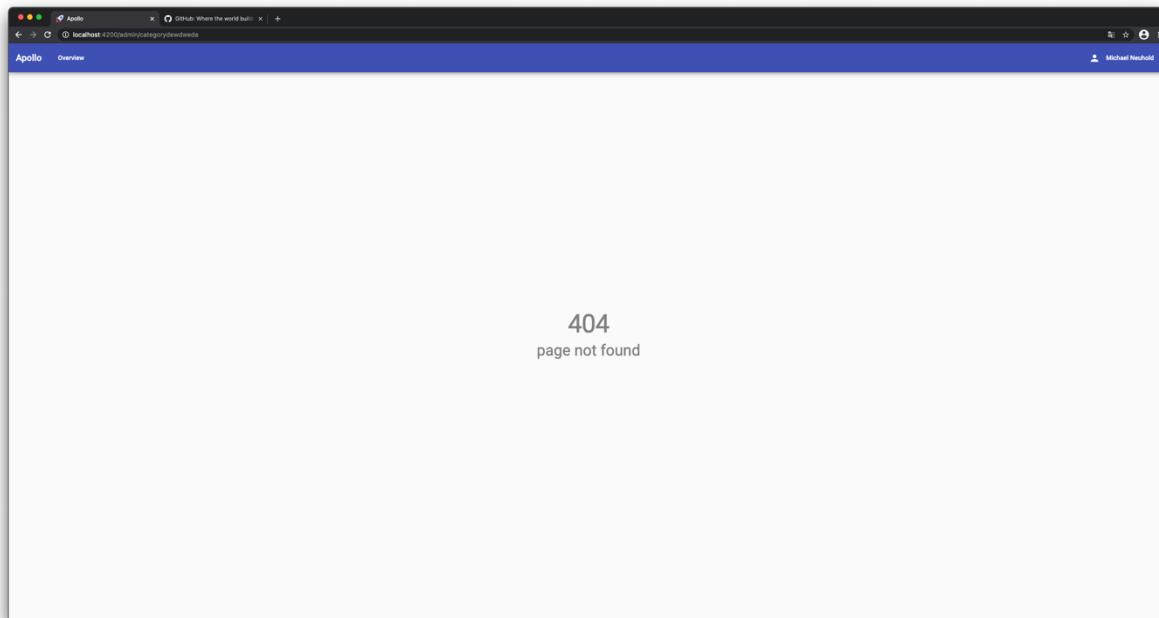
The screenshot shows the Apollo Web application's interface with a modal dialog box titled 'Add category'. The dialog contains the following fields:

- Name*: An input field with the placeholder 'Name*'.
- Bezeichnung*: An input field with the placeholder 'Bezeichnung*'.
- Preis*: An input field with the placeholder 'Preis*'.
- Icon: A dropdown menu with the placeholder 'Icon'.

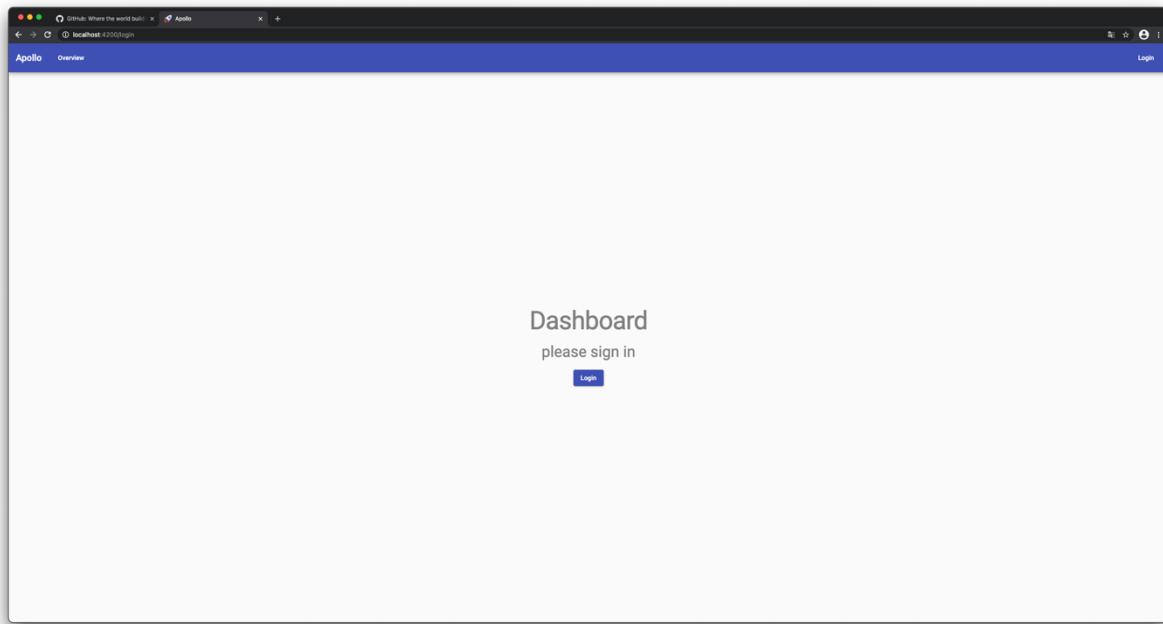
At the bottom of the dialog, there are 'Cancel' and 'Add' buttons.



