

MovieWorld alkalmazás

Kliensoldali technológiák házi feladat – Készítette: Szőke Tibor Ádám
(GQ5E7S)

Összefoglaló

A MovieWorld kliens alkalmazás lehetővé teszi, hogy a felhasználó böngészőből kapcsolódjon a Trakt oldal backendjéhez, továbbá, hogy onnan filmek/sorozatok adatait kérdezze le és a kapott adatok között böngésszen. Az alkalmazás használatához a felhasználónak meg kell nyitnia az oldalt (a fejlesztői verzióban ez a <http://localhost:3000> címen érhető el). Az oldalra térve a navigációs menüben választhatunk a sorozatok, illetve a filmek közül. Ha kiválasztottuk az egyik menüpontot filmek/sorozatok listáját fogjuk látni tízesével, melyek között lehetőségünk van lapozni. Amennyiben kiválasztunk egy elemet, mellette megjelennek az ahhoz tartozó részletesebb adatok is, továbbá, ha kiválasztottunk egy nekünk tetsző filmet, lehetőségünk van az ahhoz kapcsolódó filmeket is kilistázni ezzel ötleteket merítve az esti filmválasztáshoz. A filmhez tartozó színészek nevére kattintva lehetőségünk van részletesebb információt kapni az adott színészeiről. Az oldal fenti részén elhelyezkedik még egy kereső is, mely segítségével könnyedén kereshetünk kulcsszavak alapján a filmek/sorozatok között. Sorozatok böngészése esetén lehetőségünk van megtekinteni az adott sorozat epizódlistáját évadokra lebontva amennyiben kiválasztottunk egy elemet.

MovieWorld Movies Shows

Search for movies...

	Title	Year	Rating
Select	Deadpool	2016	8.33
Select	Guardians of the Galaxy	2014	8.37
Select	The Dark Knight	2008	8.92
Select	Inception	2010	8.68
Select	Logan	2017	8
Select	Doctor Strange	2016	7.86
Select	The Avengers	2012	8.25
Select	Interstellar	2014	8.51
Select	Wonder Woman	2017	7.69
Select	Suicide Squad	2016	6.73

List related movies

1

Title	Year	Rating
Deadpool 2	2018	7.99
X-Men: Apocalypse	2016	7.36
Thor	2011	7.33
Fantastic Four	2005	6.11
Captain Marvel	2019	7.48

Title: Deadpool

Year: 2016

Rating: 8.33

Trailer: Link

Language: en

Genres: action adventure comedy superhero

Description:

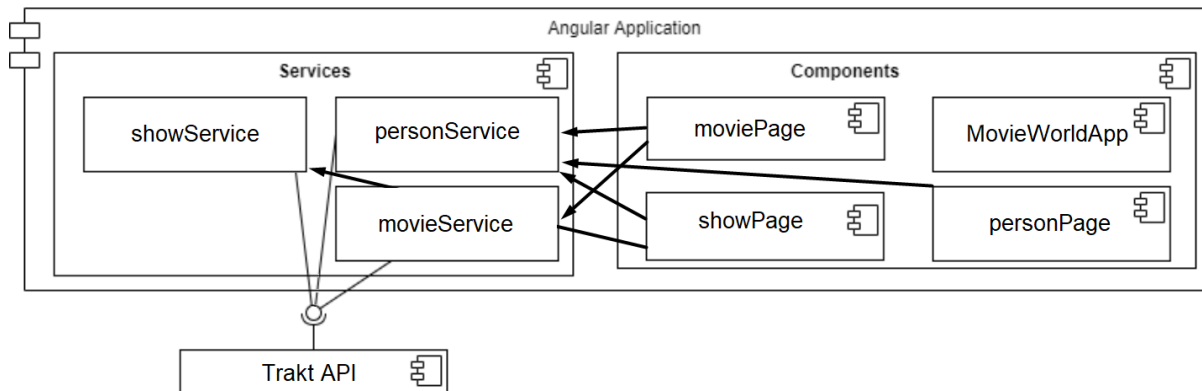
Deadpool tells the origin story of former Special Forces operative turned mercenary Wade Wilson, who after being subjected to a rogue experiment that leaves him with accelerated healing powers, adopts the alter ego Deadpool. Armed with his new abilities and a dark, twisted sense of humor, Deadpool hunts down the man who nearly destroyed his life.