Wird eine unbekannte URL eingegeben, so wird eine 404 – Error Message / Komponente angezeigt.



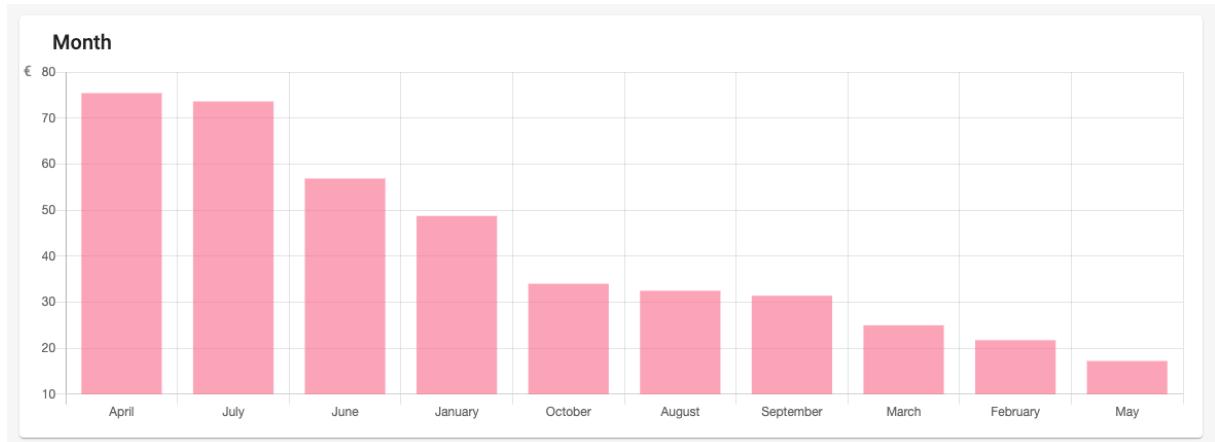
Wird zur Admin-URL navigiert, ohne Anmeldung, so wird die folgende Komponente angezeigt.



4. Besondere Komponenten

4.1. Diagramme

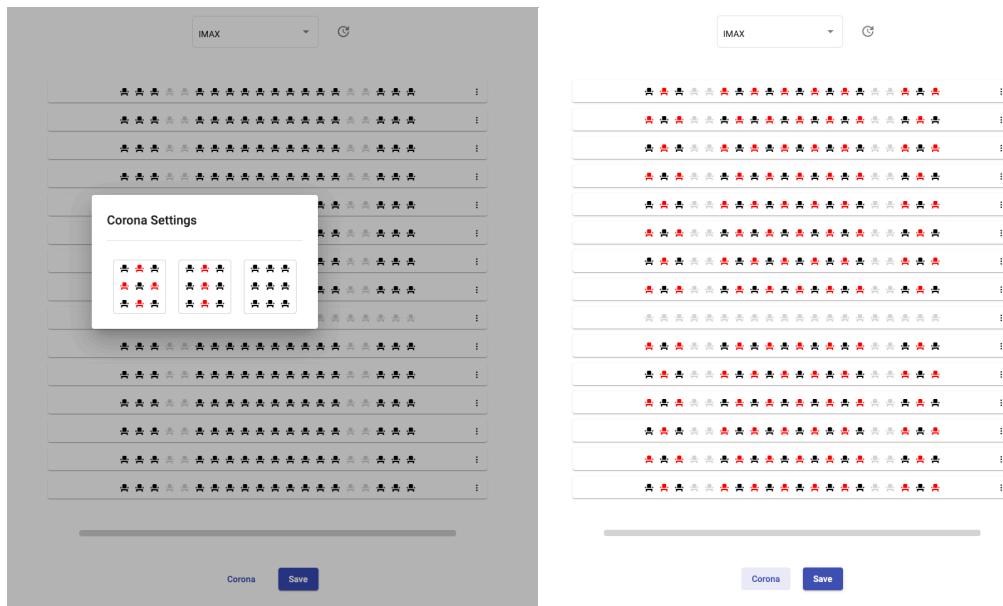
Die Diagramme wurden mit Hilfe der Bibliothek „ng2-charts“ umgesetzt. Die Implementierung wurde durch die API bereits vorgesehen. Es wurde eine Chart-Komponente entwickelt, die mehrmals mit unterschiedlichen Daten angezeigt wird.



4.2. Corona Settings

Es wurden drei Corona-Modi vorgesehen, die über das Admin-Dashboard auf einen Kinosaal angewandt werden können. Es können die folgenden Modi gewählt werden:

- Chessboard (siehe Bild)
- Alternately
- No-Restrictions



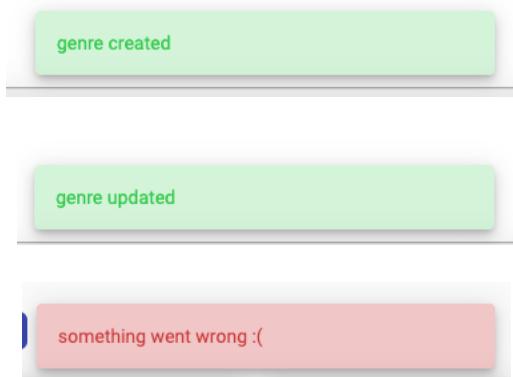
4.3. Id Validierung

Durch die ID-Validierung kann bereits vor dem Absenden des Formulars die Verfügbarkeit einer ID überprüft werden. Wie bereits zuvor erläutert wurde dafür ein Interface implementiert, wodurch eine generische Weitergabe der Services an den Validator ermöglicht wurde.

The screenshot shows a mobile application interface with a header bar displaying '99' and '878'. Below it is a modal dialog titled 'Add genre'. Inside the dialog, there are two input fields. The first field, labeled 'Id *', contains the value '12' and has a red border, indicating an error. A red error message 'genre id already exists' is displayed below this field. The second field, labeled 'Name *', is empty and has a blue border. At the bottom of the dialog are two buttons: 'Cancel' and 'Add'.

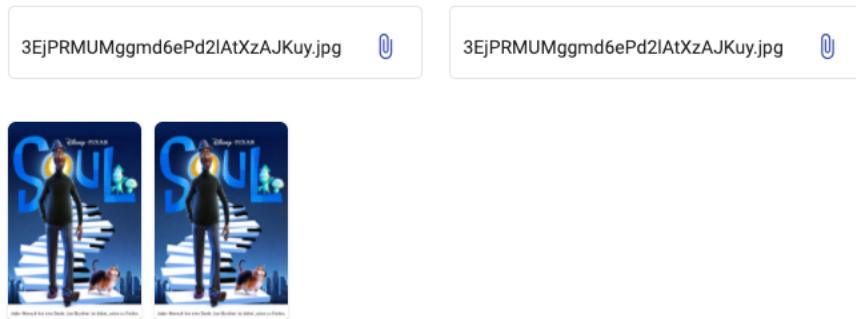
4.4. Messages – Snackbar

Alle Service Methoden verwenden die zuvor gezeigten Fehler- bzw. Erfolgsfunktionen. Durch diese Funktionen werden die jeweiligen Snackbar-Nachrichten ausgegeben. Die nachstehenden Screenshots zeigen einige Beispiele:



4.5. Bild Upload

Beim Einfügen von neuen Filmen wird die Möglichkeit geboten, Bilder zu laden. Nach Auswahl durch die File-Upload-Komponente werden die Bilder angezeigt. Durch das Absenden des Formulars werden die Bilder als „base64“ formatierter String an das Backend übertragen und dort gespeichert.



4.6. Pipes

Es wurden einige Pipes für diverse Umwandlungen implementiert. Die folgenden zwei Listings zeigen Beispiele.

image-endpoint.pipe.ts

```
@Pipe({
  name: 'imageEndpoint'
})
export class ImageEndpointPipe implements PipeTransform {

  transform(endpoint: string): string {
    return `${BASE}:${PORT}/${endpoint}`;
  }
}
```

Image-base-64.pipe.ts

```
@Pipe({
  name: 'imageBase64'
})
export class ImageBase64Pipe implements PipeTransform {

  transform(value: string): string {
    return 'data:image/jpeg;base64,' + value;
  }
}
```