Actors:

- Ryan Reynolds - Karan Soni - Ed Skrein - Michael Benyaer - Stefan Kapicic - Brianna Hildebrand - Style Dayne - Kyle Cassie - Taylor Hickson - Ayzee - Naika Toussaint - Randal Reeder - Stan Lee - T. J. Miller - Isaac C. Singleton Jr. - Morena Baccarin - Justyn Shippelt - Donna Yamamoto - Jed Rees - Hugh Scott - Gina Carano - Cindy Piper - Emily Haine - Aatash Amir - Chad Riley - Paul Belitto - Darcey Johnson - Leslie Uggams - Kyle Rideout - Jason William Day - Ben Wilkinson - Rachel Sheen - Paul Lazenby - Rob Hayter - Kayla Adams - Heather Ashley Chase - Fabiola Colmenero - Victoria De Mare - John Dryden - Jesse Michael Fullington - Tony Chris Kazoleas - Rob Liefeld - David Longworth - Sean Quan - Olesia Shewchuk - Bronwynne Sloley - Dan Zachary -

Architektúra

Az alkalmazást Angular frontenddel valósítottam meg és a Trakt publikus API-ját használtam backendként. A rendszer felépítése és az elemek főbb kapcsolódási pontjai a következők:



Komponensek

- **MovieWorldApp**: Az alkalmazás elindulásakor ez a komponens példányosodik. Szerepe a fejléc megjelenítése.
- **moviePage**: Filmekkel kapcsolatos információk megjelenítése az elsődleges szerepkörre, továbbá a personPage-re navigálásra ad lehetőséget. Az információkhoz a personService és a movieService segítségével jut hozzá.
- **showPage**: Sorozatokkal kapcsolatos információk megjelenítése az elsődleges szerepkörre, továbbá a personPage-re navigálásra ad lehetőséget. Az információkhoz a personService és a showService segítségével jut hozzá.
- **personPage**: Amennyiben valamelyik film / sorozat szereplői közül rákattintottunk az egyikre ez a komponens fog példányosodni. Segítségével szemünk elé tárul a kiválasztott színész oldala, ahol további információt szerezhethetünk róla.

Szolgáltatások

- **movieService**: A moviePage komponens számára tesz elérhetővé adatokat azáltal, hogy a Trakt publikus API-jának segítségével adatokat kér hálózaton keresztül JSON formátumban.
- **showService**: A showPage komponens számára tesz elérhetővé adatokat azáltal, hogy a Trakt publikus API-jának segítségével adatokat kér hálózaton keresztül JSON formátumban.
- **personService**: A personPage, moviePage és showPage komponensek számára tesz elérhetővé adatokat azáltal, hogy a Trakt publikus API-jának segítségével adatokat kér hálózaton keresztül JSON formátumban.

Főbb egységek

Az alkalmazás működése szempontjából az alábbi két komponens a legfontosabb:

[moviePage](#), [showPage](#)

Ez a két komponens kommunikál úgymond a felhasználóval és a felhasználó cselekedetei függvényében adatokat jelenít meg számára.

Mindkettő az alábbi oldallal kapcsolatos információkat tárolja:

- filmek / sorozatok listája
- egy kiválasztott elem ezek közül
- kapcsolódó filmek / sorozathoz tartozó epizódok listája
- kiválasztott színész

Utóbbi segítségével lehetőségünk van elnavigálni arra az oldalra, ahol a színész részletes adatait tekinthetjük meg. Az API híváshoz szükséges ennek a színésznek az ID-ja.

A listában szerepel továbbá még 4 paraméter, mely alapján a lekérdezések történnek, ezek rendre:

- egy oldalon szereplő elemek száma,
- aktuális oldalszám,
- egy kereső string

Végül de nem utolsó sorban szerepel még egy segédváltozó, melynek segítségével számon tarthatjuk, hogy az utolsó oldalon vagyunk-e vagy sem, és ennek függvényében le kell-e tiltani az előre lapozó gombot, vagy sem.

Fontos megemlíteni, hogy amennyiben a két oldal bármelyikéről a listázott színészek valamelyikét kiválasztjuk és ezáltal elhagyjuk a filmeket/sorozatok tartalmazó oldalt, a későbbiekben lehetőségünk van úgy visszatérni az elhagyott oldalra (Back gomb segítségével például), hogy annak tartalmát nem az API segítségével kell majd újra lekérnünk. Ez azért lehetséges, mert a színész oldalára való navigálás előtt az alkalmazás elmenti az oldal aktuális állapotát a localStorage-ra JSON formátumban. Visszatéréskor pedig ahelyett, hogy egy ismételt GET hívást végeznénk a Trakt backendje felé, inkább a localStorage-ből töltjük vissza az oldal állapotát ezzel erőforrást megspórolva.

Az alkalmazás működéséhez elengedhetetlen az alábbi három komponens:

[movieService](#), [showService](#), [personService](#)

Ezek a szolgáltatások lehetőséget biztosítanak a komponensek számára az adatok megjelenítésére, hiszen ezek segítségével végezzük az összes GET hívást az API segítségével. Tehát ezek segítségével történik a szerverrel való kommunikáció melyről a következőkben olvashatunk.

Trakt API

A Trakt 2-es verziójú API-ját (<http://docs.trakt.apiary.io/>) használja az alkalmazás. Ahhoz, hogy az API segítségével információkat kérhessünk le http-n keresztül szükség van 3 fejlécmező megadására minden egyes kérésnél, melyek a következők:

Header	Value
Content-type *	application/json
trakt-api-key *	Your <code>client_id</code> listed under your Trakt applications.
trakt-api-version *	2

Az ábrán látható, hogy szükséges egy bizonyos trakt-api-key érték, melyre a Trakt oldalán való regisztrálással tehetünk szert.

A szerverrel való kommunikáció megvalósítása rendkívül egyszerű feladat, szerencsére nem okozott különösebb problémát. Az oldalon megtalálható az összes GET kérés formátuma, azt némi tanulmányozás után szinte bárki tudja azt használni.

Példa:

```
return this.http.get<Show[]>
(`https://api.trakt.tv/shows/popular?extended=full&page=${page}&limit=${pageSize}&query=${query}`,
{headers: this.headers});
```

Ebben a kérésben látható, hogy http protocol segítségével zajlik a kommunikáció és egy GET kérésről van szó. Azt is szemügyre vehetjük, hogy egy Show elemeket tartalmazó tömb fog érkezni válaszként JSON formátumban.

A szükséges információkat a headers változóba helyeztük el és ezt minden egyes hívás esetén továbbítanunk kell a Trakt szervere felé.

```
this.headers = new HttpHeaders({
  "Content-type": "application/json",
  "trakt-api-key": "161402e97f36ba2419b888c48e2ad67cb54cb4d3fc0c4daa8d79c24c9d23b7df",
  "trakt-api-version": "2"
});
```

Az url-ből kiolvasható, hogy ez a kérés a populáris sorozatok halmazából szeretne információt nyerni. A „?extended=full” rész felelős azért, hogy az érkező JSON formátumú adatok minden részletet tartalmazzanak, ne csak az alapvető tulajdonságokat mint például a cím és az id-k. Ezt követően látható a page és a pageSize érték, melyek azt határozzák meg hogy a több száz / több ezer adatból mennyit és az adathalmaz mely részéről juttasson el (elejéről/végéről/stb). Ezek segítségével lehetőségünk van ergonomikusabb formában böngészni a témérdek adatot. Végül pedig látható a query melynek segítségével kulcsszavakat megadva szűrhetünk a találatként rendelkezésünkre bocsátott sorozatok listáján.

Film kiválasztása

Az alábbiakban részletesen áttekintjük, milyen lépések történnek, amikor a felhasználó kiválaszt egy filmet a listából.

Rákattintunk a bekeretezett gombra:

MovieWorld	Movies	Shows
Fight Club		
Title	Year	Rating
<div>Select</div> Fight Club	1999	8.85

Az alábbi ábrán látható, hogy milyen folyamatok fognak végbe menni a kattintás hatására:

```
<a [class.active]="movie === selectedMovie"
  (click)="selectedMovie = movie;
  getActors();
  getRelatedMovies();
  toggleRelateMovies = false">
  <td><button type="button" class="btn btn-primary">Select</button></td></a>
```

1. Az oldal selectedMovie tulajdonságába elmentjük a kiválasztott filmet.
2. Lekérjük a filmhez tartozó színészeket:

```
getActors() {
  this.personService.getPeopleOfMovie(this.selectedMovie.ids.trakt)
```

Látható, hogy a personService segítségével véghez viszünk egy hívást a kiválasztott film Trakt ID-jának paraméterként való megadásával.

```
getPeopleOfMovie(movieId: number) : Observable<People> {
  return this.http.get<People>(`https://api.trakt.tv/movies/${movieId}/people`, {headers: this.headers});
}
```

Ez a hívás egy olyan adatszerkezettel tér vissza, melyben szerepelnek az adott filmhez tartozó szereplők.

```

getActors() {
  this.personService.getPeopleOfMovie(this.selectedMovie.ids.trakt)
    .subscribe(people => {
      if (!people)
        throw "Hiba történt az adatlekérdezés során.";
      this.selectedMovie.actors = [];
      people.cast.forEach(cast => this.selectedMovie.actors.push(cast.person));
    });
}

```

Az így kapott adatot először meg kell vizsgálnunk, hogy üres-e, mert amennyiben igen, úgy hiba történt az adatlekérdezés során és ezt jeleznünk kell a felhasználó számára. Ezt követően pedig feltöltjük a kiválasztott filmhez tartozó színészek tömböt, hogy azt a komponens meg tudja majd jeleníteni.

Fontos megjegyezni, hogy a subscribe metódus segítségével aszinkron működést tudunk megvalósítani. Amikor a hívást kiadtuk nem várakozunk a válasza a szálát blokkolva egészen addig amíg az meg nem érkezik, hanem visszaadjuk azt és majd akkor foglaljuk le ismét a szálát amikor megérkezik az adat (ez már a subscribe híváson belül van).

3. Ezek után lekérdezzük a kiválasztott filmhez kapcsolódó filmeket is.

Ez szintén egy API hívás-t fog eredményezni, azonban ezt a kérést már nem a personService, hanem a movieService fogja számunkra végrehajtani.

```

getRelatedMovies() {
  this.movieService.getRelatedMovies(this.selectedMovie.ids.trakt)
    .subscribe(movies => {
      if (!movies)
        throw "Hiba történt az adatlekérdezés során.";
      this.relatedMovies = movies;
      this.ratingRounding(this.relatedMovies);
    });
}

```

```

getRelatedMovies(movieId: number) : Observable<Movie[]> {
  return this.http.get<Movie[]>(`https://api.trakt.tv/movies/${movieId}/related?extended=full`, {headers: this.headers});
}

```

A folyamat nagyon hasonló az előzőhöz így ezt most nem részletezem ismételtén, a lényeges folyamat az az, hogy az oldal tulajdonságai közé elmentjük a kiválasztott filmhez kapcsolódó filmek listáját, hogy amennyiben a felhasználó később szeretné azokat listázni, azok rendelkezésre álljanak.

4. Legeslegvégül pedig beállítjuk a *toggleRelatedMovies* értéket hamisra.

Azért teszünk így, mert a felhasználó új filmet választott ki és nem biztos, hogy minden kiválasztott filmhez rögtön szeretné kilistázni a kapcsolódó filmeket is.

Források

Az ábra a <https://www.diagrams.net/> segítségével készült.